

Prikupljanje nestrukturiranih dnevničkih zapisa za otkrivanje poslovnih procesa

Delač, Josip

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:067197>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-14**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 3169

**PRIKUPLJANJE NESTRUKTURIRANIH DNEVNIČKIH ZAPISA
ZA OTKRIVANJE POSLOVNIH PROCESA**

Josip Delač

Zagreb, veljača 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 3169

**PRIKUPLJANJE NESTRUKTURIRANIH DNEVNIČKIH ZAPISA
ZA OTKRIVANJE POSLOVNIH PROCESA**

Josip Delač

Zagreb, veljača 2024.

DIPLOMSKI ZADATAK br. 3169

Pristupnik: **Josip Delač (0036502684)**
Studij: Računarstvo
Profil: Računarska znanost
Mentor: izv. prof. dr. sc. Bruno Blašković

Zadatak: **Prikupljanje nestrukturiranih dnevničkih zapisa za otkrivanje poslovnih procesa**

Opis zadatka:

Dubinska analiza procesa sastoji se u otkrivanju, analizi i poboljšanju procesa otkrivanjem znanja iz dnevničkih zapisa. Prvi korak dubinske analize procesa je definiranje strukture podataka iz nestrukturiranih zapisa s internetskih stranica. U radu je potrebno za odabrani skup internetskih stranica realizirati prikupljanje zapisa u stvarnom vremenu. Nakon toga potrebno je nestrukturirane zapise pretvoriti XES standardizirani format podataka. Za ilustraciju koristiti odabrani skup javno dostupnih internetskih stranica. Dobile XES datoteke potrebno je verificirati primjenom odabranog skupa programskih alata iz domene analize poslovnih procesa. Posebnu pažnju treba posvetiti dohvatima dinamičkih web stranica, definiranju konzistentnih tipova podataka iz dnevničkih zapisa kao i konzistentnom povezivanju XES datoteka sa programskim alatima za dubinsku analizu procesa.

Rok za predaju rada: 9. veljače 2024.

Sadržaj

Table of Contents

Uvod.....	1
1. Dubinska Analiza Procesa	2
1.1. XES Format.....	4
1.1.1. Struktura.....	5
2. Strukturiranje podataka za dubinsku analizu procesa.....	10
3. Pomoćni alati	11
3.1. Selenium.....	11
4. Alati za dubinsku analizu procesa.....	14
4.1. PM4PY	14
4.2. PMTK.....	16
4.3. ProM.....	18
5. Implementacija rješenja	20
5.1. Izvor podataka	20
5.2. Pretvorba u dnevnički zapis	25
6. Analiza dobivenih rješenja.....	28
6.1. Provjera ispravnosti.....	28
6.2. Analiza jednog procesa	29
7. Moguće nadogradnje.....	31
7.1. Nadogradnja podataka.....	31
7.2. Nadogradnja pretvarača.....	31
7.3. Nadogradnja analize.....	32
Zaključak.....	33

Literatura.....	34
Sažetak	35
Summary	36

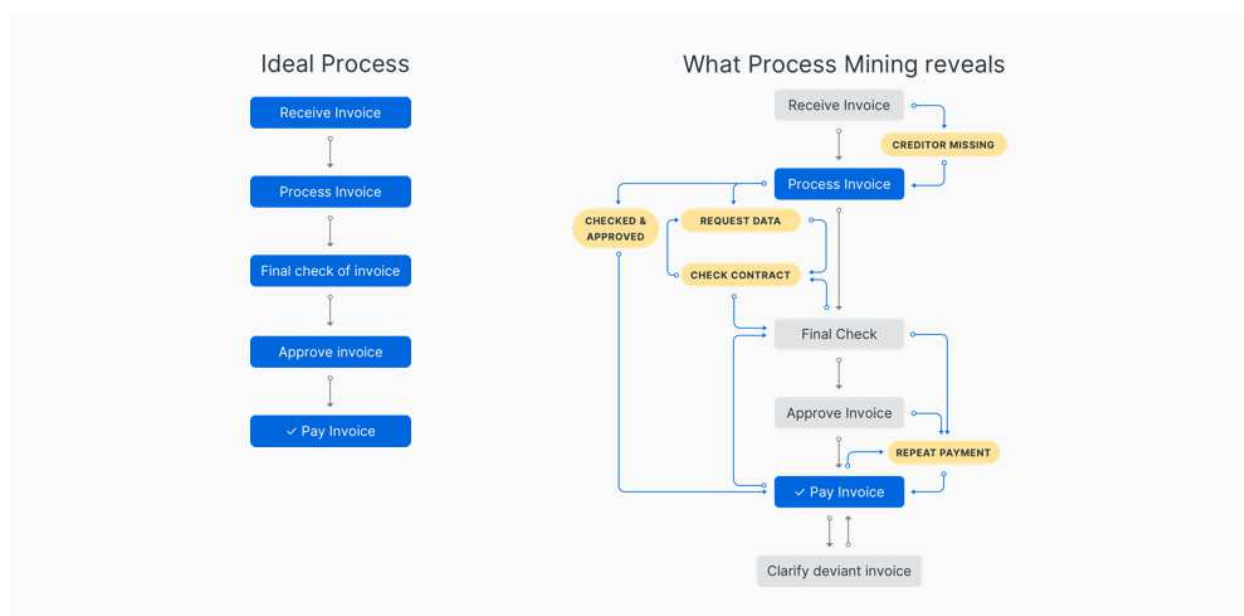
Uvod

U današnjem svijetu poslovnih procesa stalno se teži pronalaženjem mogućih rješenja povećanja efikasnosti. Pronalaženje tih mogućih optimizacija često radimo analizom stvarnih podataka i praćenjem odstupanja od očekivanih vrijednosti. Znanost koja se bavi baš ovakvim problemima se zove „Process Mining“. Process mining postupcima možemo preko zapisa u dnevniku o događajima unutar procesa konstruirati potpuni proces koji je onda jednostavnije usporediti sa očekivanim ponašanjem. Problem se ovdje pojavljuje kod nalaženja konzistentnog, reprezentativnog i potpunog izvora tih podataka za analizu. Ako nam naši zapisi ne pokrivaju svaki korak u poslovnom procesu generirani model neće biti precizan te bi moglo doći do pogrešnih zaključaka.

U ovom radu ćemo modelirati poslovne investicijskih fondova dostupnih na stranici HR Portfolio. Ovi podatci su zapravo akumulirani zapisi svih unutarnjih poslovnih procesa određenog fonda te izravno govore o efikasnosti rada poslovnog procesa. Problem u ovom slučaju je dinamičnost stranice što nam otežava dohvaćanje i pohranu podataka. U ostatku rada će biti prikazan način rješavanja ovog problema te formatiranje dohvaćenih podataka radi daljnje anali

1. Dubinska Analiza Procesa

Dubinska Analiza Procesa (Process Mining) je disciplina koja se bavi specijaliziranim algoritmima za obradu dnevničkih zapisa radi identificiranja uzoraka, trendova i detalja u izvršavanju poslovnog procesa. Process mining se fokusira na slijed označenih događaja koji su se dogodili u određenom slučaju.



Slika 1 Primjer korištenja modeliranja procesa

U ovom slučaju vidimo da je postupak analize dnevničkih datoteka pronašao dodatne korake izvedbe koji nisu bili planirani. Ovi neplanirani koraci nam omogućuju uvođenje mjera koje bi te korake mogli ili eliminirati ili ubrzati da dobijemo izvršenje što sličnije željenom.

Iako Process Mining i Data Mining oboje rade nad velikim skupovima podataka postoje ključne razlike. Process Mining se bavi isključivo dnevničkim zapisima koji se potom mogu koristiti za modeliranje stvarnih procesa dok se Data Mining bavi puno širim skupom mogućih podataka za svrhe koje su daleko šire od samog modeliranja procesa. Na primjer to bi mogli biti pronalaženje neispravnih podataka, predviđanje budućih vrijednosti itd.

Process Mining disciplina se sama dijela na tri podvrste:

- **Otkrivanje (Discovery)**

Generiramo model procesa izravno iz podataka bez vanjskih utjecaja. Pod ovom klasifikacijom nemamo nikakvo prijašnje znanje o tome kako bi model traženog procesa izgledati. Ova vrsta se najčešće koristi u praksi.

- **Sukladnost (Conformance)**

Analiza sukladnosti procesa se koristi kada imamo pristup modelu traženog procesa i stvarnim dnevničkim zapisima izvršavanja tog procesa. Usporedbom ovih podataka možemo dobiti lagani uvid da li je stvaran proces sukladan traženome. Ovo u kompliciranim sustavima nije moguće napraviti „odokativno“ te nam Process Mining omogućuje robustan i ponovljiv način provjere.

- **Optimizacija (Enhancement)**

U ovoj vrsti koristimo dodatnu vanjsku informaciju za doradu modela kako bi dobili povećanje produktivnosti procesa (zarada, uloženo vrijeme...). Optimizacija se često događa nakon provjere sukladnosti procesa pri kojem određene neefikasnosti postanu vidljive u procesu.

1.1. XES Format

Dnevnički zapisi se mogu pojaviti u mnogobrojnim oblicima, skoro svaki složeniji sustav ima neku vrstu zapisivanja događaja. Iako je ovo dobra stvar sa pogleda da je uvijek moguće doći do nekakvih podataka o procesu donosi nam novi problem. Kako obrađivati toliko različitih vrsta podataka?

Rješenje ovog problema se nalazi u standardizaciji dnevničkih zapisa. Standard koji se koristi u ovom slučaju je XES.

XES format je format za dnevničke datoteke koji je baziran na XML zapisu te služi za pružanje jedinstvenog formata za interakciju između aplikacija i alata koje koristimo za njihovu analizu. Kod dizajna XES formata vodilo se računa o 4 osnovna načela:

- **Jednostavnost**

Korištenje najjednostavnije moguće strukture za prikaz podataka radi jednostavnog parsiranja i pregleda sa programske podrške i isto tako i sa strane ljudi.

- **Fleksibilnost**

XES standard mora biti u mogućnosti prikazati bilo koji događaj koji se može dogoditi u bilo kojem sustavu. Zbog ovoga nisu rađene pretpostavke o tome da su događaji iz poslovnog procesa te se zato mogu koristiti za sve dnevničke zapise.

- **Proširljivost**

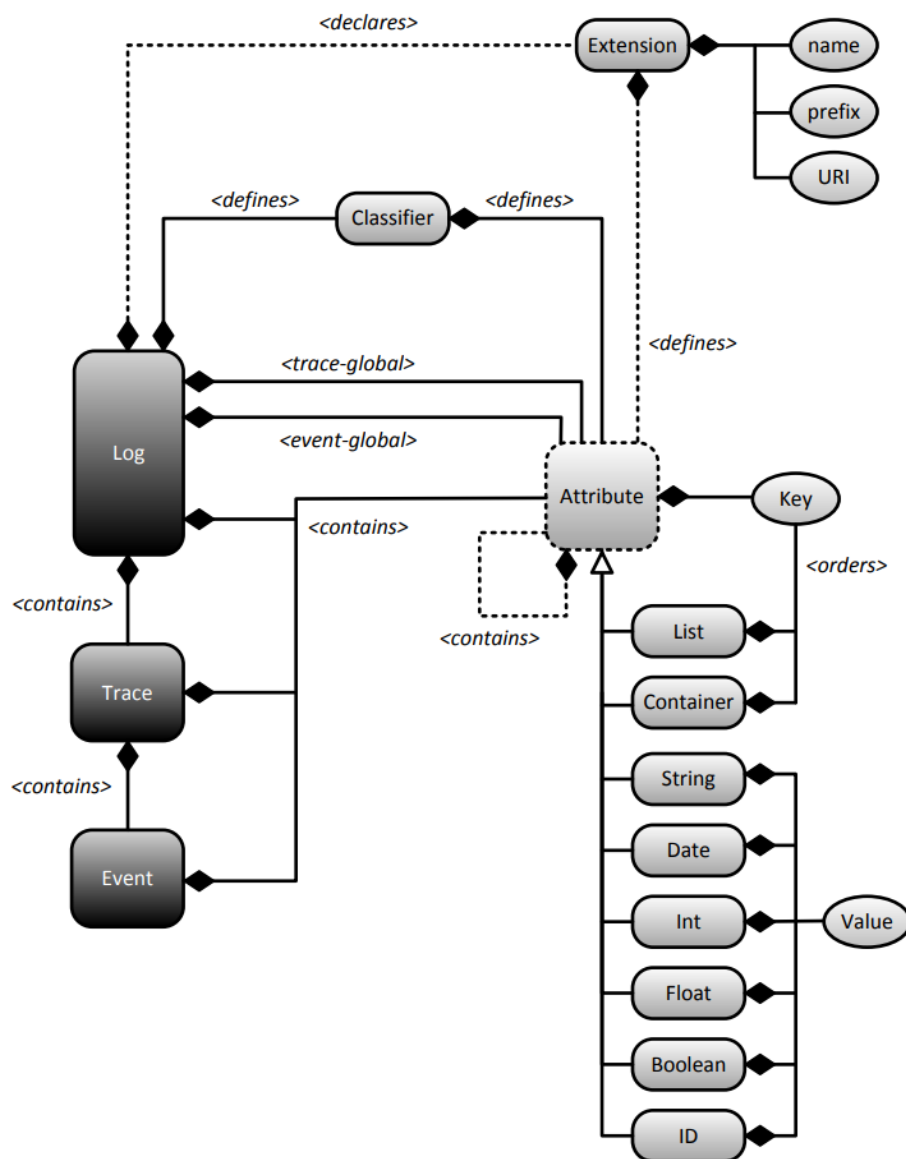
Standard mora biti proširljiv u budućnosti ako je potrebno bez gubitka transparentnosti i kompatibilnosti sa budućim ili prijašnjim verzijama. Primjer proširivanja standarda bi mogao biti dodatak dodatnih potrebnih podataka o događaju u nekoj domeni (obavezan OIB za pacijenta..)

- **Izražajnost**

Iako se teži što većoj fleksibilnosti standarda ne smije doći do gubitka informacije u bilo kojem slučaju. Ovo se postiže strogim tipovima podataka pri upisu te mogućnosti dodavanja ljudske semantike događaju.

1.1.1. Struktura

IEEE je 2016 godine izbacio prvi standard za XES format za zapis dnevnika datoteka. Iako je glavna svrha Process Mining može se koristiti i za generalan Data Mining, Text Mining i statističku analizu.



Slika 2 UML Dijagram XES Formata

U ovom UML dijagramu vidimo strukturu standarda te ćemo proći malo opširnije kroz osnovnu strukturu.

1.1.1.1 Log

Svaka XES dnevnička datoteka se sastoji od jednog glavnog objekta i to je log. On se sadrži sve moguće podatke o događajima procesa. Ovaj objekt sadrži dva atributa:

- **xes.version** - Verzija XES standarda koji se koristi (eg. 1.0)
- **xes.features** – Lista korištenih vrsta atributa u dokumentu odvojenih razmakom

```
<log xes.version="2.0" xes.features="nested-attributes">
```

Kod 1 Primjer Log objekta

1.1.1.2 Trace

Jedan log sadrži skup **Trace** objekata koji svaki označuju jedan slučaj izvršavanja procesa. Primjer jednog slučaja izvršavanja procesa bi mogao biti jedan pristup jednog korisnika web stranici. Trace objekt nema definirane attribute.

```
<trace>  
  ....  
</trace>
```

Kod 2 Sintaksa Trace objekta

1.1.1.3 Event

Event objekt sadrži skup atributa koji opisuju jedan događaj. U analogiji gdje je jedan pristup jednog korisnika web stranici Trace, Event bi bio skidanje jedne slike.

```
<event>  
  ....  
</event>
```

Kod 3 Sintaksa Event objekta

1.1.1.4 Atributi

Do sad navedeni objekti ne sadrže nikakve informacije sami nego samo strukturiraju zapis. Sve vrijednosti o događajima u dnevničkoj datoteci su spremljeni u atributima. Svi atributi moraju imati ključ jedinstven u svome objektu.

Atributi mogu biti različitih vrsta koje su nam poznate iz područja programske potpore.

```
<string key="city" value="NewYork" />
<date key="timestamp" value="2023-11-25T19:45:32.345+02:00" />
<int key="counter" value="25" />
<float key="float" value="32.23" />
<boolean key="failed" value="false" />
<id key="unique_id" value="ab7425fa4c2a32245bb568ff4c6c9639" />
<list key="revisions">
  <string key="name" value="XES standard" />
  <boolean key="stable" value="true" />
  <string key="revision" value="2.0" />
  <string key="revision" value="1.4" />
  <string key="revision" value="1.3" />
  <string key="revision" value="1.2" />
  <string key="revision" value="1.1" />
  <string key="revision" value="1.0" />
</list>
```

Kod 4 Primjer atributa

1.1.1.5 Ugniježdeni atributi

Radi fleksibilnosti strukture postoji mogućnost ugniježdenih atributa. Ako će dnevnički zapis zahtijevati ovakve zapise to se treba navesti unutar log objekta u xes.features atributu.

```
<string key="country" value="Uzbekistan">
  <boolean key="visited" value="true" />
</string>
```

Kod 5 Primjer ugniježdenih atributa

1.1.1.6 Globalni Atributi

Globalni atributi se definiraju na razini log objekta i definiraju se odvojeno za Trace i za Event objekte. Oni nam govore koja polja će biti dostupna i definirana za svaki objekt.

```
<global scope="event">
  <string key="name" value="" />
</global>
```

Kod 6 Definiranje globalnih atributa

U ovom primjeru nam je definirano da će svaki Event u sebi sadržavati atribut „name“.

1.1.1.7 Klasifikatori događaja

Klasifikatori događaja nam omogućuju fleksibilan način definicije logike identificiranja i usporedbe više događaja. Nabrojani atributi u klasifikatoru se pri analizi koriste za usporedbu događaja te ako su isti ti se događaji računaju kao istovjetni.

```
<classifier name="Activity classifier" keys="lifecycle:transition"/>
```

Kod 7 Definiranje klasifikatora

U ovom slučaju su dva događaja istovjetna ako su im atributi sa ključem „lifecycle:transition“ iste vrijednosti.

1.1.1.8 Ekstenzije

Ekstenzije nam služe kao nadogradnja XES standarda koja nam omogućuje definiranje seta atributa koji su dostupni na bilo kojoj razini XES hijerarhije.

```
<extension name="Concept" prefix="concept" uri="http://www.xes-
standard.org/concept.xesext"/>
```

Kod 8 Dodavanje Ekstenzije

```

<log xes.version="2.0" converter.version="0.3_JD" xmlns="http://www.xes-
standard.org/">
  <classifier name="Activity classifier" keys="lifecycle:transition" />
  <string key="origin" value="csv" />
  <trace>
    <string key="concept:name" value="OTP start" />
    <event>
      <string key="lifecycle:transition" value="INITIAL" />
      <string key="concept:name" value="INITIAL" />
      <float key="change" value="0.0124" />
      <float key="value" value="17.803" />
      <float key="fearAndGreed" value="74.2857142857143" />
      <date key="time:timestamp" value="2024-01-30T15:02:39.448+01:00"
    />
    <int key="@@index" value="0" />
  </event>
  <event>
    <string key="lifecycle:transition" value="UP" />
    <string key="concept:name" value="UP" />
    <float key="change" value="0.0028" />
    <float key="value" value="17.8035" />
    <float key="fearAndGreed" value="72.4857142857143" />
    <date key="time:timestamp" value="2024-01-31T15:02:39.448+01:00"
  />
  <int key="@@index" value="1" />
</event>
<event>
  <string key="lifecycle:transition" value="DOWN" />
  <string key="concept:name" value="DOWN" />
  <float key="change" value="-0.0073" />
  <float key="value" value="17.8031" />
  <float key="fearAndGreed" value="66.9714285714286" />
  <date key="time:timestamp" value="2024-02-03T15:02:39.448+01:00"
/>
  <int key="@@index" value="2" />
</event>
</trace>
</log>

```

Kod 9 Primjer XES datoteke

2. Strukturiranje podataka za dubinsku analizu procesa

Kao što je spomenuto u uvodu veliku prepreku kvalitetnoj i uspješnoj dubinskoj analizi procesa čini dohvaćanje potpunih standardiziranih podataka. Zbog velike proširenosti zapisivanja podataka u tabličnom obliku postoji nekoliko automatskih rješenja koji se bave pretvorbom u XES format.

The screenshot shows a configuration window titled "Assign the columns to the key attributes". It includes instructions: "Pick a key attribute (e.g. Case ID, Activity), and then click the columns you want to assign to it. (Hold # or CTRL to deselect any column.)". There are four attribute selection boxes: "Case ID *", "Activity *", "Timestamp *", and "Resource". Below these, a table of columns is displayed with a "Datum" column selected. A red message states "At least one column is required for CASE." The table contains the following data:

Fond	Datum	Cijena	Val.	+/- %	2023 %	2024 %	1mj %	3mj %	6mj %	12mj %	PGP 3g %	PGP 5g %	Trajanje (s)
Erste Adriatic Multi Asset2	2024-01-30	94364	€	0.32	7.73	0.33	0.35	6.72	3.8	6.02	-2.74		4.54
Erste Adriatic Multi Asset2	2024-01-31	943838	€	0.02	7.73	0.35	0.35	6.56	3.82	6.34	-2.74		4.54
Erste Adriatic Multi Asset2	2024-02-03	948015	€	0.11	7.73	0.58	0.58	6.54	4.23	6.44	-2.72		4.55

At the bottom, there are buttons for "Change Import Settings", "Close", and "Start Import".

Slika 3 Primjer postojećeg pretvarača

Problem sa automatiziranim alatima za pretvorbu je što još uvijek nije moguće jednostavno dodati neka izračunata polja svakom zapisu i odraditi filtriranja ili neke druge željene operacije.

Na sreću CSV format je vrlo jednostavan za rad u raznim programskim jezicima te nam je uz praćenje XES standarda relativno jednostavno ručno obaviti pretvorbu kao što ćemo raditi i u ostatku ovog rada.

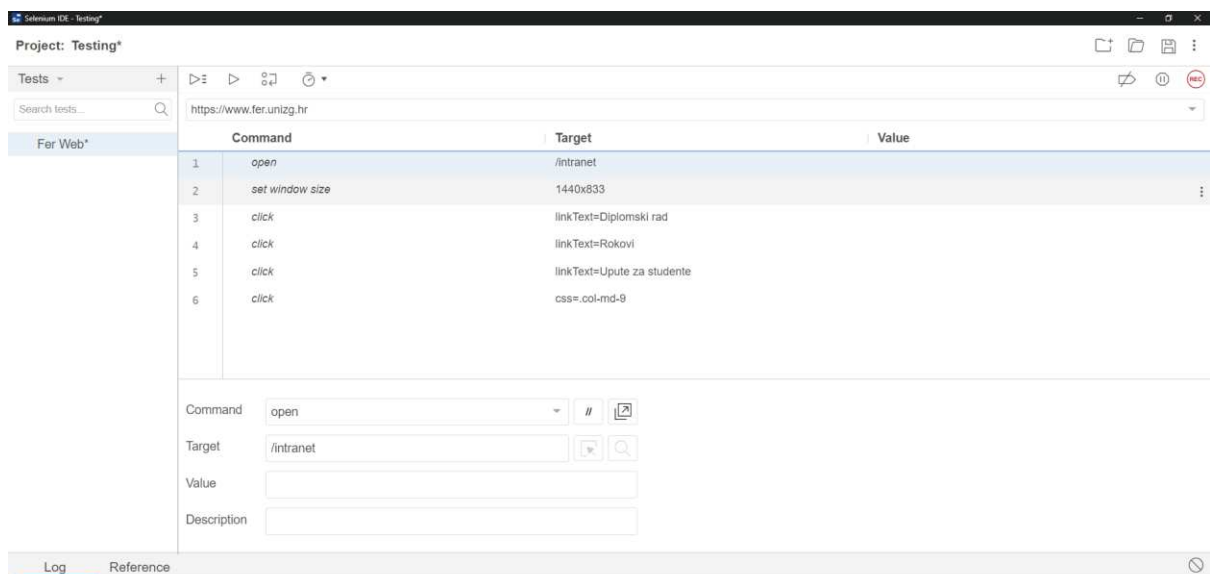
3. Pomoćni alati

3.1. Selenium

Selenium je projekt koji se sastoji od raznih alata i biblioteka koji služe za automatizaciju Internet preglednika. Najčešće korištenje ovog skupa alata je za automatizirano testiranje web stranica da se pri manjim promjenama na ponovljiv način može brzo i kvalitetno testirati da nije došlo do neželjenih promjena. Sam projekt se dijeli na više komponenti koje ćemo pobježe opisati u sljedećim poglavljima.

Selenium IDE

Selenium IDE je integrirano okruženje u kojemu možemo razvijati automatizirane testove za stranice. Omogućuje nam snimanje interakcije, uređivanje postojećih testova. Samo okruženje je dostupno kao dodatak za Google Chrome ili Mozilla Firefox preglednike.



Slika 4 Primjer izgleda IDE

U primjeru sa slike se snimalo korištenje stranice FER gdje vidimo kako je korisnik mijenjao veličinu ekrana i koristio stranicu. Same interakcije su snimljene u Selenese jeziku. Jezik se

sastoji od komandi njihovih traženih elemenata i vrijednosti. Sami primjer ovog jezika se može također vidjeti na slici.

Selenium client API

Selenium također omogućuje korištenje svojih alata i biblioteka preko sučelja za većinu popularnijih programskih jezika. U ovom slučaju osim samih testova stranice Selenium se često i koristi za automatsko dohvaćanje podataka sa dinamičkih stranica koje imaju zaštitu protiv automatskih agenata.

```
from selenium import webdriver

driver = webdriver.Chrome()

driver.get("https://www.fer.unizg.hr/intranet")

driver.quit()
```

Kod 10 Primjer jednostavnog korištenja

U ovom radu ćemo koristiti ovu komponentu Seleniuma preko sučelja za Python.

Selenium Remote Control

Ova komponenta je implementirana kao server koji koristi programski jezik Java. U ovom slučaju Selenium instanca se može pokrenuti na potpuno odvojenom računalu jer s njom komuniciramo preko HTTP protokola. U svakom zahtjevu se šalje skripta napisana u prije spomenutom Selenese skriptnom jeziku. Ovo omogućuje korištenje ovih alata iz svakog programskog jezika koji podržava HTTP zahtjeve.

```
self.selenium = selenium("localhost", 4444, "*firefox",
                        "https://www.fer.unizg.hr/intranet")
self.selenium.start()
```

Kod 11 Selenium RC korištenje

Selenium WebDriver

Selenium WebDriver je sučelje koje omogućuje da željene naredbe rade očekivano i konzistentno na različitim preglednicima. Ovo je moguće zbog implementacije jedinstvenog upravljačkog programa za svaki preglednik. Svaki upravljački program prihvaća Selenese naredbe preko konzistentnog sučelja te ih pretvara u naredbe jedinstvene pregledniku.

```
from selenium import webdriver
with webdriver.Firefox() as driver:
    driver.get("https://www.fer.unizg.hr/intranet")
    print(driver.page_source)
```

Kod 12 Korištenje WebDriver komponente

4. Alati za dubinsku analizu procesa

4.1. PM4PY

PM4PY je biblioteka za programski jezik Python koja omogućuje jednostavan i pristupačan početak rada sa dnevničkim zapisima. U samoj biblioteci su implementirani mnogi moderni Process Mining algoritmi. Sam projekt je otvorenog koda čime se i omogućuje korisnicima da doprinose i rade na mogućim problemima.

Sama biblioteka podržava rad sa dnevničkim zapisima u CSV i u XES formatu. Upis CSV formata je moguć korištenjem poznate Python biblioteke zvane Pandas.

```
import pm4py

if __name__ == "__main__":
    log = pm4py.read_xes('Log.xes')
    process_model = pm4py.discover_bpmn_inductive(log)
    pm4py.view_bpmn(process_model)
```

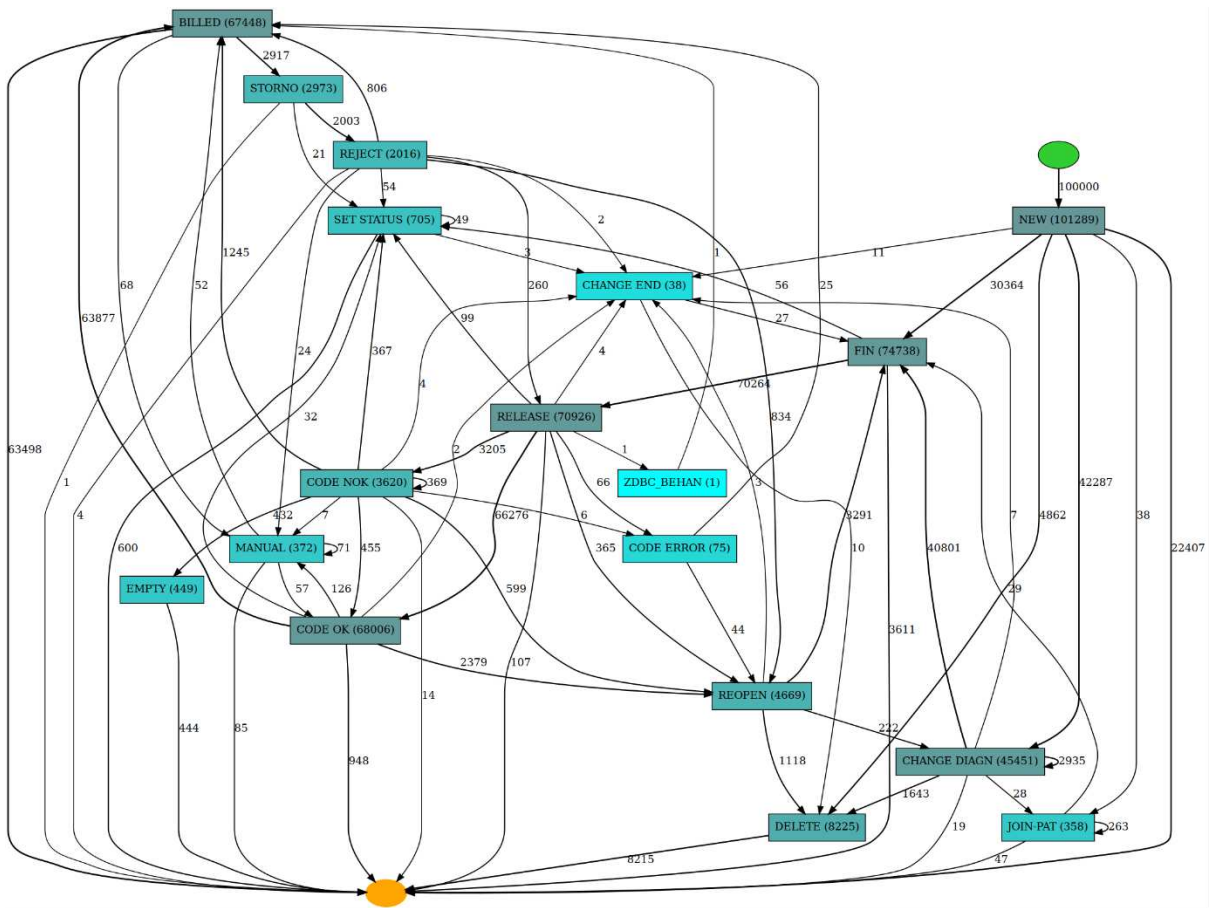
Kod 13 Rad sa XES datotekom

```
import pandas
import pm4py

def import_csv(file_path):
    event_log = pandas.read_csv(file_path, sep=';')
    event_log = pm4py.format_dataframe(event_log, case_id='case_id',
activity_key='activity', timestamp_key='timestamp')
    start_activities = pm4py.get_start_activities(event_log)
    end_activities = pm4py.get_end_activities(event_log)

if __name__ == "__main__":
    import_csv("Log.csv")
```

Kod 14 Rad sa CSV datotekom

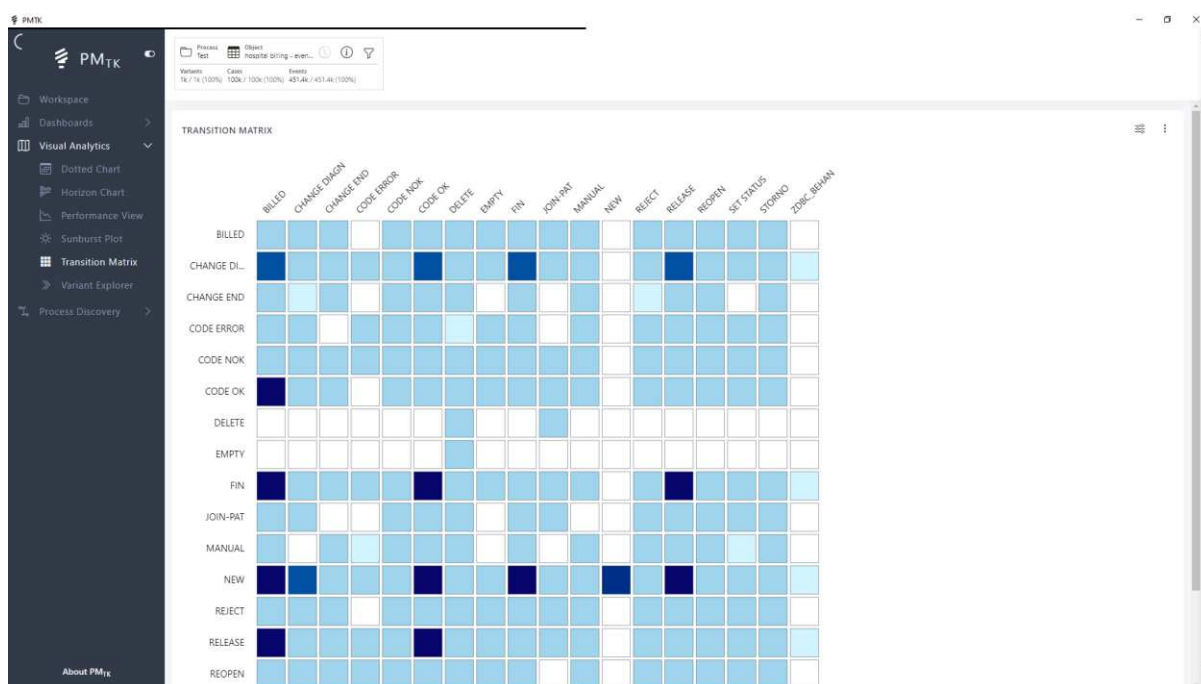


Slika 5 Primjer generiranog heurističkog grafa

4.2. PMTK

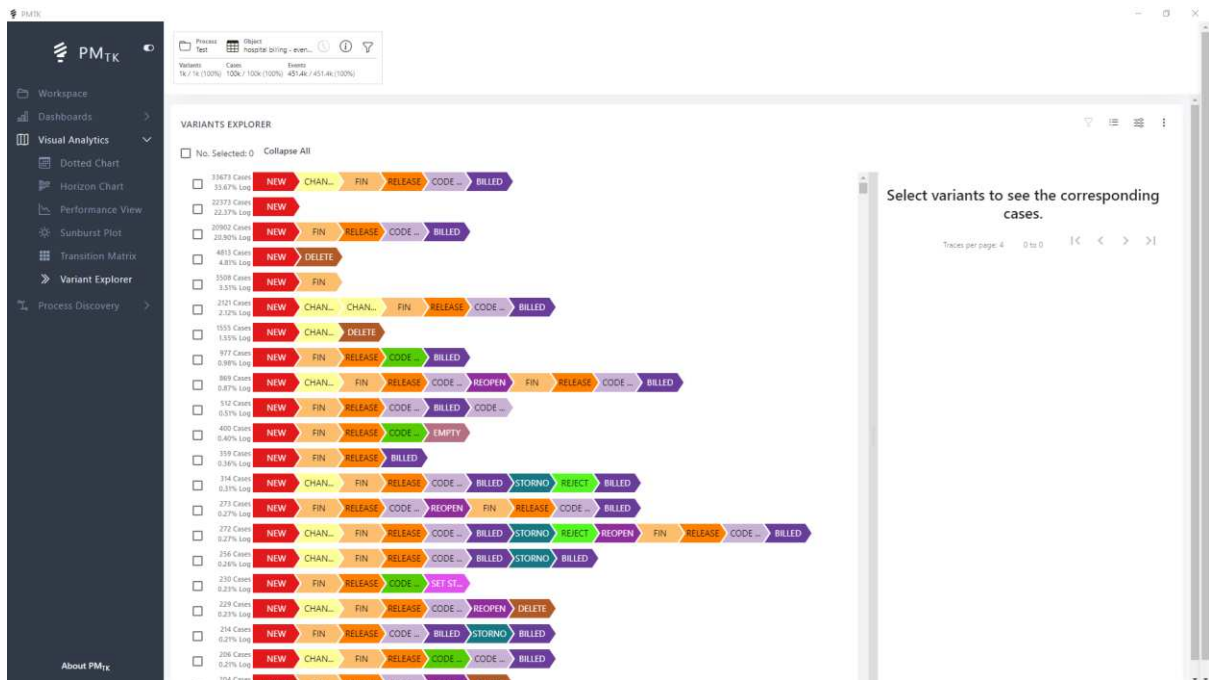
PMTK je alat od tvorca PM4PY biblioteke koja pomoću web baziranog sučelja daje grafičko sučelje za PM4PY implementacije. U samom sučelju je jednostavno vidjeti analizu ponašanja procesa.

Iako je program sposoban obaviti otkrivanje poslovnog procesa i generiranje stabla izvršavanja procesa razlikuje se od većine ostalih odličnim vizualnim prikazom analitike događaja.



Slika 6 Tranzicijska matrica

U ovoj slici je prikazana generirana tranzicijska matrica koja nam na jednostavan način vizualizira frekvencije prijelaza iz pojedinih stanja što nam može biti vrlo korisno u optimizaciji.

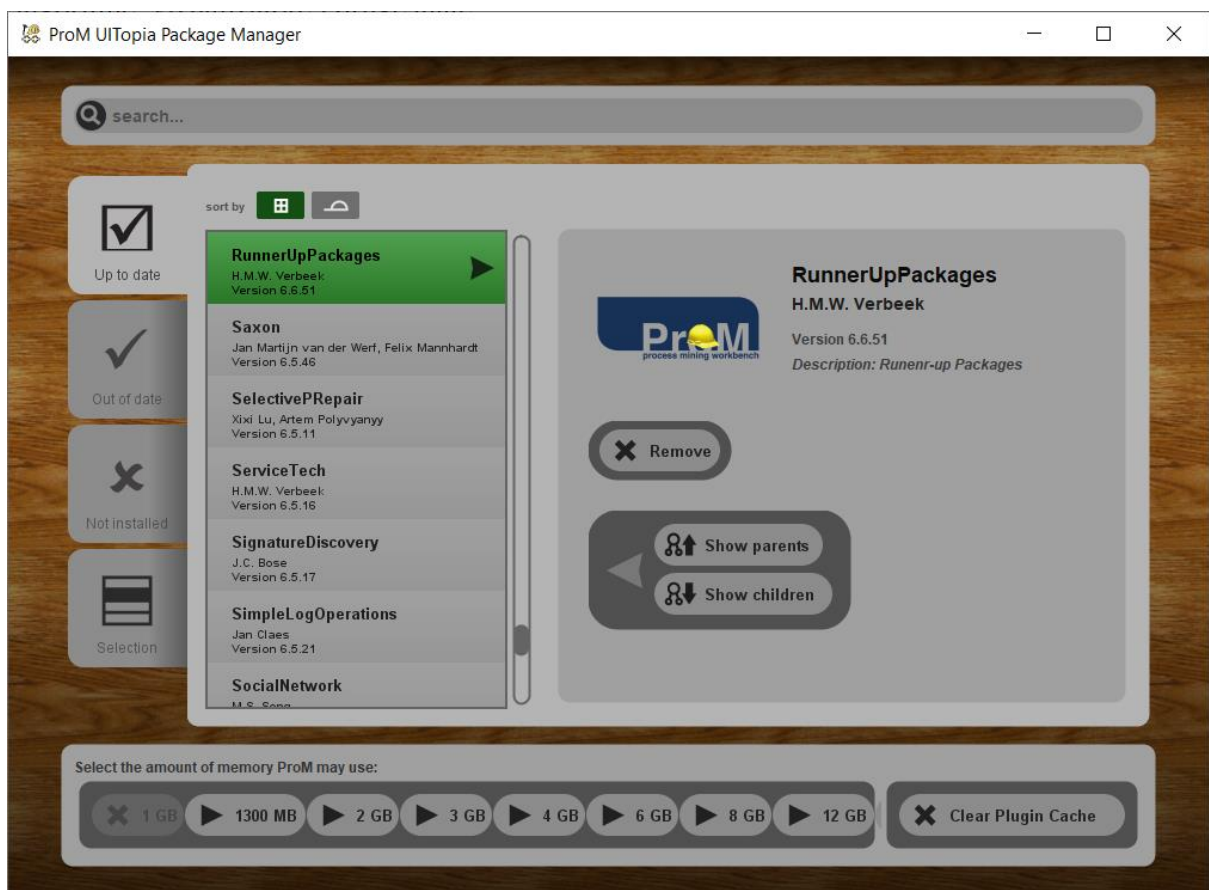


Slika 7 Variant explorer

U ovom ispisu su prikazani svi mogući slučajevi izvršavanja procesa što nam omogućuje da označimo samo one koji su imali neželjene korake te po njima pokušamo optimizirati proces.

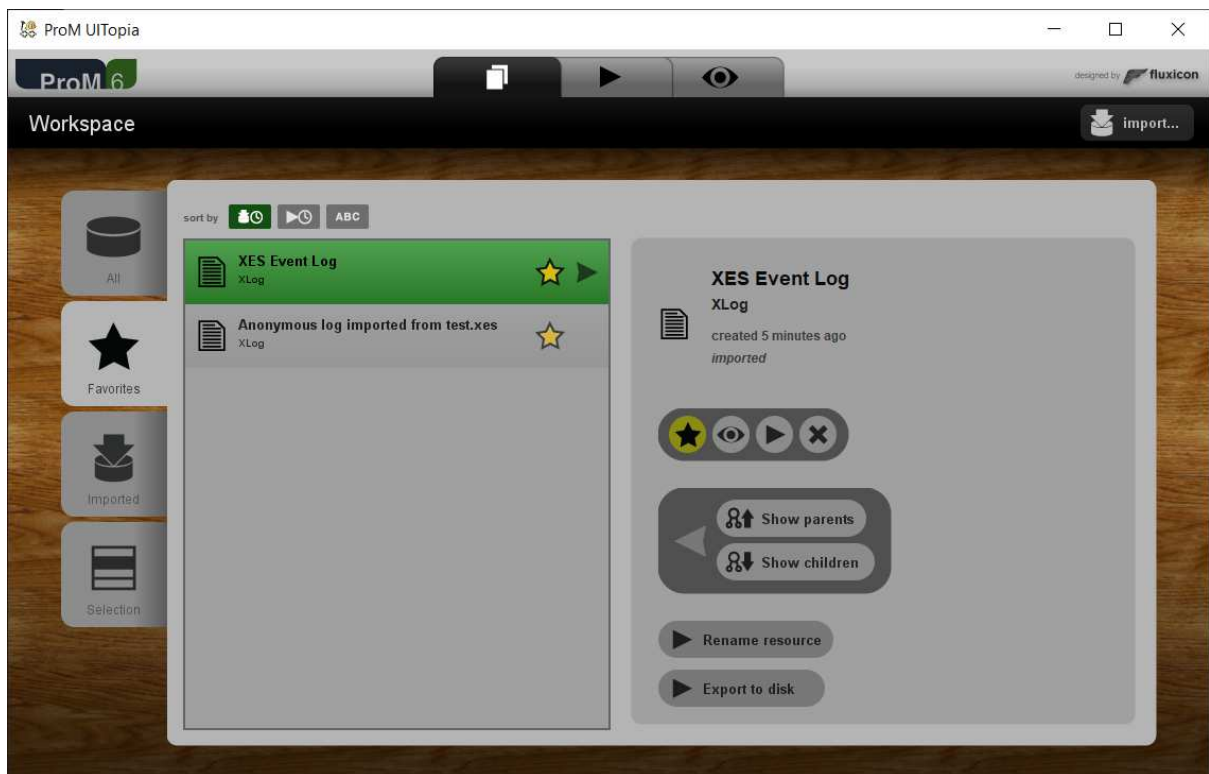
4.3. ProM

ProM je proširiv skup alata napisan u programskom jeziku Java što mu omogućuje jednostavno podržavanje više operacijskih sustava. Iako postoji osnovan standardni skup proširenja koji se može instalirati preko upravljača paketima ProM potiče istraživače da dodaju svoje dodatne algoritme, vizualizatore i druge alate.



Slika 8 ProM upravljač paketima

Na prijašnjoj snimci su vidljivi standardni alati pod nazivom „RunnerUpPackages“ ali su također vidljivi ostali specijalizirani paketi koje po potrebi možemo dodati.



Slika 9 Izgled samog ProM programa

Iako je ovaj projekt najduže dostupan za dubinsku analizu procesa često dolazi do neočekivanih problema sa smrzavanjem programa i neočekivanim gašenjem. Radi ovoga ćemo većinom koristiti prošla dva programa za što je moguće.

5. Implementacija rješenja

5.1. Izvor podataka

Kao glavni izvor podataka koristimo web stranicu HrPortfolio koja prati promjene vrijednosti različitih investicijskih fondova dostupnih na njihovoj platformi. Za koristi ovog rada smo uzeli promjene cijena kao dnevničke zapise koji opisuju poslovno proces. Možda ova poveznica nije očigledna na prvi pogled ali cijena pojedinog fonda nam je izravan indikator rada unutarnjeg poslovnog procesa u jednom fondu.

The screenshot shows the HrPortfolio website interface. At the top, there is a header with contact information and navigation links. Below the header, there is a main navigation bar with categories like 'NASLOVNA', 'OTVORENI FONDOVI', 'MIROVINSKI FONDOVI', 'INVESTICIJSKI RJEČNIK', 'DIONICE', 'VIJESTI', 'INFO', and 'KONTAKT'. The main content area is divided into two columns: 'ULAGAČKI CENTAR' (Investor Center) and 'KORISNICI' (Users). The 'ULAGAČKI CENTAR' section contains four service cards: 'Izdavanje / kupnja udjela u fondu', 'Otkup / prodaja udjela u fondu', 'Zamjena udjela među fondovima', and 'Pitajte nas'. The 'KORISNICI' section includes a login form for 'Joshy' and a 'LOGIN' button, along with utility icons for 'Filter fondova', 'Usporedba fondova', 'Moja lista', and 'Kalkulator imovine'. Below these sections, there is a table titled 'DIONIČKI FONDOVI - EQUITY FUNDS' with columns for 'Fond', 'Datum', 'Cijena', 'Val.', and various performance metrics.

Fond	Datum	Cijena	Val.	+/- %	2023 %	2024 %	1mj %	3mj %	6mj %	12mj %	PGP 3g %	PGP 5g %	Trajanje [god]	Imovina [mil]	Riz.	Info	Kupi	
InterCapital Global Equity - klasa B	06.02.	17,5136	€	0,38	11,97	2,71	2,53	5,47	2,03	10,66	4,23	6,79	21,96	23,69	4	☆	🔍	🛒
ZB trend	06.02.	357,28069	\$	0,12	23,50	5,42	6,99	12,86	12,13	21,17	9,71	12,32	21,29	39,05	5	☆	🔍	🛒

Slika 10 HrPortfolio stranica

Na ovoj stranici postoji resurs „prosirena-tecajnica“ koji nam prikazuje šest različitih tablica koja svaka sadrži dnevne cijene i promjene cijena za svoju vrstu fondova.

U skupu ovog rada ćemo analizirati ovih 6 vrsti fondova:

- Dionički fondovi
- Mješoviti fondovi
- Obveznički fondovi
- Kratkoročni obveznički fondovi
- Novčani fondovi
- Posebni fondovi

Korisno je što su nam stupci između različitih vrsta fondova standardizirani što nam omogućava korištenje istog programa za formatiranje za sve slučajeve. Problem se događa kod toga da izravnim preuzimanjem stranice pomoću GET naredbe ne dobijemo podatke jer se oni dinamički dohvaćaju nakon početnog dohvaćanja stranice.

Jedno od mogućih rješenja je ručno preuzimanje podataka svaki dan i spremanje u bazu za kasnije parsiranje ali to lako dovodi do nedostajanja podataka jer dolazi do mogućnosti da ta osoba zaboravi. Želimo rješenje koje se može aktivirati automatski i konzistentno svaki dan preuzeti najnovije podatke bez intervencije korisnika.

Automatsko rješenje je napravljeno korištenjem alata Selenium koji nam je omogućio simuliranje Internet preglednika i dohvaćanje dinamičkih vrijednosti. Automatsko pokretanje skripte svaki dan je napravljeno dodavanjem Cron zadatka. To smo postigli slijedećim postupkom.

```
crontab -e
## U editoru dodamo ovu liniju
0 21 * * 1-5 /mnt/96F8FD80F8FD5F45/Diplomski/kod/scraper_runner.sh
```

Kod 15 Kod za dodavanje Cron zadatka

Ova druga naredba se sastoji od 5 brojeva i konačne skripte koju želimo pokrenuti.

1	2	3	4	5	6
Minute	Sat	Dan	Mjesec	Dan u tjednu	Skripta
0-59	0-23	1-31	1-12	0-6	String

Tablica 1 Argumenti cron zadatka

Po ovome naša se skripta pokreće svaki radni dan u tjednu (ponedjeljak-petak) u 21:00. Kao što se može primijetiti ne pokreće se izravni python skripta nego Shell pokretač.

Ovo je tako strukturirano jer se za python skriptu koristi virtualno okruženje radi očuvanja izoliranosti aplikacije. U ovakvoj strukturi dodavanje, brisanje ili ažuriranje paketa na sistemskoj Python instalaciji neće utjecati na naše dohvaćanje podataka.

```
##Aktiviranje virtualnog okruženja
source env/bin/activate
./scraper.py
```

Kod 16 Bash skripta za pokretanje

U samoj python skripti smo morali napraviti početnu inicijalizaciju našeg Selenium Internet preglednika.

```
options = webdriver.ChromeOptions()
options.add_argument('--no-sandbox')
options.add_argument("--headless")
options.add_argument('--disable-dev-shm-usage')
driver = webdriver.Chrome(service=ChromeService(
    ChromeDriverManager().install()), options=options)
```

Kod 17 Inicijalizacija WebDriver-a

Tu vidimo da su dodane postavke za Internet preglednik. Postavke „no-sandbox“ i „disable-dev-shm-usage“ su tu radi kompatibilnosti i stabilnosti sustava te nije potrebno poblize ulaziti u njih. Postavka „headless“ nam omogućuje korištenje ove skripte u pozadini na serveru jer bi se inače na svaki zahtjev otvorio prozor Google Chrome preglednika. Ova postavka omogućuje da se simulira preglednik bez aktiviranja grafičkog sučelja.

```
url = 'https://hrportfolio.hr/prosirena-tecajnica#dionicki-fondovi'
driver.get(url)
df_list = pd.read_html(driver.page_source,thousands=".", decimal=",")
```

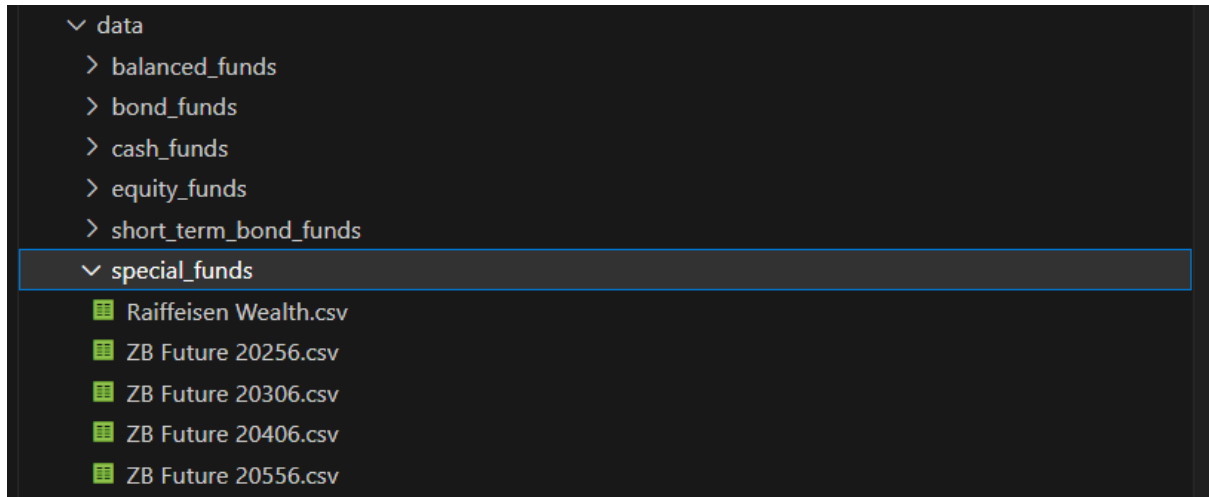
Kod 18 Dohvaćanje podataka

U ovom odsječku koda vidimo kako se koristi Selenium preglednik za dohvaćanje podataka sa web stranice. Potom korištenjem biblioteke Pandas pronalazimo sve moguće tablice u izvornom html kodu koji smo dohvatili.

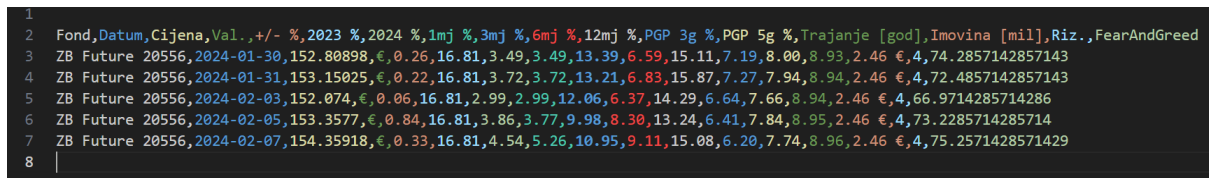
```
folder_mapping=["equity_funds", 'balanced_funds', 'bond_funds', 'short_term_bond_funds', 'cash_funds', 'special_funds']
for folder in folder_mapping:
    if not os.path.isdir(f'data/{folder}'):
        os.mkdir(f'data/{folder}')
for index,df in enumerate(df_list[2:8]):
    for j,x in df.iterrows():
        clear_entry(x)
        for key in x.keys():
            temp= str(x[key]).split(' ')
            if '%' in key and len(temp)>1:
                x[key]= temp[1]
x["FearAndGreed"]= fear_and_grid['fear_and_greed']["score"]
x["Datum"]= date.today().isoformat()
filepath= f'data/{folder_mapping[index]}/{x["Fond"]}.csv'
header_flag= not os.path.isfile(filepath)
with open(filepath, "a+", newline='') as file:
    if header_flag:
        print("Header")
        file.write(','.join(x.keys())[0:] + "\n")
    temp=list(x.values)
    converted= map(lambda x: str(x).replace(',','.'),temp)
    # print("to write:",','.join(list(converted))+ "\n")
    file.write(','.join(list(converted)) + "\n")
    file.flush()
    print("Written:")
```

Kod 19 Zapisivanje današnjih vrijednosti u datoteke

U ovom dijelu koda možemo vidjeti da se za svaku vrstu fondova stvara novi poddirektorij u direktoriju /data/ te da se u njemu za svaki fond stvara odvojeni CSV dokument u koji se nadodaje novi zapis. Ovaj pristup nam omogućuje veću fleksibilnost kod kasnijeg izbora kako želimo strukturirati dnevničku datoteku jer ne radimo nikakvo filtriranje nego samo izravno za svaki fond zapisujemo podatke i svi su potom dostupni.



Slika 11 Konačna struktura direktorija



Slika 12 Primjer izlazne CSV datoteke

5.2. Pretvorba u dnevnički zapis

Kao što smo prije spomenuli u poglavlju 2 iako postoje razne implementacije automatske pretvorbe CSV zapisa u XES ne omogućavaju nam fleksibilnost potrebnu za izvršavanje ovog zadatka. Naša jednostavna implementacija pretvarača napisana je u Python programskom jeziku. Ova implementacija nije zamišljena za opću upotrebu nego je specifično napravljena za trenutni skup podataka na kojem radimo. Iako je pretvarač zamišljen specifično za ovu svrhu i dalje je moguće preko komandne linije mu predati listu klasifikatora aktivnosti. To se može postići nabranjem polja odvojenih zarezom pri pokretanju.

```
python3 converter.py lifecycle:transition,concept:name
```

Kod 20 Primjer poziva

Fleksibilnost koja nam je bila potrebna je nadodati određena stanja ovisno o promjeni vrijednosti fonda. Ova stanja su spremljena u „lifecycle:transition“ i „concept:name“ attribute.

```
def get_eventEnum(change):
    parsed= float(change)
    if parsed == 0:
        return "NO_CHANGE"
    if parsed > 0:
        if abs(parsed)<1:
            return "UP"
        else:
            return "UP_BIG"
    else:
        if abs(parsed)<1:
            return "DOWN"
        else:
            return "DOWN_BIG"
```

Kod 21 Kod za odabir stanja

Samo formatiranje zapisa za zapis u XES kompatibilnom formatu je zamišljeno korištenjem lambda funkcija sa ulazima za ključ i vrijednost.

```
6 CONVERTER_VERSION= '0.3_JD'|
7 string_xes= lambda key,value : f'<string key="{key}" value="{value}"/>\n'
8 float_xes= lambda key,value : f'<float key="{key}" value="{value}"/>\n'
9 int_xes= lambda key,value: f'<int key="{key}" value="{value}"/>'
10 trace_xes= lambda children : f'<trace>\n{children}</trace>\n'
11 event_xes= lambda children : f'<event>\n{children}</event>\n'
12 date_xes= lambda key,value : f'<date key="{key}" value="{value}"/>\n'
13 log_xes= lambda content : f'<log xes.version="2.0" openxes.version="{CONVERTER_VERSION}" xm
14
```

Kod 22 Lambda funkcije za zapis vrijednosti

Program pretvarač prolazi kroz svaku CSV datoteku koja se nalazi u data direktoriju te ih obrađuje. Za svaku vrstu fonda generira jednu datoteku u kojoj je svaki <trace> jedan fond koji pripada toj vrsti. Osim toga generira i jedan zajedničku datoteku koja se sastoji od svih fondova neovisno o njihovoj vrsti.

```
37 accumulated_file= open("../converted/accumulated.xes","w+")
38 for dirpath, dirnames, filenames in os.walk("../data/"):
39     if(len(dirnames)!=0):
40         continue
41     single_file= open(f'../converted/{dirpath.replace("../data/","")}.xes','w+')
42     single_content=''
43     for filename in [f for f in filenames if f.endswith(".csv")]:
44         with open(os.path.join(dirpath, filename), "r", newline='') as file:
45             file_lines=file.readlines()
46             trace_content=''
47             trace_content+= string_xes("concept:name",file_lines[1].split(",")[0])
48             for line_index,line in enumerate(file_lines[1:]):
49                 event_content=''
50                 values = line.split(',')
51                 values= [x.strip() for x in values]
52                 event_content+= string_xes("lifecycle:transition","INITIAL" if line_index== 0 else get_eventEnum(values[4]))
53                 event_content+= string_xes("concept:name","INITIAL" if line_index== 0 else get_eventEnum(values[4]))
54                 event_content+= float_xes("change", values[4])
55                 event_content+= float_xes("value", values[2])
56                 event_content+= float_xes("fearAndGreed", values[-1])
57                 event_content+= date_xes("time:timestamp", f'{values[1]}T15:02:39.448+01:00')
58                 event_content+= int_xes('@@index',index)
59                 trace_content+= event_xes(event_content)
60                 index+=1
61             single_content+=trace_xes(trace_content)
62             output+= trace_xes(trace_content)
63     single_file.write(log_xes(single_content))
64     single_file.close()
65     accumulated_file.write(log_xes(output))
66     accumulated_file.close()
67     # print(log_xes(output))
```

Kod 23 Kod za pretvaranje CSV zapisa u XES datoteke

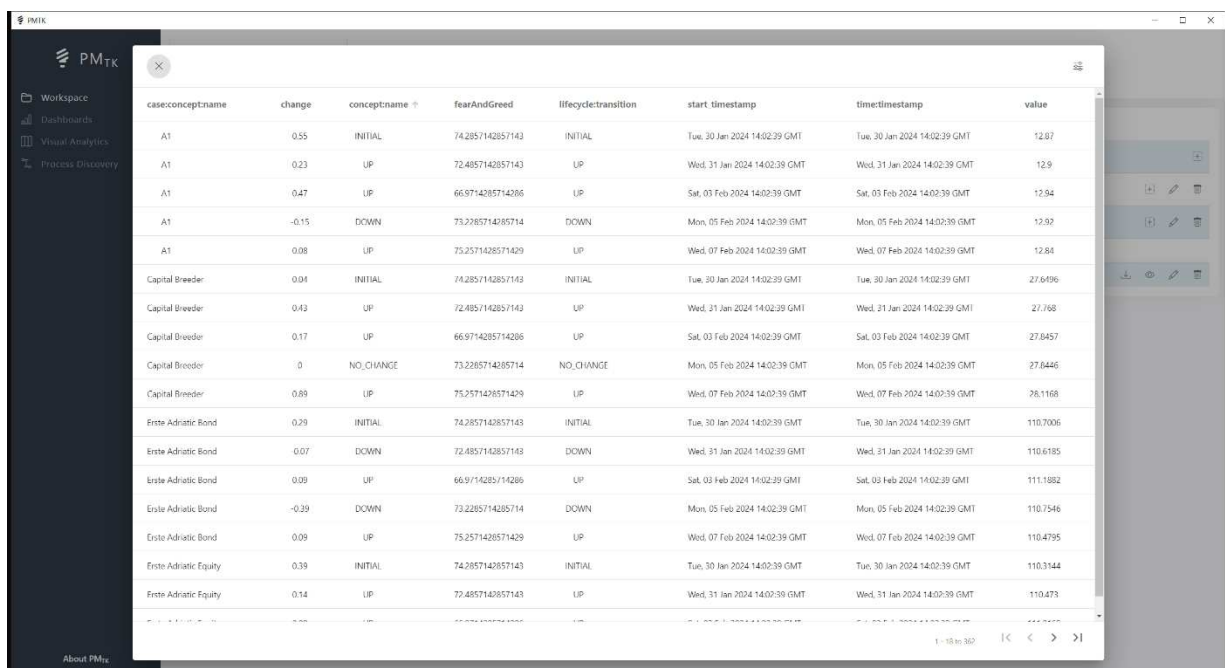
```
kod > converted > balanced_funds.xml > log > trace > event
1 <log xes.version="2.0" converter.version="0.3_JD" xmlns="http://www.xes-standard.org/">
2   <classifier name="Activity classifier" keys="lifecycle:transition" />
3   <string key="origin" value="csv" />
4   <trace>
5     <string key="concept:name" value="Erste Adriatic Multi Asset2" />
6     <event>
7       <string key="lifecycle:transition" value="INITIAL" />
8       <string key="concept:name" value="INITIAL" />
9       <float key="change" value="0.32" />
10      <float key="value" value="94.364" />
11      <float key="fearAndGreed" value="74.2857142857143" />
12      <date key="time:timestamp" value="2024-01-30T15:02:39.448+01:00" />
13      <int key="@@index" value="0" />
14    </event>
15    <event>
16      <string key="lifecycle:transition" value="UP" />
17      <string key="concept:name" value="UP" />
18      <float key="change" value="0.02" />
19      <float key="value" value="94.3838" />
20      <float key="fearAndGreed" value="72.4857142857143" />
21      <date key="time:timestamp" value="2024-01-31T15:02:39.448+01:00" />
22      <int key="@@index" value="1" />
23    </event>
24    <event>
25      <string key="lifecycle:transition" value="UP" />
26      <string key="concept:name" value="UP" />
27      <float key="change" value="0.11" />
28      <float key="value" value="94.6015" />
29      <float key="fearAndGreed" value="66.9714285714286" />
30      <date key="time:timestamp" value="2024-02-03T15:02:39.448+01:00" />
31      <int key="@@index" value="2" />
32    </event>
33    <event>
34      <string key="lifecycle:transition" value="DOWN" />
35      <string key="concept:name" value="DOWN" />
36      <float key="change" value="-0.07" />
37      <float key="value" value="94.5382" />
38      <float key="fearAndGreed" value="73.2285714285714" />
39      <date key="time:timestamp" value="2024-02-05T15:02:39.448+01:00" />
40      <int key="@@index" value="3" />
41    </event>
```

Kod 24 Primjer izlazne datoteke

6. Analiza dobivenih rješenja

6.1. Provjera ispravnosti

Sada kada smo dobili naše dnevničke datoteke u potrebno XES formatu prvo je potrebno provjeriti da li su uopće kompatibilni sa postojećim alatima za analizu. Za ovaj test ćemo akumuliranu datoteku sa zapisima iz svih vrsta fondova učitati u program PMTK.



case/conceptname	change	conceptname #	fearAndGreed	lifecycle/transition	start timestamp	timestamp	value
A1	0.55	INITIAL	74.2857142857143	INITIAL	Tue, 30 Jan 2024 14:02:39 GMT	Tue, 30 Jan 2024 14:02:39 GMT	12.87
A1	0.23	UP	72.4857142857143	UP	Wed, 31 Jan 2024 14:02:39 GMT	Wed, 31 Jan 2024 14:02:39 GMT	12.9
A1	0.47	UP	66.9714285714286	UP	Sat, 03 Feb 2024 14:02:39 GMT	Sat, 03 Feb 2024 14:02:39 GMT	12.94
A1	-0.15	DOWN	73.2857142857143	DOWN	Mon, 05 Feb 2024 14:02:39 GMT	Mon, 05 Feb 2024 14:02:39 GMT	12.92
A1	0.08	UP	75.2571428571429	UP	Wed, 07 Feb 2024 14:02:39 GMT	Wed, 07 Feb 2024 14:02:39 GMT	12.84
Capital Breeder	0.04	INITIAL	74.2857142857143	INITIAL	Tue, 30 Jan 2024 14:02:39 GMT	Tue, 30 Jan 2024 14:02:39 GMT	27.6496
Capital Breeder	0.43	UP	72.4857142857143	UP	Wed, 31 Jan 2024 14:02:39 GMT	Wed, 31 Jan 2024 14:02:39 GMT	27.768
Capital Breeder	0.17	UP	66.9714285714286	UP	Sat, 03 Feb 2024 14:02:39 GMT	Sat, 03 Feb 2024 14:02:39 GMT	27.8457
Capital Breeder	0	NO_CHANGE	73.2857142857143	NO_CHANGE	Mon, 05 Feb 2024 14:02:39 GMT	Mon, 05 Feb 2024 14:02:39 GMT	27.8446
Capital Breeder	0.69	UP	75.2571428571429	UP	Wed, 07 Feb 2024 14:02:39 GMT	Wed, 07 Feb 2024 14:02:39 GMT	28.1168
Erste Adriatic Bond	0.29	INITIAL	74.2857142857143	INITIAL	Tue, 30 Jan 2024 14:02:39 GMT	Tue, 30 Jan 2024 14:02:39 GMT	110.7006
Erste Adriatic Bond	0.07	DOWN	72.4857142857143	DOWN	Wed, 31 Jan 2024 14:02:39 GMT	Wed, 31 Jan 2024 14:02:39 GMT	110.6185
Erste Adriatic Bond	0.09	UP	66.9714285714286	UP	Sat, 03 Feb 2024 14:02:39 GMT	Sat, 03 Feb 2024 14:02:39 GMT	111.1882
Erste Adriatic Bond	-0.39	DOWN	73.2857142857143	DOWN	Mon, 05 Feb 2024 14:02:39 GMT	Mon, 05 Feb 2024 14:02:39 GMT	110.7546
Erste Adriatic Bond	0.09	UP	75.2571428571429	UP	Wed, 07 Feb 2024 14:02:39 GMT	Wed, 07 Feb 2024 14:02:39 GMT	110.4795
Erste Adriatic Equity	0.39	INITIAL	74.2857142857143	INITIAL	Tue, 30 Jan 2024 14:02:39 GMT	Tue, 30 Jan 2024 14:02:39 GMT	110.3144
Erste Adriatic Equity	0.14	UP	72.4857142857143	UP	Wed, 31 Jan 2024 14:02:39 GMT	Wed, 31 Jan 2024 14:02:39 GMT	110.473

Slika 13 Rezultat učitavanja datoteke

Na prijašnjoj slici je vidljivo da je program uspješno učitao našu datoteku što znači da zadovoljava XES standard i može se analizirati. U daljnjim poglavljima ćemo analizirati procese kroz različite vizualizacije i analitike dostupne iz naših alata za dubinsku analizu procesa.

6.2. Analiza jednog procesa

Za analizu koristimo akumulirani podatkovni zapis zato što nam daje najširu sliku stanja tržišta investicijskih fondova. U ovom slučaju modeliramo sve dostupne investicijske instrumente koje sadrže naši fondovi.

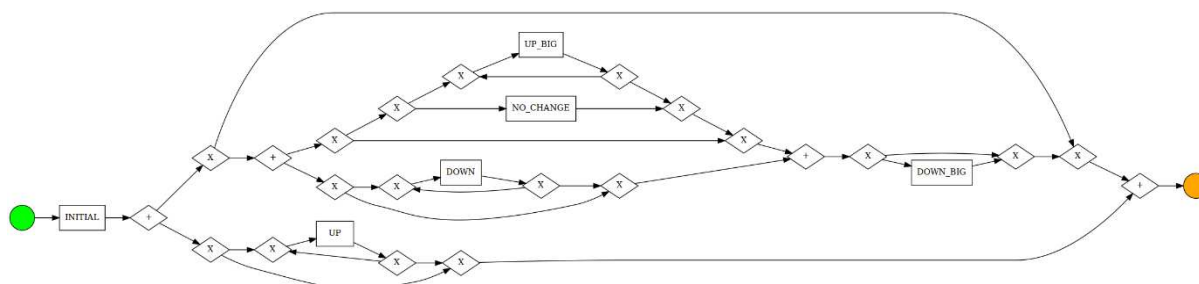
Za ovaj korak koristimo PM4PY biblioteku u pythonu za prikaz BPMN dijagrama, DFG dijagram te Heuristički dijagram.

```

kod > visualize.py > ...
1  import pm4py
2  import sys
3  if __name__ == "__main__":
4      print("args", sys.argv)
5      log = pm4py.read_xes(sys.argv[1])
6      dfg, start_activities, end_activities = pm4py.discover_dfg(log)
7      pm4py.view_dfg(dfg, start_activities, end_activities)
8      process_model = pm4py.discover_bpmn_inductive(log)
9      pm4py.view_bpmn(process_model)
10     map = pm4py.discover_heuristics_net(log)
11     pm4py.view_heuristics_net(map)
12

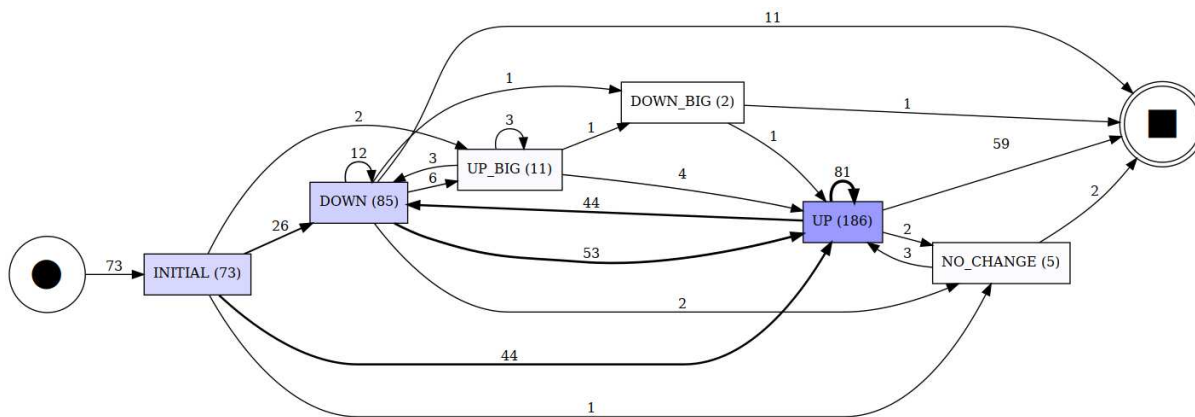
```

Kod 25 Kod za prikaz dijagrama



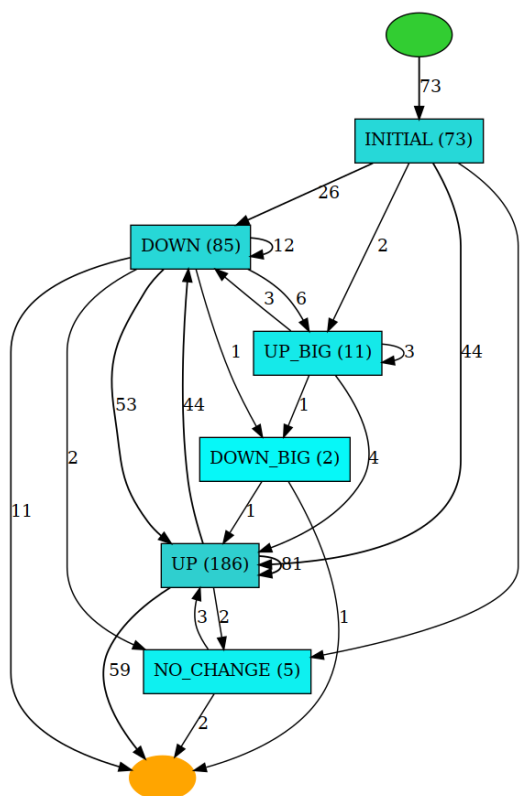
Slika 14 BPMN dijagram

BPMN (Business Process Modeling and Notation) dijagram nam vizualizira kretanje određenog poslovnog procesa od samoga početka do konačnoga stanja. U našem slučaju te počinje od početnog stanja „INITIAL“ te prolazi kroz različite moguće kombinacije ostalih stanja sve dok ne dođe do kraja.



Slika 15 DFG dijagram

DFG (Data-Flow Graph) nam prikazuje ovisnost podataka između različitih operacija. U ovom slučaju vidimo učestalost prelazaka iz jednog mogućeg stanja u drugo. Ovo je vidljivo pomoću brojeva koji govore koliko puta su se dogodili prijelazi između ta dva stanja u našoj dnevničkoj datoteci.



Slika 16 Heuristički dijagram

Zbog nedostatka matematičkih operacija u našim podacima Heuristički dijagram nam pokazuje gotovo iste informacije kao i DFG. Iako su grafovi slični po sadržaju heuristički dijagram ima više smisla u ovom slučaju jer algoritam nije fiksni te možemo samo procijeniti iduće stanje.

7. Moguće nadogradnje

Ovaj projekt se bavio samim dobavljanjem specifičnih informacija sa interneta te njihovo formatiranje za moguću obradu u alatima za dubinsku analizu procesa što je i postignuti, ali kao i uvijek postoje moguće nadogradnje. Vrste nadogradnji ćemo spomenuti u slijedećim pod poglavljima.

7.1. Nadogradnja podataka

U ovome projektu smo za pregled tržišta koristili zapise dostupne na HrPortfolio stranici zajedno sa ukupnim „Fear And Greed“ indeksom sa CNN stranice. Ovo nam je dalo prilično dobar pregled stanja tržišta ali bi se i dalje mogao dobiti precizniji pregled tržišta većom količinom podataka. Ovo se može postići praćenjem više izvora podataka radi pristupa više investicijskih instrumenata ili kroz duži vremenski period. Pošto se tržište većinom kreće u „valovima“ di raste kroz neko vrijeme i onda neko vrijeme pada ako koristimo podatke samo iz jednog od tih perioda naš pogled je subjektivan. Dugoročno praćenje bi smanjilo ovu subjektivnost.

7.2. Nadogradnja pretvarača

Naš pretvarač iz CSV u XES format je specijaliziran samo za podatke s kojima smo mi radili. Moguća nadogradnja u ovome slučaju bi bila napraviti pretvarač što generalnijim da ga se može koristiti za što više različitih CSV zapisa. Ovo bi mogli postići dodavanjem opcija koje se mogu predati našem programu preko terminala koji bi mu mogli reći kako odabrati i formatirati polja. Ovaj pristup proširenja funkcionalnosti kroz terminal bi mogao dovesti do velike kompleksnosti pozivanja programa što bi moglo biti riješeno dodavanjem grafičkog sučelja.

7.3. Nadogradnja analize

Cilj ovog rada je bio pokazati način prikupljanja dnevničkih zapisa iz dinamičkih izvora sa interneta te njihovo strukturiranje za obradu. Nažalost nije bilo u opsegu ovog rada i njihova dubinska analiza. Analize spomenute u ovom radu su bile radi validacije naših dnevničkih zapisa te prikaz da su naše dobivene datoteke spremne za daljnju obradu i analizu. Pri nadogradnji analize mogli bi proširiti više testova sa različitim pod skupovima podataka. Kako se određeni fondovi ponašaju kroz jedan tjedan ili na mjesečnoj bazi. Ova proširena testiranja bi nam mogla donijeti zanimljiv uvid u procese koji bi mogli koristiti za pomoć u donošenju odluka pri investiranju.

Zaključak

Ovaj rad prolazi kroz moguće rješenje pronalaženja izvora dnevničkih zapisa slobodno dostupnog na internetu te formatiranje tih zapisa za daljnju analizu. Cilj ovog rada je postignut korištenjem nabrojanih tehnologija koje su odabrane iz razloga koji su navedeni u specifičnim poglavljima. Programi za dohvaćanje i obradu ovih podataka su napisani u programskom jeziku Python te je dio za dohvaćanje podataka dizajniran za dugoročno skupljanje podataka bez ljudske intervencije. Samim time se ovaj rad može pokrenuti na serveru i nastaviti koristiti za dobavljanje dnevničkih zapisa i nakon završetka ovog rada. Najveći problem ove implementacije rješenja je to što naš pretvarač podataka u XES format specijaliziran i nije iskoristiv za druge slučajeve. Rad bi se mogao nastaviti implementacijom rješenja spomenutih u poglavlju 7.2 kako bi se mogao ostvariti program koji bi se na jednostavan konzistentan način mogao koristiti na početku dubinske analize procesa. Smatram da bi ovakav alat pozitivno doprinio lakšem početku rada ljudima koji tek započinju radom u dubinskoj analizi procesa.

Literatura

- [1] Process and Data Science Group - RWTH Aachen University. Poveznica: <https://processmining.org/>; pristupljeno: 13.1.2024
- [2] Eric Verbeek, Wil van der Aalst. XES standard. Poveznica: <https://www.xes-standard.org/start>; pristupljeno: 13.1.2024
- [3] EEE Task Force on Process Mining. Resources. Poveznica: <https://www.tf-pm.org/> pristupljeno 13.1.2024
- [4] Josip Delač. Strukturiranje dnevnčkih datoteka sa burzovnim indeksima. Seminarski rad. Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva. 2021
- [5] Wil van der Aalst. Process Mining Book. Poveznica: <https://fluxicon.com/book/read/> pristupljeno 15.1.2024
- [6] Selenium. Documentation. Poveznica: <https://www.selenium.dev/documentation/> pristupljeno 15.1.2024
- [7] Process Mining Group of the Fraunhofer Institute for Applied Information Technology. PM4PY. Poveznica: <https://pm4py.fit.fraunhofer.de/>; pristupljeno: 13.1.2024
- [8] Wil van der Aalst, Josep Carmona. Process Mining Handbook. Springer 2022
- [9] Thomas H. Davenport, Andrew Spanyi, What Process Mining Is, and Why Companies Should Do It. Poveznica: <https://hbr.org/2019/04/what-process-mining-is-and-why-companies-should-do-it>; pristupljeno 16.1.2024
- [10] Altexsoft. Process Mining Explained: Techniques, Applications, and Challenges . Poveznica: <https://www.altexsoft.com/blog/process-mining/>; pristupljeno 15.1.2024

Sažetak

Prikupljanje nestrukturiranih dnevničkih zapisa za otkrivanje poslovnih procesa

Ovaj diplomski rad se bavi implementacijom automatskog dohvaćanja podataka sa dinamičkog izvora na internetu te njihovim formatiranjem za dubinsku analizu procesa. U ovome radu opisane su i tehnologije i razlozi njihovog korištenja u radu. Implementacija je opisana kroz korake dohvaćanja podataka, strukturiranje dobavljenih podataka u XES format te dodavanje formatiranih podataka u alate za dubinsku analizu radi validacije ispravnosti. Na kraju rada spomenuti su mogući slijedeći koraci za nastavak rada na ovom projektu sa mogućim nadogradnjama.

Ključne riječi : Dubinska analiza procesa, PM4PY, ProM, Selenium, Python, XES format, CSV format, Poslovni process

Summary

Weekly complete event log acquisition for business process discovery

This thesis presents an example implementation of automatic data acquisition from a dynamic internet source and formatting the same data for process mining.

Technologies and reasons for their use in work are also described in this paper. The implementation is described through the steps of acquiring the data, formatting the data, and importing the formatted data into process mining tools for validation. At the end of this paper, there are also possible avenues named for expanding on the work done in it.

Keywords: Process mining, PM4PY, ProM, Python, XES format, CSV format, Financial data, Business process