VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

Fakulta informačních technologií
Faculty of Information Technology

BAKALÁŘSKÁ PRÁCE

BACHELOR‘S THESIS

Brno, 2017 David Mička

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ANALÝZA PODNIKOVÝCH PROCESŮ

BUSINESS PROCESS ANALYSIS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE David Mička
AUTHOR

VEDOUCÍ PRÁCE prof. Ing. Tomáš Hruška CSc.
SUPERVISOR

BRNO 2017

Abstrakt

Cílem práce je návrh způsobu analýzy podnikových procesů se zaměřením na optimalizaci procesů za pomoci prediktivní údržby, realizace prototypu této metody a zhodnocení výsledků a vhodnosti této metody pro použití v praxi. Základem této práce je výzkum process miningu a nástrojů pro jeho zkoumání skupiny Prof.dr.ir. W.M.P. van der Aalst z Technické Univerzity v Eindhovenu. Tato skupina také vytvořila program ProM využit pro tuto práci.

Abstract

The goal of this thesis is to design a method of business process analysis, with the focus on process optimalization by using Predictive Maintenance, creation of a suitable prototype for this method and evaluation of practical applicability. The basis of this thesis is the research of study group led by Prof.dr.ir. W.M.P. van der Aalst revolving around process mining and tools for his examination. The same study group also created the program ProM used in this thesis.

Klíčová slova

Process mining, analýza podnikových procesů, informační systémy, predikce chyb

Keywords

Process mining, business process analysis, information systems, error prediction.

Citace

Mička David: Analýza podnikových procesů, bakalářská práce, Brno, FIT VUT v Brně, 2017

Analýza podnikových procesů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením prof. Ing. Tomáše Hrušky CSc.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
David Mička
17. 5. 2017

Poděkování

Tímto bych chtěl poděkovat panu prof. Ing. Tomáši Hruškovi CSc.

© David Mička, 2017
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

Obsah 1
1 Úvod 2
1.1 Analýza podnikových procesů 2
1.2 Process Mining obecně 3
1.3 ProM 3
2 Často používané modely procesů 4
2.1 Přechodový systém 4
2.2 Petriho sítě 4
2.3 Workflow sítě 5
3 Modelová procesní analýza 6
3.1 Verifikace modelu 6
3.2 Analýza výkonnosti modelu 6
3.3 Omezení modelové analýzy 7
4 Data mining 7
4.1 Učení s učitelem 7
4.1.1 Klasifikace 7
4.1.2 Regrese 8
4.1.3 Rozhodovací stromy 8
4.2 Učení bez učitele 9
4.2.1 Shluková analýza 9
5 Process discovery 11
5.1 Zisk záznamů událostí 11
5.2 Záznamy událostí 11
5.2.1 XES 12
5.3 Úvod do process discovery 12
5.4 Tři pojmy ekvivalence chování 13
5.5 Výzvy objevování procesů 13
5.5.1 Hluk a nekompletnost 13
5.5.2 Křížová validace 14
5.5.3 Čtyři konkurenční kritéria kvality 15
6 Conformance checking 16
7 Analyzovaná data 19
8 Závěr 20

1 Úvod

S rozvíjejícím se průmyslem, pokročilejší automatizací a zvýšeným důrazem na co největší množství informací je kladen čím dál větší důraz na optimalizaci podnikových procesů, ať už se jedná o procesy v rámci firmy, oddělení či o procesy v rámci firemního software. K dispozici je dnes nepřeberné množství záznamů událostí (tzv. event logs) díky kterým je možné podnikové procesy nejen analyzovat, ale poté i revidovat a optimalizovat. Nemusí se nutně jednat o záznamy z výrobních linek, aplikací, či jakéhokoliv software, ale může se například jednat i o záznamy z nemocnic, kdy jednotlivé záznamy tvoří jednotlivá vyšetření, která pacient podstoupil, nebo o záznamy žádostí o půjčky, hypotéky apod.

Z těchto záznamů lze vyčíst například v jakých fázích daného procesu dochází ke zdržením, například kvůli příliš velkému množství požadavků na jednu osobu, zda by nebylo možné daný proces urychlit, umožnit předejití chybám ve výrobní lince, tedy využít data k prediktivní údržbě, nebo jen zjistit jaké akce se v daném procesu nacházejí nejčastěji, tedy pro statistické účely.

Tato Bakalářská práce se bude zabývat popsáním metod dostupných pro analýzu a modelování procesů v informačních systémech, zejména pak metodami process miningu, navrhnutím vhodné struktury dat a realizací zvolené metody a jejím zhodnocením. Již v začátku bych chtěl uvést, že Process mining je relativně novou disciplínou, k níž není k dispozici příliš mnoho zdrojů. Při vypracování této práce jsem tedy využíval převážně zdroje v angličtině, které jsem v případě nenalezení vhodného překladu sám překládal. Některé překlady byly tvořeny tak, aby co nejvíce vystihly původní význam slova, ale v některých případech jsem z důvodu nenalezení vhodného překladu ponechával názvy anglické.

1.1 Analýza podnikových procesů

Cílem analýzy podnikových procesů, či procesů všeobecně, je porozumění a zhodnocení efektivity zavedených firemních procesů za použití metod schopných je analyzovat detailně, na základě dat dostupných ve formě záznamů údalostí získaných zaznamenáváním reálného provozu.

Při analýze procesů se ze zmapovaných procesů vytváří vhodný model. K vytváření modelu lze použít různé notace, ať už BPMN (Business Process Modeling Notation), ARIS, UML (Unified Markup Language) či vývojové diagramy a další.

Pro tuto bakalářskou práci se zaměříme především na metodu process mining.

1.2 Process Mining obecně

Process mining je poměrně mladou disciplínou na pomezí data miningu a strojového učení na jedné straně a modelování procesů a jejich analýzy na straně druhé. Účelem process miningu je objevování, kontrola a zlepšování reálných procesů za pomoci záznamů událostí (event logs). Byť jsou event logy v dnešní době relativně snadno získatelné, ne vždy mají vhodnou strukturu a často jsou rozmístěny v mnoha tabulkách, a tak je třeba surová data upravovat, například sloučením tabulek, odstraněním nepotřebných atributů, či aplikováním určitého pohledu na daná zdrojová data.

Process mining se dělí na tři části. První částí je discovery (objevování). Technika discovery analyzuje vstupní event log za pomoci zkoumání jednotlivých tras. Trasa, anglicky trace sestává z posloupnosti jednotlivých událostí, například *spuštění procesu*, *pozastavení procesu, znovuspuštění procesu, dokončení procesu*. Z analyzování těchto tras vytváří model, který by reálnému modelu měl co nejvíce odpovídat. V této části se pracuje pouze s daty obsaženými v záznamech událostí a není třeba mít o modelovaném systému žádné další informace.

Druhou částí process miningu je conformance neboli zjišťování shody. V této části se vytvořený model srovnává jak s modelem reálným (pokud reálný model máme k dispozici), nebo s jednotlivými trasami a vypočítává se metriky modelu jako například přesnost, vhodnost, obecnost či naopak přílišná přesnost apod. Jednotlivými metrikami se budeme zabývat později. Některé modely sice mohou přesně odpovídat daným trasám, tedy každou sekvenci událostí dané trasy lze v daném modelu provést, ale ne vždy je daný model vhodný, neboť nemusí odpovídat trasám neobsaženým v daném vzorku záznamů událostí.

Třetí částí je enhancement, tedy vylepšování stávajícího modelu. Vylepšováním modelu může být míněno jak upravování modelu vytvořeného v předchozích částech, tak aby více odpovídal reálnému systému, nebo i modifikací reálného systému za účelem jeho zefektivnění. Toho může být docíleno identifikací bottlenecků, rozšířením modelu o timestamp, nebo vytvořením sociální sítě na základě interakcí mezi účastníky daného procesu. Tímto se například dá ze záznamů vyčíst, že osoba *A* se účastní veškerých důležitějších rozhodnutí, tedy zastává určitou manažerskou funkci.

1.3 ProM

Pro analýzu záznamů událostí budu v další části práce používat Open Source program ProM (zkratka pro Process Mining framework) ve verzi 6.6 Home, vytvořený skupinou vedenou Prof.dr.ir. W.M.P. van der Aalst pod Technickou univerzitou v Eindhovenu (TU/e). Program ProM je momentálně asi nejpoužívanějším nástrojem pro process mining v univerzitním prostředí a má rozsáhlou komunitu přispěvatelů, díky nimž jsou na něj vydávány další rozšíření a pluginy.

2 Často používané modely procesů

V dnešní době je majorita modelů procesů stále tvořena ručně na základě lidského pozorování a vstupů. Díky tomuto se často stává, že výsledný model je pouze modelem očekávaného, nebo požadovaného chování, ale častokrát se liší od skutečného modelu, neboť je mnohdy přítomen lidský faktor, neočekávané události, např. chyby, nebo události ovlivňující běžný chod systému/firmy atp.

Analýza vycházející ze záznamů událostí tak může odhalit skryté části modelu, které se mohou od požadovaného chování lišit, ale jsou v procesu běžné. V této kapitole se zaměříme na přechodový systém, Petriho sítě, Workflow sítě, YAWL, Business Process Modeling Notation (BPMN), EPC Notation a kauzální sítě (Causal nets).

2.1 Přechodový systém

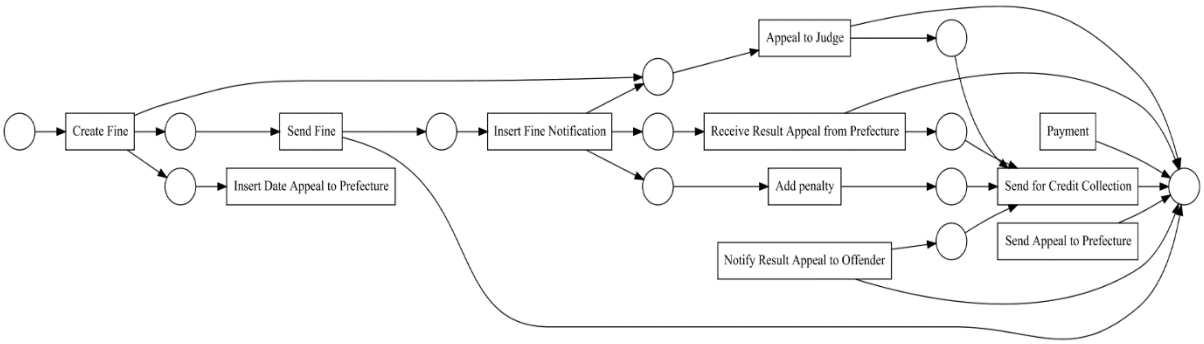
Nejjednodušší modelovací notací je přechodový systém, který jako neorientovaný graf sestává ze stavů a přechodů a jako orientovaný obsahuje ještě popisky přechodů, které pro potřeby tvorby procesních modelů značí názvy daných aktivit. U přechodových systémů není na rozdíl od například Petriho sítí jasné, který přechod se provede, ale znázorňuje všechny možné přechody pro každý stav modelu.

U přechodových systémů je výhodou, že se dají převést do jiných notací jako třeba Petriho sítí, UML apod. tedy jsou vhodné v případě, že nevíme, jakou notaci později budeme vyžadovat.
Problémem přechodových systémů je, že neumí znázorňovat souběžnost transakcí (přechodů) a v případě mnoha souběžných přechodů by bylo velmi výpočetně náročné zjistit, jaká sekvence přechodů by byla použita, neboť by bylo platných příliš mnoho možností. Proto je pro účely process miningu vhodnější používat Petriho sítě.

2.2 Petriho sítě

Petriho sítě jsou jedním z nejpoužívanějších modelovacích jazyků pro znázorňování procesů a systémů. Petriho síť sestává ze stavů, přechodů a tokenů. Aktuální stav dané sítě je znázorněn umístěním tokenu v daném stavu.

Formální popis Petriho sítí pro process mining je trojice $N = (P, T, F)$ kde P je konečná množina stavů, T je konečná množina přechodů a F je konečná množina hran spojujících stavy. Každý přechod může být tzn. odpálen, pokud každý z jeho vstupních stavů obsahuje token. Po odpálení přechodu je ze vstupních stavů token odebrán a je umístěn v každém z výstupních stavů přechodu. Aktuální stav přechodů je značen výčtem stavů, ve kterých se nachází token. Například $[a, c^2, d]$ nám říká, že se token nachází ve stavu a, dva tokeny ve stavu c, a jeden token ve stavu d. Při process miningu jsou nejčastěji využívány Petriho sítě, které jsou *bezpečné*, což znamená, že se v žádném stavu nemůže nacházet více než jeden token.



Obrázek 2.1: Petriho síť analyzovaných dat

2.3 Workflow sítě

Workflow sítě (WF-sítě) jsou typem Petriho sítí, které vždy mají jeden počáteční stav a jeden konečný stav a zároveň pro ně platí, že každý stav se nachází na trase od počátečního stavu ke stavu konečnému. Workflow sítě jsou pro potřeby modelování procesů nejvhodnější, neboť znázorňují životní cyklus každého případu daného typu, jako například případ vyřízení objednávky od počátku až po ukončení objednávky, nebo návštěvy lékaře daného pacienta. Každý model může být instanciován nekonečněkrát, dokud je daný model validní.

Pro využití workflow sítí v process miningu je důležitou vlastností *spolehlivost* (soundness). WF-síť je *spolehlivá*:

- Pokud je *bezpečná* – každý stav obsahuje nejvýše jeden token
- Pokud je pro každé značení dokončitelná, tj. tokeny z každého značení mohou dosáhnout koncového stavu
- Pokud obsahuje pouze proveditelné přechody
- Pokud je koncový stav jediným dosažitelným značením obsahujícím token

3 Modelová procesní analýza

V této kapitole v krátkosti popíšu základní způsoby analýzy modelů: *verifikaci* a *analýzu výkonnosti*.

3.1 Verifikace modelu

Jak již bylo zmíněno v předchozí části práce, jednou z metrik kvality modelu je *spolehlivost* modelu. Spolehlivost modelu je považována za jednu z vlastností modelu pro jeho verifikaci. Problémem spolehlivosti je, že je to vlastnost příliš obecná. Další z možností, jak verifikovat model je pomocí *Lineární temporální logiky* (LTL). LTL pomocí temporálních operátorů jako například □ značící vždy, nebo operátor ◊ značící časem apod. může kontrolovat platnost modelu za pomocí pravidel jako například $\Box (a \Rightarrow \Diamond b)$ značí, že po jakémkoli výskytu události *a* musí následovat výskyt události *b*. Některé nástroje pro kontrolu verifikace modelu jsou zahrnuty v jazyce YAWL, který by například v případě výskytu čekajícího tokenu v modelu i po jeho ukončení hlásil problém.

3.2 Analýza výkonnosti modelu

Základními metrikami analýzy výkonnosti jsou *čas*, *cena* a *kvalita*. Pro každou z těchto hodnot mohou být nastaveny takzvané KPI (Key Performance Indicator).
Pro *čas* existují následující KPI (jednotlivým KPI je ponechán anglický název z důvodu zachování významu, neboť překlad by byl v tomto případě nevhodný):

- *Flow time* – celkový čas od vytvoření případu po jeho dokončení. V rámci WF-sítí by to byla doba od obsazení počátečního stavu, po dosažení koncového stavu. Na tuto hodnotu se nastavuje tzv. *service level* – procento případů s *flow time* menším než předem definovaná hodnota.
- *Service time* – čas strávený prací na případu. Měří se buď pro jednotlivé události, nebo pro celý případ. V případě aktivit může být service time delší než flow time (případ souběžnosti událostí a jejich součet), nebo naopak pouhým zlomkem flow time (průměrný service time jednotlivých událostí).
- *Waiting time* – čas po který případ čeká na uvolnění potřebných zdrojů. Opět je možné ho měřit pro jednotlivé události, nebo pro celý případ. Příkladem je pacient čekající na rentgen. Pro měření výkonnosti je opět možné určit určitý service level.
- *Synchronization time* – čas po který není událost ještě aktivována z důvodu čekání na token, tedy na synchronizaci vstupních tokenů (při více než jednom vstupním stavu události).

V případě hodnocení *ceny* může být na věc pohlíženo z mnoha stran. Jedním z pohledů je cena zdrojů potřebných pro provedení události jako například cena provedení jednoho rentgenu, nebo i z hlediska času a zdroje (mzda pracovníka obsluhujícího rentgen po dobu snímání).

U hodnocení *kvality* modelu lze získávat informace formou dotazníků pro zákazníky, množstvím stížností, či náhledem, zda model dlouhodobě splňuje časová KPI.

3.3 Omezení modelové analýzy

Základním předpokladem pro hodnocení výkonnosti modelu, či jeho verifikace je přítomnost kvalitního modelu. Pokud by model byl moc obecný a nekoreloval se skutečným modelem, byla by tato hodnocení zbytečná. Stejně tak platí pro modely popisující požadované, nikoliv reálné chování modelu. Na tento problém cílí právě process mining, díky němuž je možné problém nekvalitního modelu vyřešit.

4 Data mining

Tato kapitola se bude zabývat přehledem základních technik data miningu.

4.1 Učení s učitelem

Učení s učitelem má za předpoklad označená data. Z těchto dat je vybrána jedna proměnná jako tzv. *response variable*, nebo také *závislá proměnná, vysvětlovaná proměnná* a ostatní proměnné jsou označeny jako *predictor variables* neboli *nezávislé proměnné, vysvětlující proměnné*. Cílem je vysvětlit závislou proměnou ve vztahu na nezávislé proměnné, tedy zjistit možnou kauzalitu mezi hodnotami nezávislých proměnných a hodnoty proměnné závislé. Učení s učitelem lze dále dělit na *klasifikaci* a *regresi* v závislosti na typu závislé proměnné. V případě, že závislá proměnná nabývá číselných hodnot, jedná se o *regresi*, v opačném případě nabývá *kategorických* hodnot a jedná se o *klasifikaci*.

4.1.1 Klasifikace

Klasifikace má, jak již název napovídá, za cíl rozdělit dat do tříd (classes) v závislosti na hodnotě závislé proměnné. Příkladem rozdělení může být třeba rozdělení skupiny dotazovaných lidí na sportovce a nesportovce, na vegetariány, vegany a všežravce apod. Poté by bylo možné zjišťovat kupříkladu závislost hodnoty sportovec, či nesportovec na typu stravy, či konzumaci alkoholu.

V některých případech je možné převést číselnou hodnotu na hodnotu kategorickou. Hodnoty váhy pod 80 kg tak mohou být označený za lehké a nad 80 kg za těžké. Poté je opět možno srovnávat závislost váhy na tom, zdali člověk sportuje, či nikoliv, popřípadě i na typu stravy. Hodnoty nula a jedna mohou být také mapovány na hodnoty *true* a *false*, nebo školní známky na slovní ohodnocení.

4.1.2 Regrese

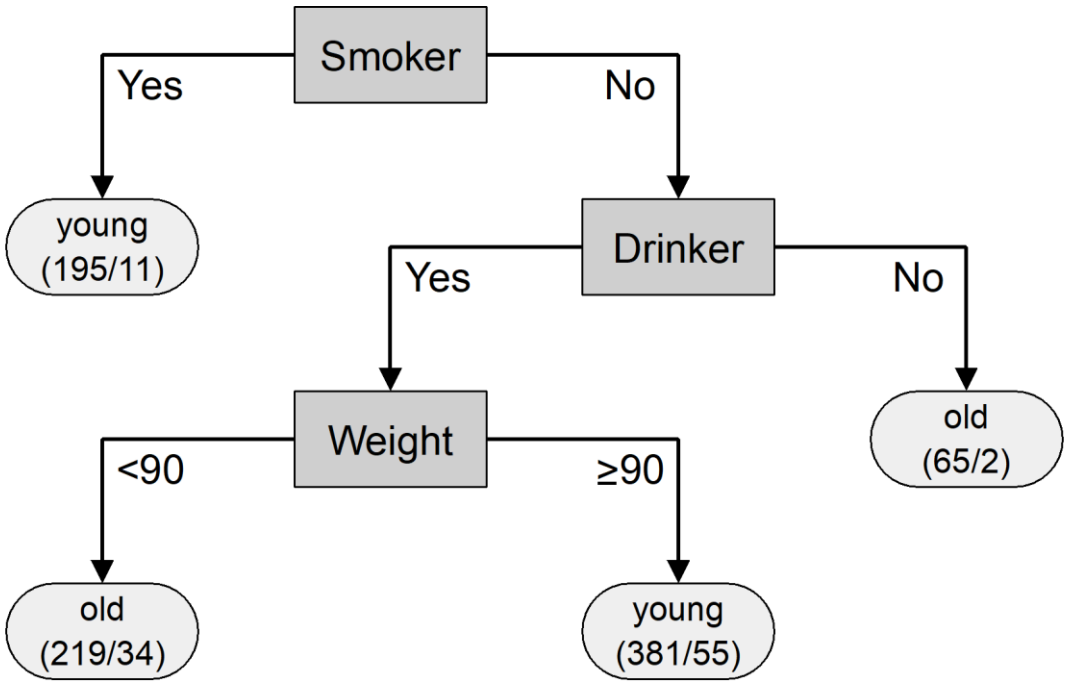
Při regresi lze jako závislou proměnnou zvolit například školní známku a zjišťovat závislost známky na chování a docházce a docházku tak třeba predikovat pro nové studenty z výpisu předchozích známek.

Drinker	Smoker	Weight	Age
Yes	Yes	120	44
No	No	70	96
Yes	No	72	88
Yes	Yes	55	52
No	Yes	94	56
No	No	62	93
…	…	…	…

Tabulka 4.1: Vzorek z dat o zesnulých osobách určený ke studiu vlivů konzumace alkoholu, kouření a tělesné váhy na délce života. Vzorek dat je zachován stejný jako ve výše zmíněné knize.

4.1.3 Rozhodovací stromy

Tato metoda učení s učitelem cílí na klasifikaci instancí na základě prediktorových proměnných. K dispozici je jedna kategorická *vysvětlované proměnné* označující data a výsledek je uspořádán ve formě stromu. Koncové uzly odpovídají možným hodnotám *vysvětlované proměnné*. Ostatní uzly odpovídají prediktorovým proměnným. V kontextu učení rozhodovacího stromu jsou prediktorové proměnné označovány jako atributy. Každý atribut rozdělí sadu instancí na dvě nebo více podmnožin. Kořenový uzel odpovídá všem instancím. Příklad rozhodovacích stromů si ukážeme na tabulce 4.1 převzaté z knihy Process mining od W. M. P. van der Aalst.



Obrázek 4.1: Rozhodovací strom vytvořený podle tabulky 4.1. 860 zesnulých lidí je klasifikováno mezi "mladé" a "staré" (zemřeli před 70 lety života, či po 70 letech života).

V obrázku 4.1 kořenový uzel zastupuje všechny případy, v tomto případě 860 osob. Na základě atributu *kuřák* jsou tyto případy rozděleny na ty, které kouří (195 osob) a ty, které nekouří (665 osob). Kuřáci nejsou dále rozděleni. Na základě těchto informací jsou tyto instance již označeny jako "mladí", tj. u kuřáků se očekává, že zemřou před dosažením věku 70 let. Nekuřáci jsou rozděleni na „pijáky“ a „nepijáky“. Druhá skupina lidí by měla žít dlouho a je označena jako "staří". Všechny uzly stromu mají dvě čísla. První číslo udává počet případů klasifikovaných jako patřících do dané skupiny (kuřáci, pijáci, …). Druhé číslo udává počet případů odpovídajících danému uzlu, ale špatně klasifikovaných. Z 195 kuřáků, kteří byli klasifikováni jako "mladí" bylo 11 lidí špatně klasifikovaných, tzn. nezemřelo dříve než v 70 letech i přesto že kouřili.

Při tvorbě rozhodovacích stromů hraje velkou roli vlastnost *entropie*. Entropie značí, kolik bitů je potřeba pro zakódování daného elementu. Rozdělování uzlů na další větve vždy sníží celkovou entropii, tedy v extrémním případě rozdělení na jednotlivé hodnoty by byla entropie rovna nule.

4.2 Učení bez učitele

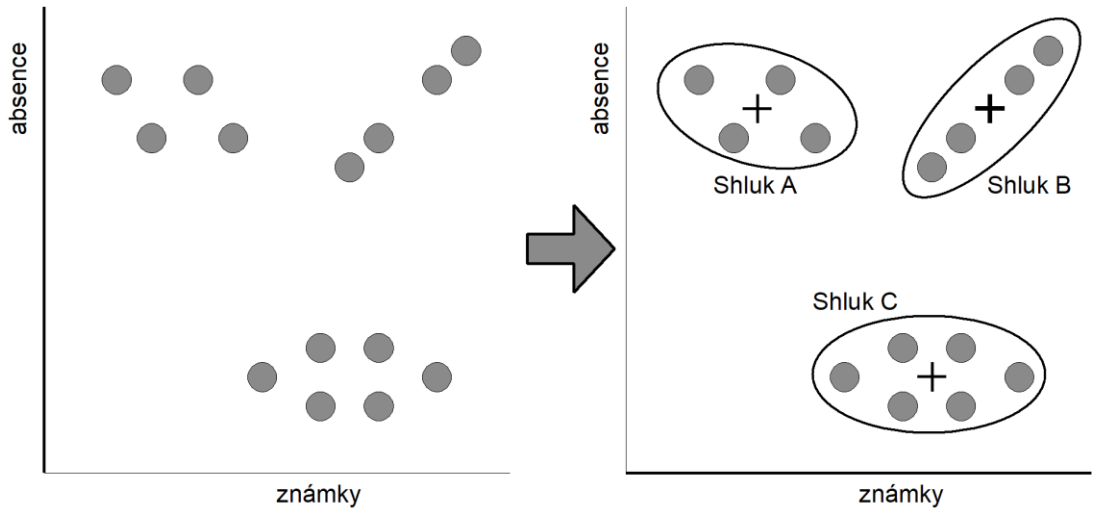
Učení bez učitele používá *neoznačená data*, tedy data nerozdělená na *závislé proměnné* (response variable) a nezávislé proměnné (predictor variable). Cílem učení bez učitele v rámci process miningu je buď *shluková analýza* (cluster analysis), nebo *objevování vzorů* (pattern discovery).

4.2.1 Shluková analýza

Shluková analýza má za cíl ve vstupních datech nalézt skupiny (shluky) sobě podobných instancí. Na rozdíl od *klasifikace* se u shlukové analýzy klade důraz na celek, nejen na jednotlivé části instancí. Cílem může být například nalezení podobné skupiny pacientů, či studentů. Pro naši potřebu se budeme soustředit na metody *k-means* a *aglomerativní hierarchické shlukování*.

4.2.1.1 K-means clustering

Jak již bylo zmíněno, jedná se o metodu shlukování instancí do shluků (tzv. clusters). Instance v jednom shluku by měly bát navzájem podobné a odlišné od instancí v jiných shlucích. Pro základní představu si představme data set se dvěmi hodnotami: *známky* a *absence*. Jednotlivé tečky v obrázku 4.1 představují studenty s daným průměrem známek a absencí za určité období. Pro vytvoření shluků se nejprve náhodně zvolí tři centroidy („středy“) a nyní se každá tečka (student) přiřadí k nejbližšímu centroidu. Poté jsou centroidy každého shluku přepočítány tak, aby byly těžištěm daného shluku.
Nyní se opět každý objekt přiřadí k nejbližšímu centroidu a proces se opakuje, dokud se centroidy a shluky neustálí.
Obrázek 4.2 znázorňuje situaci před a po shlukování a představuje závislost absence na průměrných známkách.



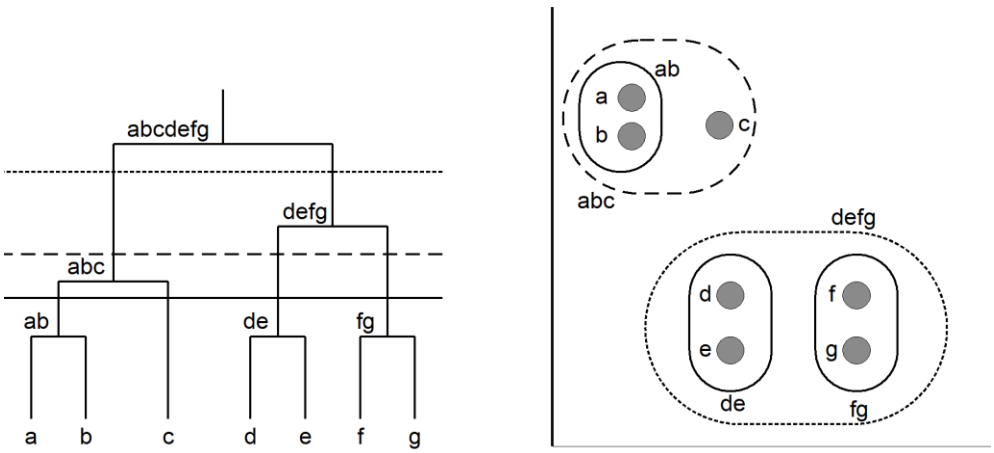
Obrázek 4.2: Vizualizace objektů před a po shlukování

4.2.1.2 Aglomerativní hierarchické shlukování

Postup u této metody je jednodušší, než u metody k-means. Každá instance je přiřazena do shluku. Dále jsou vyhledány dva shluky, které jsou si nejblíže a ty jsou spojeny do většího shluku. Stejný proces se opakuje až do doby, kdy jsou všechny instance v jednom shluku.

Díky hierarchickým vlastnostem této metody lze shlukování vizualizovat takzvaným *dendrogramem*. Dendrogram je je strom znázorňující jednotlivé shlukování instancí. Skrze dendrogram je vedena horizontální čára a v závislosti na její vertikální pozici lze vybrat nejvhodnější kombinaci shluků. V obrázku 4.2 jsou znázorněny tři stavy shluků. Plná čára znázorňuje tři shluky dvojic po dvou instancích a samostatném shluku *c*, přerušovaná čára znázorňuje shluk *abc* vzniklý v dalším kroku shlukování a tečkovaná čára zobrazuje shluky po předposledním kroku, tedy shluky *abc* a *defg*. Obecně lze říci, že čím výše je horizontální čára umístěna, tím znázorňuje méně shluků o více instancích, a naopak čím níže je umístěna, tím znázorňuje více shluků o méně instancích, nebo i jedné instanci.

Obrázek 4.3:



Znázornění shluků v různých úrovních abstrakce

5 Process discovery

Process discovery není možná bez správných záznamů událostí. Následující kapitola popisuje informace, které by měly být v těchto záznamech událostí přítomny. V závislosti na použitém postupu se mohou tyto požadavky lišit. Úkolem je získat takové údaje z různých zdrojů dat, například databází, protokolů zpráv, protokolů transakcí, systémů ERP apod. Při sloučení a extrahování dat hrají důležitou roli jak syntaxe, tak sémantika. Navíc v závislosti na otázkách, které se snažíme zodpovědět, jsou potřebné různé pohledy na dostupné údaje.

5.1 Zisk záznamů událostí

Výchozím bodem pro process mining jsou surová data skrytá v různých zdrojích dat. Nicméně nelze očekávat, že všechna data budou v jediném dobře strukturovaném zdroji dat. Skutečnost spočívá v tom, že data o událostech jsou obvykle rozptýlena mezi různými zdroji dat a často je třeba vynaložit značné úsilí na shromáždění relevantních dat.

Jedním z potenciálních problémů je rozdílný formát dat z různých datových zdrojů. V některých zdrojích může být datum uloženo ve formátu DD/MM/YYYY a v jiném zase ve formátu YYYY-MM-DD.

Data musí být také z jednotlivých zdrojů extrahována a převedena do záznamů událostí. Zde je nesmírně důležité stanovení rozsahu. Problém často není syntaktická konverze, ale výběr vhodných dat. Typické formáty pro ukládání protokolů událostí jsou XES (eXtensible Event Stream) a MXML (Mining eXtensible Markup Language).

5.2 Záznamy událostí

O struktuře záznamu událostí můžeme říci následující.

- *Proces* sestává z *případů (cases)*
- *Případ* sestává z *událostí* s tím, že každá událost se vztahuje právě k jednomu případu
- Události v případu jsou seřazené
- Události mohou mít *atributy*. Jako příklad atributů lze uvést například aktivitu, čas, cenu, zdroje atp.

Ne všechny události musí mít stejnou sadu atributů, ovšem typicky události vztahující se k téže aktivitě mají stejnou sadu atributů.

Příklad formálního zápisu několika standartních atributů je následující:

- $\#_{activity}(e)$ je *aktivita* spojená s událostí *e*
- $\#_{time}(e)$ je *timestamp* události *e*
- $\#_{resource}(e)$ je *zdroj* události *e*
- $\#_{trans}(e)$ je *typ transakce* události *e*, jako například *start*, *complete*, či *suspend*

Události mohou mít mnoho atributů. Často se na událost odkazujeme jménem jeho aktivity což však technicky není správné. Může dojít k mnoha událostem, které odkazují na stejnou aktivitu. V případě, že tyto události mohou odkazovat na stejnou instanci aktivity (například spuštění a dokončení událostí) nebo různé instance činnosti (například v smyčce).
Tento rozdíl je zvláště důležitý při měření časů služeb, čekacích dob atd. V situaci, ve které pro jeden případ stejná činnost začíná dvakrát, tj. dvě instance činnosti běží paralelně a pak se jedna z nich dokončí, nelze s jistotou určit, která činnost byla dokončena jako první a která jako druhá. Tento problém lze vyřešit přidáním informací do protokolu. Při zavádění systémů mohou být tyto informace do protokolů přidány snadno, a to přidáním atributu instance k jednotlivým aktivitám. Při řešení stávajících systémů to však tak jednoduché není, neboť by se musel modifikovat celý systém. Dalším řešení těchto problémů je přidáním timeoutu pro každou aktivitu.

5.2.1 XES

Standard pro ukládání protokolů událostí byl MXML až do nedávné doby, když byl nahrazen novějším standardem XES, který eliminuje mnoho nevýhod MXML. Největší výhodou XES je jeho rozšíření.

5.3 Úvod do process discovery

V této kapitole se zaměříme na α-algoritmus.
Tento α-algoritmus je základním algoritmem process miningu. Je-li dán jednoduchý záznam událostí, α-algoritmus z něj vytvoří Petriho síť, která může záznam zopakovat. A-algoritmus byl jedním z prvních algoritmů pro zjišťování procesů, které byly schopny řešit souběžnost. Nicméně α-algoritmus není ideálním algoritmem pro zjišťování procesů, protože má problémy s hlukem, neobvyklým nebo neúplným chováním a složitými směrovacími konstrukty. Nicméně poskytuje dobrý úvod do tématu. A-algoritmus je jednoduchý a mnoho jeho myšlenek je zakotveno v složitějších a robustnějších technikách, či jeho modifikacích.

Vstup pro algoritmus je jednoduchý záznam událostí. Záznam událostí se skládá z činností, které odpovídají přechodům v následně objevené Petriho síti. Výstupem algoritmu je označená Petriho síť nebo WF-síť.

Algoritmus prohledává protokol událostí pro vzory chování. Například pokud je událost *a* následována událostí *b*, ale *b* není nikdy následována *a*, pak lze předpokládat, že mezi *a* a *b* existuje kauzální závislost. Aby tato závislost byla znázorněna, měla by odpovídající Petriho síť mít místo propojující *a* a *b*.

Rozlišujeme čtyři relace pořadí založené na záznamech, jejichž cílem je zachytit relevantní vzory v protokolu:

- $>L_1$ zachycuje všechny páry aktivit, které jsou za sebou v přímé posloupnosti
- $\rightarrow L_1$ zachycuje kauzalitu aktivit, např. *a* je následováno *b*, ale nikdo tomu není naopak
- $\|L_1$ zachycuje aktivity, které se mohou následovat v libovolném pořadí
- $\#L_1$ zachycuje aktivity, které se nikdy nevyskytují jedna za druhou

5.4 Tři pojmy ekvivalence chování

V této kapitole si popíšeme tři nejznámější pojmy: *ekvivalence stop, bisimilarita* a *bisimilarita větvení*. Tyto pojmy jsou definovány pro pár přechodových systémů, nikoliv pro modely vyšších jazyků jako WF-sítě, BPMN apod. Každý model s podobnou sémantikou lze ale snadno převést do přechodového systému.

Ekvivalence stop je vlastnost dvou přechodových systémů, pro které platí, že jejich prováděcí sekvence jsou identické. Tato vlastnost však nezaručuje, že jsou modely stejné.
Bisimilarita neboli *bisimulační ekvivalence* je vlastnost beroucí v potaz moment volby. Dva přechodové systémy jsou bisimilarní v případě, že první systém je schopný napodobit každou změnu stavu druhého modelu a naopak.
Bisimilarita větvení bere v potaz takzvané tiché akce, které nejsou viditelné v záznamech událostí. Stejně jako u předchozí vlastnosti musí být systémy schopné napodobit každou změnu systému druhého, včetně tichých přechodů.

5.5 Výzvy objevování procesů

Jak již bylo zmíněno, α-algoritmus byl prvním algoritmem pro objevování procesů, který úspěšně zachycoval souběžnost událostí. V dnešní době však již jsou mnohem lepší algoritmy, které překonávají nedostatky α-algoritmu. Ať už se jedná o varianty α-algoritmu, nebo algoritmy založené na úplně jiném principu jako například syntéza založená na regionech, nebo genetic mining.

5.5.1 Hluk a nekompletnost

Další výzvou je výskyt takzvaného hluku a nekompletnost. Pro objevení dostačujícího procesního modelu je předpokládáno, že záznam událostí obsahuje *reprezentativní vzorek chování*. Ne všechny záznamy událostí však toto pravidlo splňují.

5.5.1.1 Hluk

Hluk v kontextu záznamu událostí neznamená nesprávné logování událostí, ale referuje spíše k výskytu speciálních událostí. Speciálními událostmi jsou myšleny události, které se v záznamu událostí nevyskytují tak často, nebo jsou speciální svým chováním. Pro process mining je důležité toto chování filtrovat, neboť nereprezentuje typické chování daného procesu.

5.5.1.2 Nekompletnost

Na rozdíl od hluku, který popisuje příliš velké množství událostí, a tedy i události vzácné, u nekompletnosti se jedná o situaci, kdy má záznam událostí tak malé množství dat, že není možné z něj odhalit strukturu daného modelu.

5.5.2 Křížová validace

V závislosti na problémy zmíněné v předchozí podkapitole, je zřetelné, že je na modelech nutné provádět takzvanou *křížovou validaci*. Záznam událostí je rozdělen na *tréningový log* a *testovací log*. Tréningový log je využit k získání daného procesního modelu a testovací log k zhodnocení kvality modelu na záznamech událostí, které vzorek, dle kterého bylo modelováno neobsahoval. Pokud tedy bude v testovacím logu mnoho záznamů, které nelze vytvořeným modelem replikovat, lze považovat daný model za nekvalitní.

Další metodou cross validace je *k-fold validace*, kdy je záznam událostí rozdělen na *k* částí a poté probíhá *k* testů. V každém testu je jedna podmnožina použita jako testovací vzorek a zbylé jako tréningový vzor.

Nevýhodou záznamů událostí je, že neobsahují tzv. negativní události, které daný model nemá být schopný reprezentovat. Jednou z možností, jak tuto skutečnost otestovat, je nejprve model otestovat za použití skutečných událostí a poté model otestovat testovacím logem obsahujícím uměle vytvořené události. Poté je porovnána výsledná kvalita získaná z testů a logicky by měla být kvalita vyšší pro test se skutečnými událostmi.

5.5.3 Čtyři konkurenční kritéria kvality

Kompletnost a hluk popisují kvalitu záznamů událostí, ale neříkají mnoho o kvalitě výsledného modelu. Pro zjištění kvality vytvořeného modelu jsou využívány čtyři hlavní kritéria kvality.

5.5.3.1 Vhodnost modelu

Vhodnost modelu může být brána z hlediska případů, tedy poměr stop, které lze ze záznamu plně přehrát od začátku až do konce, nebo z hlediska událostí, tedy poměr událostí, které jsou dle modelu opravdu uskutečnitelné.

5.5.3.2 Jednoduchost modelu

Jednoduchost modelu referuje na princip Occamovy břitvy: „Entity se nemají zmnožovat více, než je nutné.“, tedy by model neměl být složitější, než je nezbytně nutné a nejjednodušší model schopný popsat chování obsažené v záznamu, je modelem nejlepším.
Dvě zatím zmíněné metody však nejsou dostačující, neboť existují modely, jako například květinový model, které umožňují přehrání jakékoliv sekvence událostí v jakémkoliv pořadí. Z tohoto modelu však nelze nic vyčíst a pouze modeluje chování ze záznamu. Tento model splňuje jednoduchost a má perfektní vhodnost. Toto značí, že vhodnost a jednoduchost jsou nutnými kritérii, ale nejsou dostačující.

5.5.3.3 Přesnost a všeobecnost modelu

Model je přesný, pokud neumožňuje kromě stop v záznamu událostí příliš mnoho stop navíc. Model, který není přesný je příliš generalizovaný.

Model je příliš obecný, když kromě stop v záznamu událostí umožňuje mnoho dalších stop.

6 Conformance checking

Conformance checking (*kontrola souladu*) srovnává události ze záznamu událostí a události v procesním modelu. Cílem je najít shody či nedostatky v modelovaném chování a záznamu událostí a data, či procesní model na základě oněch shod upravovat.

Na neshody objevené mezi procesním modelem a záznamy událostí je možné nahlížet z několika pohledů. Pokud má být výsledný model popisný, tak nalezené rozdíly značí, že by měl být model vylepšen ta, aby zachycoval reálné chování popsané záznamy událostí lépe.
Pokud má být model normativní, můžou rozdíly poukazovat jak na *nechtěné odchylky*, tak na *chtěné odchylky*. U chtěných odchylek to může vést například k vylepšení reálného systému, neboť nekonformní procesní model mohl odhalit chování pro nestandartní situace.

Na conformance checking je tedy nutné nahlížet jak z pohledu „procesní model je špatný“, tak z pohledu „jsou data v záznamech událostí správná“. Tímto tématem se zabývá metoda *business alignment*, jejímž cílem je zajistit, že si model i záznamy událostí co nejlépe odpovídají.
Jedním z hlavních důvodů neshod je i fakt, že mnoho firem používá obecný software, který je pouze upraven, aby odpovídal požadavkům firmy. To může vést k neschopnosti daného software produkovat přesné záznamy událostí. Dalším velkým důvodem může být například fakt, že se procesy ve firmě mění rychleji než software, který je zaznamenává do záznamů událostí, tedy stejně jako v případě předchozím nakonec vyprodukuje nekvalitní záznamy událostí, které nejsou pro danou firmu aktuální.

Conformance checking je také vhodným nástrojem pro auditory společností, kterým umožňuje zkontrolovat, zda se procesy v dané firmě vykonávají v rámci hranic určených zúčastněnými stranami.

6.1 Opravování modelů

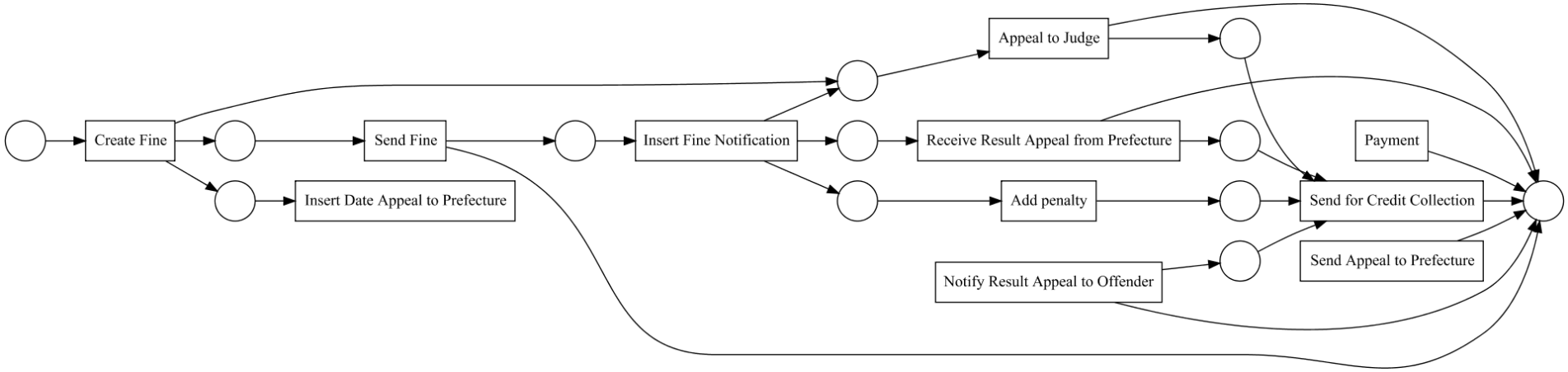
Důležitou součástí conformance checkingu je opravování modelů. Případy, kdy se procesní model a záznam událostí neshodují na procesech, vedou k úpravám modelu, nebo i jednotlivých procesů.
Předpokládejme však, že se za pomocí zjištěných neshod snažíme opravit vytvořený model, tedy ho převést do stavu, kdy odpovídá realitě co nejvíce. To může vést k odstranění přechodů, které nejsou využívány, nebo slouží pouze k provedení událostí ve speciálních situacích.
Srovnáním *matricí stop* (*footprint matrix*) záznamů a modelu lze zobrazit společné problémy. Tyto informace poté mohou být využity ať už návrhářem procesů, či softwarem, za účelem vylepšení modelu.

Na základě výsledků conformance checkingu například může dojít k zjištění, že pro kvalitnější model je nutné, aby jeho součástí byl další konkrétní atribut. Tato část conformance checkingu se nazývá *propojování záznamů a procesního modelu*.

6.2 Propojování záznamů událostí a modelu

Na první pohled by se mohlo zdát, že záznamy událostí a procesní model jsou propojené již od počátku, avšak jak bylo v začátku zmíněno, záznamy událostí musí být upravovány tak, aby odpovídaly požadavkům pro tvorbu modelu.

Propojením záznamů a modelu lze například obohatit model o časové značky, či informace o tom, kdo danou činnost prováděl. Tyto informace pak mohou například vést k vytvoření sociální sítě na základě množství interakcí mezi jednotlivými zaměstnanci a zkoumat tak vzory jejich chování.



Obrázek 6.1: Petriho síť získaná ze vstupních dat za pomoci nástroje ProM

7 Analyzovaná data

Pro potřeby této bakalářské práce jsem si jako zdrojová data zvolil volně dostupná data o managementu dopravních pokut [2]. Tyto data obsahují jeden proces, 150370 jednotlivých případů a 561470 událostí. O daných datech známe následující:

- Každý trace začíná událostí typu Create Fine
- Ve 44 % případů trace končí událostí Payment
- Celkem se ve zdrojových datech nachází 11 druhů událostí
    - Create Fine s výskytem 26.8 %
    - Send Fine s výskytem 18.5 %
    - Add penalty s výskytem 14.2 %
    - Insert Fine Notification s výskytem 14.2 %
    - Payment s výskytem 13.8 %
    - Send for Credit Collection zastoupené v 10.5 %
    - Insert Date Appeal to Prefecture
    - Send Appeal to Prefecture
    - Receive Result Appeal from Prefecture
    - Notify Result Appeal to Offender
    - Appeal to Judge

    Posledních 5 událostí bylo ve datech zastoupeno méně než 1 % (celkem 1.921 %)

Na následující straně je k nahlédnutí Petriho síť vytvořená za pomoci nástroje ProM z těchto vstupních dat.

8 Závěr

Tato bakalářská práce se ukázala být velice složitou, zejména kvůli absenci českých zdrojů. Mnoho textů bylo nutné pro potřeby bakalářské práce překládat, neboť jejich české překlady jsou nedostupné. Dostupnost vzorových dat pro process mining byla poměrně dobrá, ale častokrát byla data v jiném jazyce, než je angličtina či čeština. Téma process miningu pro mě bylo velice složité, částečně i z důvodu toho, že mu není při studiu přikládána větší pozornost, byť má v dnešní době, kdy je na množství dat a informací kladen velký důraz, čím dál tím větší význam. Práce je pojata teoreticky především kvůli co nejlepšímu obsažení prvních tří bodů zadání. Bod čtyři zadání si mi bohužel nepodařilo splnit, byť jsem se za pomoci nástroje ProM snažil k prediktivní údržbě dopracovat.

Celkové bych práci zhodnotil jako velmi časově náročnou a pro člověka bez předchozích znalostí process miningu velice komplikovanou.

Literatura

[1] VAN DER AALST, Wil. Process mining. Communications of the ACM [online]. ACM, 2012, 55(8), 76-83 [cit. 2017-02-01]. DOI: 10.1145/2240236.2240257. ISSN 0001-0782. Dostupné na URL: < http://dl.acm.org.ezproxy.lib.vutbr.cz/citation.cfm?doid=2240236.2240257>

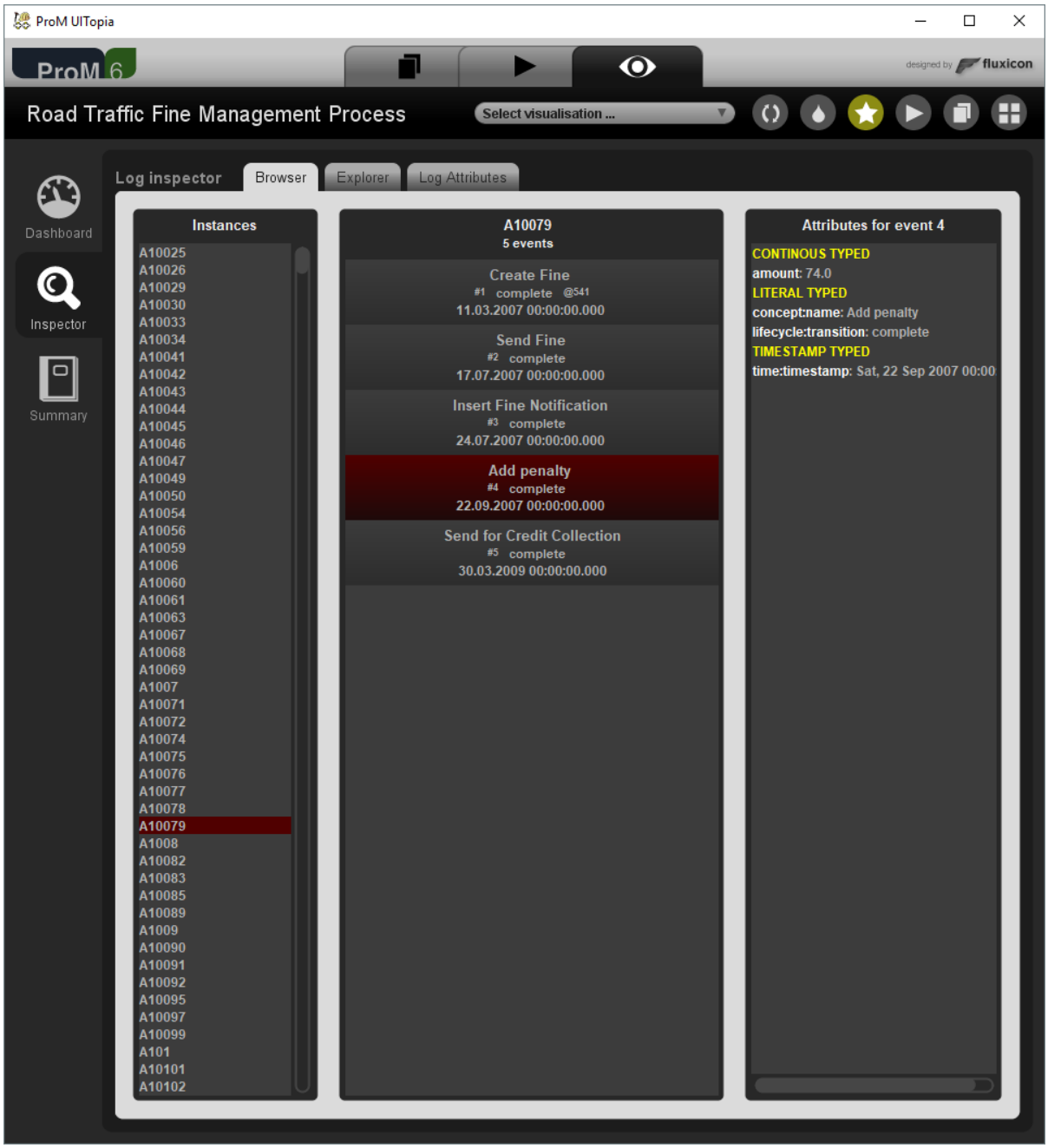
[2] Road Traffic Fine Management Process. *4TU – Centre for research data* [online]. Delft, 2015 [cit. 2017-04-10]. Dostupné z: http://data.4tu.nl/repository/uuid:270fd440-1057-4fb9-89a9-b699b47990f5

[3] WIL M. P. VAN DER AALST. *Process mining: discovery, conformance and enhancement of business processes*. Heidelberg; New York: Springer, 2011. ISBN 9783642193446.

Seznam příloh

Příloha 1. Náhled na uživatelské prostředí programu ProM ...
Příloha 2. Náhled na uživatelské prostředí programu ProM s pohledem na prohlížení událostí ...
Příloha 3. CD/DVD ...

Obrázek 8.1: Úvodní obrazovka aplikace ProM 6.6 po provedení alpha analýzy zdrojových dat a vytvoření petriho sítě



Obrázek 8.2: Náhled na uživatelské rozhraní programu ProM 6.6 s detailem na jednu konkrétní událost daného případu