



UHASSELT

KNOWLEDGE IN ACTION

Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Een data-gestuurd conformance checking methode: aligneren met het juiste modelpad

Caitlin Vanlessen

Scriptie ingediend tot het behalen van de graad van master handelsingenieur in de beleidsinformatica

PROMOTOR :

Prof. dr. Benoit DEPAIRE



UHASSELT

KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

2022
2023



Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Een data-gestuurd conformance checking methode: aligneren met het juiste modelpad

Caitlin Vanlessen

Scriptie ingediend tot het behalen van de graad van master handelsingenieur in de beleidsinformatica

PROMOTOR :

Prof. dr. Benoit DEPAIRE

Een data-gestuurde conformance checking methode: aligneren met het juiste modelpad

Caitlin Vanlessen

UHasselt

Abstract Conformance checking laat toe om na te gaan waar het werkelijk uitgevoerd proces afwijkt van het gemodelleerde proces. De meest gebruikte conformance checking techniek is de cost-based alignment methode waar de focus ligt op het minimaliseren van de kost. Er is echter geen garantie dat deze methode aligneert met het modelpad dat men in realiteit had moeten volgen. Om te weten welk pad men in realiteit had moeten volgen, moet men rekening houden met de beslissingslogica van een procesmodel. In deze paper stellen we een context-aware alignment methode voor die eerst focust op het aligneren met het correcte modelpad en dan pas op het minimaliseren van de kost. Alsook zetten we een experiment uit een die de impact van het al dan niet meenemen van beslissingslogica in de alignment methode meet. Hieruit blijkt dat de cost-based alignment methode herhaaldelijk aligneert met het foutieve modelpad en bijgevolg wordt de kostmaatstaf onderschat. Over het algemeen stellen we vast dat het al dan niet meenemen van de beslissingslogica in de alignment methode wel degelijk een verschil uitmaakt.

Keywords: Conformance checking, alignment, data perspectief, beslissingslogica

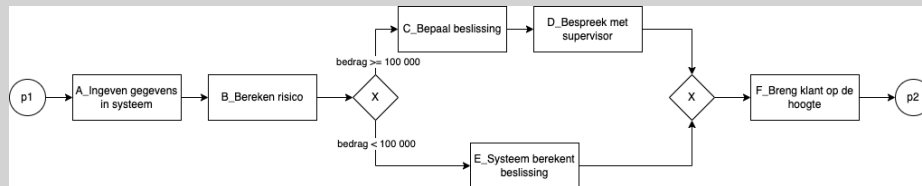
1 Introductie

Organisaties definiëren de werking van hun organisatie in de vorm van processen. De werkelijke uitvoering van deze processen wordt (deels) bijgehouden in een eventlog. Processen omvatten verschillende systemen en menselijke activiteiten. Dit maakt dat er in de realiteit kan afgeweken worden van het gemodelleerd procesmodel. Deze afwijkingen worden ontdekt en onderzocht met behulp van conformance technieken. In conformance checking wordt het geobserveerd gedrag in de eventlog vergeleken met het gemodelleerd gedrag in het procesmodel [13]. Dit stelt organisaties in staat om de mate van conformance met het procesmodel te meten en problemen te identificeren [8]. Zo is bijvoorbeeld conformance checking in de context van auditing essentieel om inbreuken ten opzichte van de wetgeving vast te stellen [7].

De eventlog vergelijken met het procesmodel is zeer complex. Een procesmodel kan namelijk op vele manieren gevolgd worden. Hierdoor bevat een procesmodel een grote set aan mogelijke procesuitvoeringen en is het niet altijd

duidelijk met welke procesuitvoering je een trace uit de eventlog moet vergelijken. Een mogelijke oplossing voor dit probleem is om de trace te vergelijken met elk mogelijk modelpad en uiteindelijk het pad te selecteren dat de laagste kost oplevert. Dit is wat Adriansyah et al. (2013) [2] doen in hun bekende cost-based alignment techniek. Het doel van Adriansyah et al. (2013) [2] was om de optimal alignment tussen een trace van het procesmodel en een trace van de eventlog te vinden. Een optimal alignment is de alignment met de laagste kost tussen een trace van het model en een trace van de eventlog. Wat die kost juist bepaalt, is afhankelijk van de gedefinieerde kostenfunctie. Vaak wordt het aantal afwijkingen gezien als een kost, maar dit is niet altijd het geval. Het exhaustief vergelijken van een trace in de eventlog met elk mogelijk modelpad is een tijdrovende opdracht. Vandaar hebben Adriansyah et al. (2013) [2] het vinden van een optimal alignment vertaald in een shortest-path probleem. Hierdoor kunnen sommige modelpaden snel genegeerd worden omdat deze altijd een hogere kost zullen opleveren dan het beste modelpad dat men reeds geïdentificeerd heeft [2].

In wat volgt schetsen we een praktisch voorbeeld om onze kritiek op de cost-based techniek van Adriansyah et al. (2013) [2] te verduidelijken. Figuur 1 stelt een fictief procesmodel voor, afgebeeld als een BPMN-model. Het proces situeert zich binnen een kredietinstelling en start telkens wanneer een klant een kredietaanvraag doet. Een eerste stap bestaat erin om alle gegevens van de klant en de aanvraag in te geven in het operationele systeem. Op basis van deze gegevens wordt vervolgens het risico berekend. Afhankelijk van het bedrag in de kredietaanvraag wordt er een ander pad gevolgd. De exclusieve (X) gateway geeft aan dat slechts één van de twee paden gevolgd mag worden. Als het bedrag beneden de 100 000 euro is, maakt het systeem een automatische beoordeling en beslist het of er al dan niet een lening wordt toegekend. Als het bedrag gelijk aan of boven de 100 000 euro is, volgt er een handmatige beoordeling en beslissing. Deze beslissing dient vervolgens voorgelegd te worden aan de supervisor wiens goedkeuring vereist is. Tenslotte, brengen we de klant op de hoogte van het besluit over zijn kredietaanvraag. Volgens het procesmodel in Figuur 1 zijn er slechts twee procespaden mogelijk. Voor de eenvoud geven we hier enkel de eerste letter van elke activiteit weer. De twee mogelijke paden zijn: (A,B,E,F) voor kredietaanvragen met een bedrag beneden de 100 000 euro en (A,B,C,D,F) in het andere geval.



Figuur 1: BPMN-model kredietaanvraagproces

Stel in de eventlog vinden we volgende trace (A,B,E,D,F) terug. Dit wil zeggen dat in de realiteit het procesmodel in Figuur 1 niet gevolgd is want deze trace kunnen we niet afspelen op het model in Figuur 1. Verder onderzoek naar de oorzaak van deze afwijking is dus noodzakelijk. We voeren de cost-based conformance checking techniek van Adriansyah et al. (2013) [2] uit in de veronderstelling dat de kostenfunctie wordt bepaald door het aantal afwijkingen. De afwijkingen worden weergegeven door >>. De alignments worden afgebeeld op Figuur 2. Een alignment bestaat uit moves die we verticaal aflezen. We spreken van een synchronous move als we eenzelfde activiteit zowel in de trace als in het model zien. Een non-synchronous move komt voor wanneer we in de trace of het model een afwijking (>>) zien. We spreken van een model move als de afwijking (>>) zichtbaar is in de trace. Bij een log move is de afwijking (>>) zichtbaar in het model. In alignment A zien we twee non-synchronous moves: een model move en een log move. (>>, C) is een model move die aangeeft dat activiteit C in het model wordt uitgevoerd, maar ontbreekt in de trace. (E, >>), daarentegen, is een log move aangezien activiteit E in de (log) trace wordt uitgevoerd, maar in het model vindt deze activiteit niet plaats. De andere moves zoals (A,A) zijn synchronous moves en worden niet gezien als een afwijking [1].

Trace	A	B	>>	E	D	F
Model	A	B	C	>>	D	F

alignment A

Trace	A	B	E	D	F
Model	A	B	E	>>	F

alignment B

Figuur 2: Alignments van trace (A,B,E,D,F) uit eventlog

Aangezien de kostenfunctie wordt bepaald door het aantal afwijkingen (>>), heeft alignment A een kost van 2 en alignment B een kost van 1. Alignment B heeft de laagste kost dus dit is, volgens de techniek van Adriansyah et al. (2013) [2], de optimal alignment voor de trace (A,B,E,D,F) uit de eventlog.

Echter, zou het doel van de alignment methode moeten zijn om de trace te vergelijken met het modelpad dat men had moeten volgen in de realiteit. Om te weten welk modelpad men had moeten volgen, moeten we de data-attributen van die trace vergelijken met de beslissingslogica in het procesmodel. Zo weten we dat de trace (A,B,E,D,F) is uitgevoerd door resource X en gekoppeld is aan een specifieke kredietaanvraag van 150 000 euro. Door deze extra gegevens weten we met zekerheid dat het modelpad dat in de realiteit had moeten gebeuren (A,B,C,D,F) is, aangezien dit het modelpad is voor kredietaanvragen boven de 100 000 euro.

In ons specifieke voorbeeld is dus alignment A de meest correcte alignment gegeven de context, aangezien hier gealigneerd wordt met het correcte modelpad (A,B,C,D,F). We zien dat men activiteit E heeft uitgevoerd, terwijl men activiteit C had moeten uitvoeren. In het procesmodel in Figuur 1 vertaalt zich dit in: men heeft de beslissing door het systeem laten berekenen, terwijl dit eigenlijk een handmatige beslissing had moeten zijn. Bijgevolg resulteert deze alignment in een kost van 2. In alignment B heeft men de supervisor aangesproken (activiteit D), terwijl dit niet had moeten gebeuren. Dit resulteert in een kost van 1. Desondanks dat men aligneert met het foutieve modelpad en bijgevolg ook de kostmaatstaf onderschat, is alignment B de optimal alignment volgens de techniek van Adriansyah et al. (2013) [2]. Deze techniek vergelijkt de trace met het modelpad dat de laagste alignment kost oplevert en veronderstelt impliciet ook dat dit het pad is dat men had moeten volgen.

Dit voorbeeld toont dus duidelijk aan dat de cost-based techniek van Adriansyah et al. (2013) [2] niet verzekert dat de optimal alignment de alignment is met het modelpad dat men had moeten volgen overeenkomstig met de beslissingslogica.

De optimal alignment van de cost-based techniek van Adriansyah et al. (2013) [2] is dus niet per definitie de alignment tussen de werkelijk uitgevoerde trace en het modelpad dat men, volgens de beslissingslogica, had moeten volgen. Hun techniek focust op het minimaliseren van de kost, maar zonder rekening te houden met de context waardoor zij vaak aligneren met het verkeerde modelpad. Bijgevolg is de techniek in sommige gevallen te optimistisch en wordt de kostmaatstaf onderschat.

In deze paper zetten we een nieuwe methode uiteen namelijk de context-aware alignment methode. Deze methode bouwt verder op de cost-based techniek van Adriansyah et al. (2013) [2], maar met de inclusie van beslissingslogica in het procesmodel. Door rekening te houden met de beslissingslogica, gebaseerd op data-attributen, kunnen we het procesmodel vereenvoudigen tot het modelpad dat in realiteit gevolgd had moeten worden. Voorts, zetten we een experiment uiteen over de impact van het al dan niet meenemen van beslissingslogica in je alignment methode.

In wat volgt positioneren we ons onderzoek in de huidige literatuur, om vervolgens onze context-aware alignment methode uiteen te zetten. Hierna bespreken we de gehanteerde data en ons experiment, gevolgd door onze resultaten. We ronden af met een conclusie en toekomstige onderzoeksgebieden.

2 Related work

In deze paper focussen we op het onderzoeksgebied conformance checking waar geobserveerde event data vergeleken wordt met het gemodelleerde procesmodel. In de literatuur bestaan drie grote conformance checking algoritmes: rule checking, token replay en alignments [13]. Een procesmodel kan op verschillende manieren worden uitgevoerd en elke manier wordt gedefinieerd in een modelpad. Hieruit kunnen we regels afleiden waaraan al deze modelpaden voldoen. Regels kunnen gezien worden als beperkingen opgelegd door het procesmodel [14]. Een voorbeeld van zo'n regel is 'Een kredietaanvraag kan hoogstens één keer geaccepteerd worden'. In rule checking wordt voor elke trace in de log nagegaan in welke mate aan deze regels voldaan is. Er bestaan verschillende types regels (unair, binair, kardinaliteit, exclusiviteit,...), maar zelfs als men rekening kan houden met al deze types blijft de conformance analyse nog steeds onvolledig. Voorts, is dit een vrij inefficiënte techniek, aangezien het aantal regels redelijk groot kan worden bij een complex procesmodel [13].

De tweede conformance checking techniek, token replay, neemt de eventlog als startpunt. In deze techniek wordt elke trace in de eventlog, activiteit per activiteit, afgespeeld in het model. Wanneer een taak in het model wordt uitgevoerd, consumeert en produceert deze taak een token. Wanneer bij het afspelen van de trace op het model een taak moet uitgevoerd worden en er is geen token om te consumeren, noemen we dit een missing token. Een remaining token is een token die geproduceerd wordt door het uitvoeren van een taak, maar uiteindelijk niet meer geconsumeerd wordt en dus overblijft in het model. Na het afspelen van de trace wordt de mate van conformance bepaald door het aantal missing en

remaining tokens in het model [11]. Token replay slaagt erin om de conformance analyse vollediger te maken dan de rule checking techniek, maar de analyse is nog steeds niet volledig. Daarenboven, komt er heel veel ambiguïteit kijken bij token replay wat de techniek niet-deterministisch maakt [13].

Deze paper focust echter op de trace alignment methode. De alignment methode is gegroeid uit de vorige techniek, token replay. Waar token replay een link legt tussen een trace en een set van sequenties waarop taken uitgevoerd kunnen worden in het model, legt de alignment methode een directe link tussen een trace en een geldige procesuitvoering [13]. In een alignment wordt een trace uit de eventlog geconnecteerd met een procesuitvoering uit het model. Dit kan resulteren in afwijkende moves tussen de trace en het model, aangegeven door $>>$. Er worden kosten toegekend aan deze afwijkingen en het doel is om deze kost te minimaliseren. De alignment met de laagste kost is de optimal alignment. De alignment methode is superieur aan de voorgaande twee conformance checking technieken, aangezien de conformance analyse volledig is en men afwijkingen op het niveau van individuele events en taakuitvoeringen kan detecteren [13].

Echter, wordt er nog veel onderzoek gedaan naar de rekenkracht die noodzakelijk is om de (optimal) alignment te berekenen [13]. Elke trace vergelijken met elk mogelijk modelpad in een procesmodel is namelijk een tijdrovende opdracht. De methode van Adriansyah et al. (2013) [2] probeert deze duurtijd te reduceren door het vinden van een optimal alignment te transformeren in een shortest path probleem. Zij identificeren een beste modelpad met een laagste kost en elk modelpad dat altijd een hogere kost zal opleveren dan het beste modelpad kan genegeerd worden. Als ze een modelpad tegenkomen met een lagere kost, wordt het beste modelpad geüpdatet [2].

Bestaande conformance checking technieken focussen voornamelijk op de control-flow i.e. de volgorde van activiteiten. Echter, is de data-flow ook belangrijk en kan dit interessante inzichten leveren. Het bestuderen van de data-flow zorgt er namelijk voor dat men regels kan ontdekken die verklaren waarom het ene pad gevolgd wordt en niet het andere [5]. Het meenemen van het data-perspectief in de alignment methode vereist nog meer rekenkracht. De Leoni et al. (2014) [3] ontwikkelde daarom een techniek om data-aware conformance checking problemen te decomposeren in kleinere problemen door de SESE - decomposition.

De context-aware alignment methode, gepresenteerd in deze paper, neemt het data-perspectief mee door rekening te houden met de beslissingslogica van een model. Beslissingslogica is een essentieel onderdeel van decision mining waar het doel is om regels te ontdekken die verklaren onder welke condities men het ene pad volgt en niet het andere pad [12]. De Leoni et al. (2012) [6] onderzochten of het data-perspectief kon meegenomen worden in een aangepaste versie van het A* algoritme. Het doel van dit algoritme is het vinden van het pad met de laagste kost tussen twee knooppunten in een netwerk waar kosten worden toegekend aan elk knooppunt. Hun techniek geeft de mate van conformance weer en benadrukt waar de afwijkingen gebeuren waarbij het control-flow en data-perspectief worden aangetoond [6]. Echter, kan deze techniek niet om-

gaan met variabelen met een oneindig domein. Dit zijn data-attributen die een oneindig aantal waarden kunnen aannemen. In praktijk zien we voornamelijk variabelen met een oneindige domein terugkomen wat de praktische relevantie van bovenstaande techniek in gedrang brengt [10]. Een andere manier om het data-perspectief mee te nemen is de ontwikkeling van een ILP-based techniek. Zij gebruiken state-of-the-art technieken om control-flow alignments te verkrijgen om vervolgens deze alignments uit te breiden zodat ze rekening houden met het data perspectief. Hiervoor wordt een Integer Linear Programming probleem geconstrueerd en opgelost voor elke trace [4]. Mannhardt et al. (2016) [10] verwerken het data-perspectief in een customizable cost function en houden op die manier gelijktijdig rekening met zowel het control-flow als het data-flow perspectief. De ILP-based techniek, daarentegen, focust eerst op het control-flow perspectief en dan pas op het data-flow perspectief. Mannhardt et al. (2016) [10] uiten hun kritiek op de ILP-based techniek, aangezien deze techniek misleidend is wanneer het control-flow en het data-flow perspectief sterk gerelateerd zijn.

3 Context-aware alignment methode

In deze paper zetten we een methode uit een die rekening houdt met de beslissingslogica van een procesmodel. Deze methode geeft een meer correcte alignment en bijbehorende kost, aangezien de context-aware alignment methode verzekert dat de alignment gebeurt met het modelpad dat men in realiteit had moeten volgen. De methode is omgezet in een algoritme en wordt weergegeven in Algorithm 1. In wat volgt bespreken we kort hoe elke stap exact wordt uitgevoerd.

Algorithm 1 Context-aware alignment methode

Input: procesmodel met beslissingslogica, eventlog

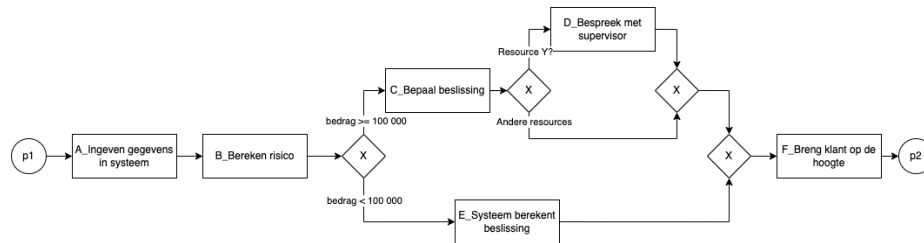
Output: alignment per case

- 1: Verdeel de eventlog in sublogs van cases die dezelfde beslissingslogica volgen
 - 2: **for** elke sublog **do**
 - 3: Vereenvoudig procesmodel tot het deel dat overeenkomt met de beslissingslogica
 - 4: Pas cost-based alignment methode toe
 - 5: **end for**
-

3.1 Eventlog verdelen in sublogs

Een eerste stap bestaat erin om de eventlog te verdelen in sublogs van cases die dezelfde beslissingslogica volgen. Men dient case per case stapsgewijs door het procesmodel te lopen en op de beslissingspunten na te gaan welke beslissingslogica naar True evolueert. Om dit te verduidelijken wordt er een praktisch voorbeeld gegeven. In Figuur 3 is een uitbreiding van het eerdere kredietaanvraagproces terug te vinden. De beslissing wordt nu enkel besproken met de

supervisor als de beslissing genomen wordt door resource Y. Andere resources dienen dit niet te doen. In Tabel 1 vinden we een uittreksel van de log bijbehorend aan het kredietaanvraagproces. In de log zien we vier cases telkens gekenmerkt door hun case id, een event en een timestamp. Alsook zien we voor elke case de waarden van hun data-attributen (Bedrag en Resource).



Figuur 3: BPMN-model kredietaanvraagproces uitbreiding

Case id	Event	Timestamp	Bedrag	Resource
1	A_Ingeven gegevens in systeem	13:07:49	50 000	
1	B.Bereken risico	16:56:07	50 000	
1	C.Bepaal beslissing	11:34:02	50 000	Z
1	F_Breng klant op de hoogte	14:17:39	50 000	Z
2	A_Ingeven gegevens in systeem	16:46:09	150 000	
2	B.Bereken risico	10:00:56	150 000	
2	C.Bepaal beslissing	15:08:34	150 000	Y
2	D.Bespreek met supervisor	09:25:15	150 000	Y
3	B.Bereken risico	16:49:04	200 000	
3	C.Bepaal beslissing	10:02:47	200 000	Z
3	F_Breng klant op de hoogte	13:48:02	200 000	Z
4	A_Ingeven gegevens in systeem	17:04:29	70 000	
4	B.Bereken risico	10:37:36	70 000	
4	E_Systeem berekent beslissing	16:05:27	70 000	

Tabel 1: Log kredietaanvraagproces

Om te bepalen tot welke sublog case id 1 behoort, lopen we stapsgewijs door het procesmodel in Figuur 3. We stoppen bij het eerste beslissingspunt dat beslissingslogica bevat, namelijk $\text{bedrag} \geq 100\,000$ of $\text{bedrag} < 100\,000$. We lezen in Tabel 1 af dat case id 1 geassocieerd wordt met een kredietaanvraag voor een bedrag van 50 000 euro. Bijgevolg weten we tot dit punt dat het modelpad, dat case id 1 in realiteit had moeten volgen, (A,B,E) is. Hierna komen we geen nieuw beslissingspunt meer tegen en volgen we het enige pad verder dat ons tot het einde van het model brengt. We concluderen dat case id 1, volgens de beslissingslogica, modelpad (A,B,E,F) moet volgen. De vier events in Tabel 1

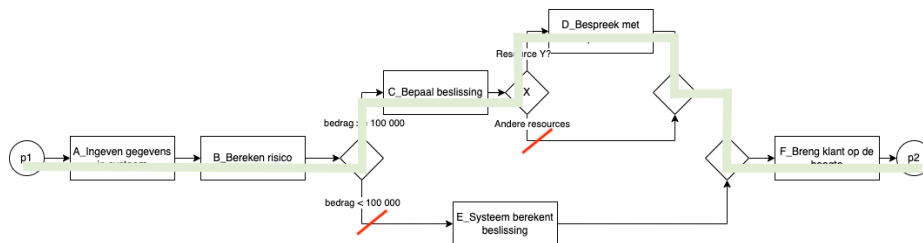
geassocieerd met case id 1 worden in een eerste sublog geplaatst. Deze sublog wordt gekenmerkt door het modelpad (A,B,E,F).

We doen exact hetzelfde voor case id 2. Aangezien we een bedrag van 150 000 euro aflezen in Tabel 1 voor case id 2, weten we dat we het bovenste pad in Figuur 3 moeten volgen. Dit levert al een tussentijds modelpad (A,B,C) op. We komen opnieuw een beslissingspunt tegen, namelijk of de beslissing gemaakt is door resource Y of niet. In Tabel 1 lezen we dat resource Y de beslissing maakt voor case id 2 dus we kunnen het modelpad verder aanvullen tot het einde namelijk (A,B,C,D,F). Alle events in Tabel 1 die horen bij case id 2 worden opgeslagen in een tweede sublog, die gekenmerkt wordt door het modelpad (A,B,C,D,F).

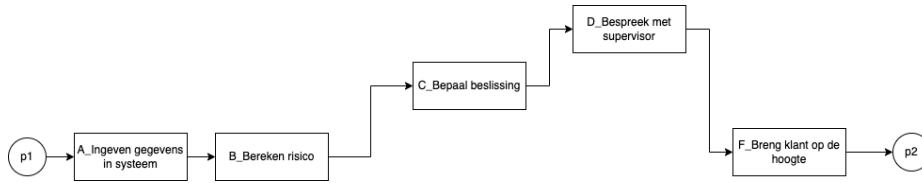
Voor case id 3 volgt exact dezelfde procedure. Het resultaat is een derde sublog gekenmerkt door modelpad (A,B,C,F). Wanneer we de procedure volgen voor case id 4 merken we op dat het modelpad, dat deze case id in realiteit moet volgen, exact hetzelfde modelpad is als dat van case id 1 namelijk (A,B,E,F). Alle events in Tabel 1 geassocieerd met case id 4 worden toegevoegd aan de eerste sublog, die ook alle events van case id 1 reeds omvat. Case id 1 en 4 volgen dezelfde beslissingslogica en bijgevolg zitten deze in dezelfde sublog.

3.2 Procesmodel vereenvoudigen

Na de eerste stap in Algorithm 1, volgt er een for-loop voor elke gedefinieerde sublog in stap 1. Aangezien elke sublog geassocieerd wordt met een specifiek modelpad, dienen we telkens het procesmodel zo te vereenvoudigen zodat het model correspondeert met het specifieke modelpad. Opnieuw lopen we stapsgewijs door het procesmodel, we elimineren de paden die niet mogelijk zijn volgens het modelpad en we volgen het model tot het einde. Een praktisch voorbeeld wordt afgebeeld in Figuur 4. Dit is het modelpad (A,B,C,D,F) geassocieerd met sublog 2. Enkel de activiteiten op de groene lijn worden behouden in het model en de rode paden worden verwijderd uit het model. Omdat het een BPMN-model is, moet een split-gateway minstens 2 uitgaande pijlen hebben en een joint-gateway minstens 2 inkomende pijlen. Om ervoor te zorgen dat het BPMN-model nog steeds een correct model is, moeten de gateways verwijderd worden. Het eindresultaat, waarop in een volgende stap de klassieke cost-based alignment methode wordt uitgevoerd, wordt weergegeven in Figuur 5.



Figuur 4: Tussenstap Modelpad (A,B,C,D,F) van sublog 2



Figuur 5: Finale Modelpad (A,B,C,D,F) van sublog 2

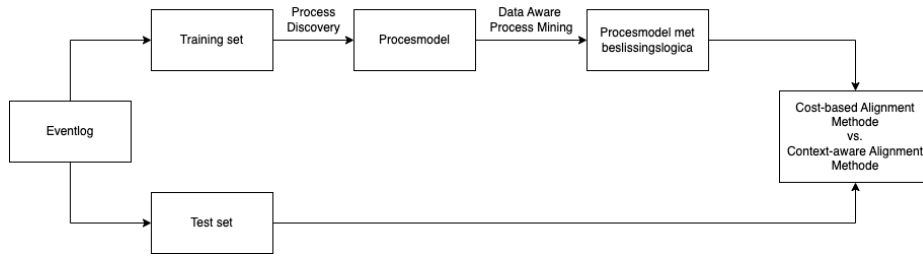
Voorts, hebben we twee relevante assumpties gemaakt in onze context-aware alignment methode. Een eerste assumptie gaat als volgt: als er op een beslissingspunt geen beslissingslogica gegeven is, dan vereenvoudigen we dit beslissingspunt niet. Stel dat bijvoorbeeld de beslissingslogica op de exclusive gateway voor activiteit D ontbreekt, dan kunnen we dit deel van het procesmodel niet vereenvoudigen. Bijgevolg zullen beiden paden van deze gateway nog zichtbaar zijn in het finale model in Figuur 5. Hierdoor kunnen we niet altijd één modelpad aanduiden als het pad dat men in realiteit had moeten volgen, aangezien er nog steeds meerdere modelpaden mogelijk zijn. Echter, hebben we het model wel nog steeds enorm kunnen vereenvoudigen. Er doet zich ook een situatie voor waar wel beslissingslogica gegeven is, maar een essentieel data-attribuut wordt niet bijgehouden. De tweede assumptie houdt rekening met deze situatie. Stel dat het data-attribuut resource in Tabel 1 niet wordt bijgehouden, dan kunnen we de beslissingslogica op dit punt niet nagaan en bijgevolg kunnen we dus ook niets vereenvoudigen in het model. In deze situatie is het dan ook niet mogelijk om één modelpad aan te duiden als het pad dat men in realiteit had moeten volgen. Echter, is dit geen onrealistische situatie, aangezien ontbrekende data in realiteit nog steeds een vaakvoorkomend probleem is.

4 Experiment

In deze sectie beschrijven we het doel en de opzet van ons experiment. Als ook omschrijven we de gehanteerde eventlog en het bijbehorende procesmodel. Ten slotte, bespreken we de resultaten van ons experiment. Een schematische voorstelling van ons experiment is terug te vinden in Figuur 6.

4.1 Opzet experiment

Met ons experiment willen we de impact van het al dan niet meenemen van beslissingslogica in je alignment methode nagaan. Dit doen we door zowel de cost-based alignment methode van Adriansyah et al. (2013) [2], als onze context-aware alignment methode uit te voeren op de test set en de uitkomsten te vergelijken. We focussen op verschillen tussen de alignments en de kostmaatstaf. Voorts, hebben we de eventlog opgesplitst in een training en een test set om overfitting te vermijden. De test set omvat 100 cases. De training set is de gehele eventlog exclusief deze 100 cases. De training set wordt gebruikt om het model en de beslissingslogica te ontdekken.



Figuur 6: Schematische voorstelling experiment

4.2 Input data

4.2.1 Eventlog

De eventlog¹ is een reële log over het proces van het uitschrijven van boetes in Italië. De log omvat data van 150 370 cases waarbij een case een verkeers-overtreding voorstelt. De eventlog heeft een tijdsperiode van 13 jaar en omvat 11 verschillende activiteiten. Er komen 164 unieke traces voor. Een trace is een sequentie van activiteiten. Voor elke case worden er ook een aantal data-attributen bijgehouden die essentieel zijn om de beslissingslogica op te bouwen. In totaal zijn er 12 data-attributen in de eventlog, waarvan we er enkel 10 in beschouwing nemen. Het data-attribuut *dismissal* wordt namelijk voor 108 841 cases niet bijgehouden en het data-attribuut *matricola* zelfs voor geen enkele case. Er is een gelijke verhouding tussen het aantal continue en categorische data-attributen. In wat volgt bespreken we eerst de 5 continue data-attributen met enkele kengetallen (Tabel 2) en vervolgens de 5 categorische data-attributen (Tabel 3).

Data-attribuut	Minimum	Maximum	Mediaan	Gemiddelde
payment_amount	0	3975	36	46,4
expense	0	76	11,9	11,9
total_payment_amount	0	4021	0	17,8
amount	0	8000	38	63,7
points	0	10	0	0,1

Tabel 2: Kengetallen continue data-attributen

Het eerste continue data-attribuut, *payment_amount*, geeft het boetebedrag weer. Dit bedrag ligt tussen 0 en 3975 euro met een mediaan van 36 euro en een gemiddelde van 46,4 euro. *Expense* geeft de administratiekosten voor het opstellen van de boete weer. Deze kost ligt tussen de 0 en 76 euro en heeft een mediaan en gemiddelde van 11,9 euro. *Total_payment_amount* stelt de fee voor die aangerekend wordt bij een laattijdige betaling, dit heeft een bereik van 0 tot

¹ De eventlog is publiek toegankelijk op <https://data.4tu.nl/articles/dataset/Road.Traffic.Fine.Management.Process/12683249?file=24018146>

4021 euro en heeft een mediaan van 0 euro en een gemiddelde van 17,8 euro. Het data-attribuut *amount*, is het totale boetebedrag incl. administratiekosten en fees wegens laattijdige betaling. Dit bedrag ligt tussen 0 en 8000 euro, met een mediaan van 38 euro en een gemiddelde van 63,7 euro. Sommige landen werken met een rijbewijs met punten, dit wilt zeggen dat alle overtredingen worden opgeteld en vanaf een bepaald punt wordt je rijbewijs ingetrokken. Dit wordt weergegeven door het laatste continue data-attribuut *points*. In onze eventlog ligt dit punt tussen 0 en 10 en heeft het een mediaan van 0 punten en een gemiddelde van 0,1 punten.

Data-attribuut	Aantal waarden	5 meest voorkomende waarden		
resource	148	Waarde	Absoluut	Relatief
		538	8608	0,057
		550	7935	0,053
		541	7356	0,049
		537	6931	0,046
		559	6429	0,043
vehicle_class	4	Waarde	Absoluut	Relatief
		A	146 712	0,976
		C	2456	0,016
		M	1198	0,008
		R	4	0,00002
article	66	Waarde	Absoluut	Relatief
		157	68 074	0,453
		7	42 849	0,285
		158	26 990	0,179
		142	5766	0,038
		181	1319	0,009
notification_type	2	Waarde	Absoluut	Relatief
		P	79 726	0,530
		C	134	0,0009
last_sent	3	Waarde	Absoluut	Relatief
		P	46 476	0,309
		N	30 313	0,202
		C	1440	0,010

Tabel 3: Kengetallen categorische data-attributen

Het eerste categorische data-attribuut, *resource*, geeft weer door welke medewerker de boete behandeld wordt. Dit data-attribuut heeft 148 verschillende waarden. Een resource wordt gekenmerkt door een numerieke waarde en resource 538 is de meest frequente resource, namelijk 8608 cases worden behandeld door resource 538. Het type auto waarmee de overtreding begaan is, wordt weergegeven door *vehicle_class*. Er zijn vier verschillende waarden mogelijk, [A,C,M,R], maar de meest voorkomende waarde is A. Het data-attribuut *article* geeft het

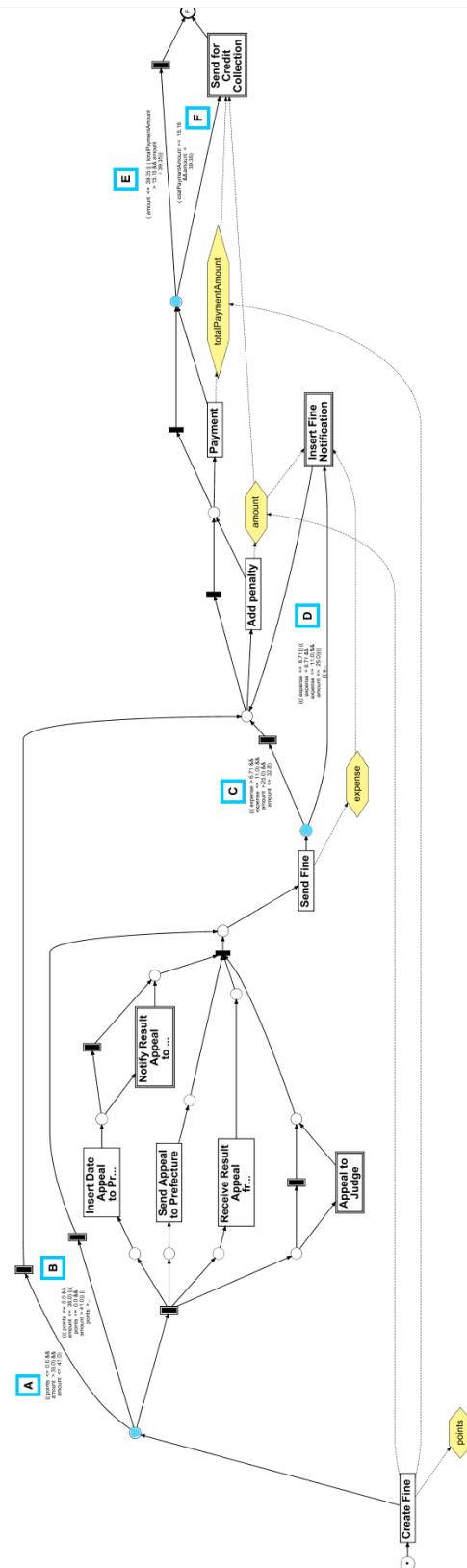
overtreden wetsartikel weer. In de eventlog vinden we 66 verschillende wetsartikelen terug en het wetsartikel dat het vaakst overtreden wordt, is artikel 157. De data-attributen *notification_type* en *last_sent* beschrijven, respectievelijk, hoe en wanneer de boete is overgebracht naar de overtreder. P is de meest voorkomende waarde voor het data-attribuut *notification_type* met een absolute frequentie van 79 726, terwijl de waarde C slechts 134 keer voorkomt in de eventlog. Voorts, krijgen 70 510 cases geen waarde toegekend voor dit data-attribuut. Ten slotte, hebben 46 476 cases de waarde P voor het data-attribuut *last_sent*. De waarde N komt 30 313 keer voor in de eventlog en de waarde C slechts 1 440 keer. 72 141 cases hebben geen waarde voor dit data-attribuut.

4.2.2 Procesmodel

De context-aware alignment methode vereist, naast een eventlog, ook een procesmodel met beslissingslogica. Er is geen normatief model bijbehorend aan de eventlog terug te vinden. Daarom zijn we genoodzaakt om verder te bouwen op een Process Discovery algoritme. Op basis van de eventlog bouwt dit algoritme een procesmodel dat het echte gedrag weergeeft. We hanteerden de 'Mine Petri net with Inductive Miner' ProM plug-in [9]. Deze plug-in produceert een Petrinet.

Echter, vereist de context-aware methode een procesmodel met beslissingslogica. Dit dwarde ons in de richting van het onderzoeksgebied 'Data aware process mining'. De data-attributen in een eventlog worden meegenomen in de ontwikkeling van een procesmodel. Dit idee is verwerkt in de 'Discovery of the Process Data-Flow (Decision-Tree Miner)' ProM plug-in [5]. Deze plug-in vereist een Petrinet en een eventlog als inputs. Op basis van de data-attributen in de eventlog gaat deze plug-in het Petrinet, bekomen in de vorige stap, aanvullen met beslissingslogica. De output, een Petrinet met beslissingslogica, is terug te vinden in Figuur 7.

Figuur 7 beschrijft een proces over het uitschrijven van verkeersboetes. Het proces start bij het aanmaken van een boete (i.e. *Create Fine*) waarbij de data-attributen *points*, *amount* en *totalPaymentAmount* een waarde krijgen toegekend. Vervolgens kan men ervoor kiezen om de boete direct te betalen, waardoor men extra administratiekosten (*expense*) vermijdt. In het andere geval wordt de boete opgestuurd (i.e. *Send Fine*) en worden er administratiekosten (*expense*) aangerekend. In een laatste geval kan men in beroep gaan tegen de vastgestelde overtreding. In dit geval wordt het bezwaarschrift naar het juiste departement gestuurd (i.e. *Send Appeal to Prefecture*). Het moment van ontvangst van het bezwaarschrift wordt bijgehouden (i.e. *Insert Date Appeal to Prefecture*). Het onderzoek van het bezwaarschrift valt buiten de scope van dit proces en enkel de resultaten van de aanvechting worden meegedeeld aan de stedelijke dienst (i.e. *Receive Result Appeal from Prefecture*). Ook de overtreder wordt op de hoogte gebracht van de resultaten (i.e. *Notify Result Appeal to Offender*). Echter, wordt deze activiteit in realiteit soms vergeten, aangezien er ook een silent transition mogelijk is. Indien gewenst, kan de overtreder nog in beroep gaan tegen dit besluit (i.e. *Appeal to Judge*). Na het verzenden van de boete (i.e. *Send Fine*),



Figuur 7: Petrinet met beslissingslogica

wordt in sommige gevallen het moment dat de overtreder de boete ontvangt (i.e. *Insert Fine Notification*) geregistreerd. Indien de boete niet betaald is binnen een redelijke termijn, wordt er een extra kost aangerekend (i.e. *Add Penalty*) en wordt het data-attribuut *amount* geüpdatet. Normaliter volgt er binnen een redelijke termijn een betaling (i.e. *Payment*). Als de overtreder alsnog niet betaalt, wordt de boete naar een krediet incassobureau gestuurd (i.e. *Send for Credit Collection*) en worden de data-attributen *amount* en *totalPaymentAmount* meegegeven ter informatie.

Beslissingspunt	Pad	Beslissingslogica
1	A	((points <= 0,0 && amount >38) && amount <=41)
1	B	((((points <= 0,0 && amount <= 38) (points <= 0,0 && amount >41)) points >0)
2	C	((expense >6,71 && expense <= 11) && amount >25) && amount <= 32,8)
2	D	((expense <= 6,71 ((expense >6,71 && expense <=11) && amount <= 25)) ((expense >6,71 && expense <= 11) && amount >32,8)) expense >11)
3	E	(amount <= 39,35 (totalPaymentAmount >15,16 && amount >39,35))
3	F	(totalPaymentAmount <= 15,16 && amount >39,35)

Tabel 4: Volledige beslissingslogica

In het Petrinet in Figuur 7 zien we op drie beslissingspunten beslissingslogica terugkomen. Deze beslissingspunten zijn blauw gearceerd en bij elke beslissingslogica staat een letter die het pad kenmerkt. Omdat de beslissingslogica niet altijd volledig wordt weergegeven in Figuur 7, noteren we in Tabel 4 per beslissingspunt en per pad de volledige beslissingslogica. De symbolen (&&) en (||) betekenen, respectievelijk AND en OR. We zien de data-attributen *points*, *expense*, *amount* en *totalPaymentAmount* terugkomen in de beslissingslogica.

4.3 Resultaten

Allereerst, bekijken we de verschillen in de kostmaatstaf onder de cost-based alignment methode van Adriansyah et al. (2013) [2] en onder onze context-aware alignment methode op niveau van de volledige test set. De kost wordt bepaald door het aantal afwijkingen tussen de trace en het modelpad. De gemiddelde kost over alle cases heen wordt weergegeven in Tabel 5. De kostmaatstaf bestaat uit de kost van log moves (LM) en de kost van model moves (MM).

Cost-based alignment methode			Context-aware alignment methode		
Gem. LM	Gem. MM	Gem. Kost	Gem. LM	Gem. MM	Gem. Kost
2,86	0,15	3,01	3,05	0,76	3,81

Tabel 5: Resultaten kostmaatstaf op log niveau

We zien dat de gemiddelde kost 26,58% hoger ligt onder de context-aware alignment methode. In absolute getallen is dit een verschil in de kostmaatstaf van 0,8 tussen de cost-based en de context-aware alignment methode. Wanneer we meer in detail gaan kijken naar de individuele veranderingen in de gemiddelde kost van log moves (LM) en model moves (MM), zien we respectievelijk een procentuele stijging van 6,64% en 406,67%. Vooral de procentuele stijging van 406,67% in de kost van model moves is opmerkelijk. Een verklaring hiervoor is het feit dat er, in beide methoden, achterliggend een gewicht van 10 000 wordt toegekend aan de kost van een model move en slechts 1 aan die van een log move. Impliciet betekent dit dat de alignment berekening wordt gestuurd in het te allen tijde vermijden van een model move, aangezien hier een gewicht van 10 000 aan toegekend wordt. Het vermijden van een model move is gemakkelijker in de cost-based alignment methode, aangezien hier het aantal mogelijke modelpaden groot is. In de context-aware alignment methode ligt dit aantal veel lager, omdat het procesmodel vereenvoudigd is. Om achteraf de vergelijking tussen de kost van log moves en model moves evenredig te houden, hebben we de kost van model moves gedeeld door 10 000. Samenvattend, zijn deze eerste resultaten reeds een indicatie dat het meenemen van beslissingslogica in je alignment methode wel degelijk een verschil uitmaakt.

Wanneer we, vervolgens, de resultaten op sublog niveau bekijken, zien we voor sommige sublogs geen verschil tussen de twee alignment methoden en voor sommige sublogs wel. De gemiddelde kost, opnieuw opgesplitst in log moves (LM) en model moves (MM), per sublog zijn terug te vinden in Tabel 6.

Sublog	Cost-based methode			Context-aware methode		
	Gem. LM	Gem. MM	Gem. Kost	Gem. LM	Gem. MM	Gem. Kost
1	3,4	0,4	3,8	3	1	4
2	2,2	0,2	2,4	2,1	0,3	2,4
3	4,7	0	4,7	4,7	0	4,7
4	2,4	0	2,4	2,6	1	3,6
5	2	0	2	2	2	4
6	3	0	3	4	1	5

Tabel 6: Resultaten kostmaatstaf op sublog niveau

Voor sublog 1, 4, 5 en 6 zien we telkens een hogere gemiddelde kost onder de context-aware alignment methode. De procentuele stijgingen bedragen respectievelijk, 5,26%, 50%, 100% en 66,67%. Voor deze sublogs maakt het dus wel degelijk een verschil uit als men de beslissingslogica meeneemt in de alignment methode of niet. Voor sublog 2 en 3 zien we dezelfde gemiddelde kost onder de cost-based alignment methode als onder de context-aware alignment methode.

De inclusie van de beslissingslogica heeft in deze sublogs dus geen invloed op de gemiddelde kostmaatstaf. Echter, zien we voor sublog 2 wel een andere verhouding in de kost van log moves en model moves terugkomen in de context-aware alignment methode namelijk een hogere model move kost en een lagere log move kost. Beide methoden produceren dezelfde kostmaatstaf voor deze sublogs, maar mogelijk produceren de methoden verschillende alignments. Een verdere analyse op alignment niveau is vereist.

Wanneer we de resultaten op alignment niveau bekijken, stellen we vast dat de cost-based alignment methode voor 56 van de 100 cases niet in staat is om de alignment uit te voeren met het correcte modelpad. De context-aware alignment methode, daarentegen, kan dit wel voor alle 100 cases omdat we rekening houden met de beslissingslogica. In dat opzicht produceert de context-aware alignment methode dus een meer correcte alignment. Echter, kan dit wel resulteren in een hogere kost, wat maakt dat de cost-based alignment methode de werkelijke kost eigenlijk onderschat. In ons experiment onderschat de cost-based alignment methode de kostmaatstaf in 39 van de 100 cases. Een praktisch voorbeeld uit ons experiment wordt weergegeven in Figuur 8 en 9, waar respectievelijk de optimal alignment onder de cost-based en onder de context-aware alignment methode wordt afgebeeld. De 'Tau' weerspiegelt een silent transition i.e. er wordt geen activiteit uitgevoerd.

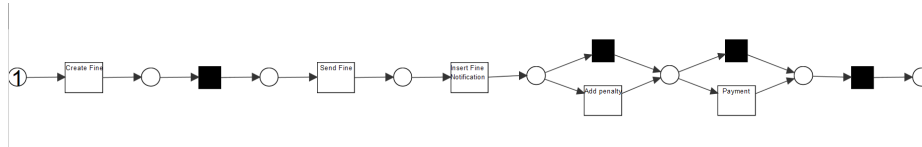
Log	Create Fine	>>	Send Fine	Insert Fine Notification	Add Penalty	>>	Send For Credit Collection
Model	Create Fine	Tau	Send Fine	Insert Fine Notification	Add Penalty	Tau	Send For Credit Collection

Figuur 8: Optimal Alignment Cost-based Methode

Log	Create Fine	>>	Send Fine	Insert Fine Notification	Add Penalty	Send For Credit Collection	>>	>>
Model	Create Fine	Tau	Send Fine	Insert Fine Notification	Add Penalty	>>	Tau	Tau

Figuur 9: Optimal Alignment Context-aware Methode

Beide alignments aligneren de trace (Create Fine, Send Fine, Insert Fine Notification, Add Penalty, Send For Credit Collection) uit de log met hun bijbehorend model. In Figuur 8 zal de cost-based alignment methode aligneren met het volledige procesmodel in Figuur 7. De context-aware alignment methode in Figuur 9, daarentegen, zal aligneren met een vereenvoudigd procesmodel. Volgens de beslissingslogica weten we dat dit vereenvoudigd procesmodel eruitziet zoals Figuur 10.



Figuur 10: Vereenvoudigd procesmodel bijbehorend aan Figuur 9

Wanneer we de optimal alignment van de cost-based methode in Figuur 8 bekijken, zien we een kost van 2 en het modelpad, waarmee de alignment wordt uitgevoerd, is (Create Fine, Send Fine, Insert Fine Notification, Add Penalty, Send For Credit Collection). Echter, is dit een foutief modelpad want dit modelpad komt niet overeen met de modelpaden uit Figuur 10. De activiteit *Send For Credit Collection* mag niet uitgevoerd worden volgens de beslissingslogica en toch maakt deze activiteit deel uit van het modelpad onder de cost-based alignment methode. De context-aware alignment methode houdt hier rekening mee en aligneert dus wel met het correcte modelpad namelijk (Create Fine, Send Fine, Insert Fine Notification, Add Penalty) in Figuur 9. Deze optimal alignment resulteert in een hogere kost namelijk 4, maar produceert een correctere alignment gegeven de context. De cost-based alignment methode is dus te optimistisch en onderschat de kostmaatstaf in dit geval met 2.

5 Conclusie

Het doel van de alignment methode zou moeten zijn om de trace te vergelijken met het modelpad dat men in realiteit had moeten volgen. Om te weten welk modelpad men in realiteit had moeten volgen, moet men rekening houden met de beslissingslogica. De alombekende cost-based alignment methode van Adriansyah et al. (2013) [2] houdt geen rekening met de beslissingslogica en aligneert daardoor niet altijd met het correcte modelpad. Bijgevolg wordt de kostmaatstaf onderschat en geeft deze methode een te optimistisch beeld weer. Door de inclusie van beslissingslogica op basis van data-attributen, aligneert de context-aware alignment methode wel met het correcte modelpad. Deze nieuwe methode produceert een correcter resultaat en gaat daarom soms gepaard met een hogere kost.

Uit een vergelijkende analyse tussen de cost-based alignment methode en de context-aware alignment methode is gebleken dat het meenemen van beslissingslogica in je alignment methode wel degelijk een impact heeft. We stelden vast dat de cost-based alignment methode in 56% van de gevallen traces aligneert met een foutief modelpad. Bijgevolg wordt de kostmaatstaf in 39% van de gevallen onderschat en kan de cost-based alignment methode bekeken worden als een te optimistische methode. Deze resultaten dienen wel genuanceerd te worden aangezien we geen zekerheid hebben dat de beslissingslogica correct is en deterministisch beschouwd mag worden. De context-aware alignment methode zal altijd aligneren met het correcte modelpad, ondanks dat hier een hogere kost

mee gepaard gaat. De focus van de cost-based alignment methode ligt op het minimaliseren van de kostmaatstaf, terwijl de focus van de context-aware methode eerst ligt op het aligneren met het correcte modelpad en dan pas op het minimaliseren van de kost.

Voorts, zijn er enkele beperkingen verbonden aan de context-aware alignment methode in deze paper. Ten eerste kan het zijn dat niet alle beslissingspunten in je procesmodel beslissingslogica bevatten. Bijgevolg kan dit beslissingspunt niet vereenvoudigd worden. Dit zorgt ervoor dat men niet één specifiek modelpad kan aanduiden als het pad dat men in realiteit had moeten volgen, aangezien er nog steeds meerdere modelpaden mogelijk zijn in het vereenvoudigd procesmodel. Echter, hebben we het procesmodel wel nog steeds enorm kunnen vereenvoudigen. Een tweede beperking omschrijft de situatie waar wel beslissingslogica is gegeven, maar essentiële data-attributen worden niet bijgehouden in de eventlog. Ook dan kunnen we het procesmodel niet vereenvoudigen en kunnen we niet één specifiek modelpad aanduiden als het pad dat men in realiteit had moeten volgen. Echter, is dit geen onrealistische situatie, aangezien ontbrekende data in realiteit nog steeds een vaakvoorkomend probleem is.

Ten slotte, omvat deze paper een uitbreiding op de klassieke cost-based alignment methode door de inclusie van beslissingslogica. De nieuwe context-aware alignment methode staat gedefinieerd in pseudo-code in Algorithm 1. Het coderen en automatiseren van dit algoritme vereist verder onderzoek. Alsook het verder analyseren van de context-aware alignment methode ten opzichte van de cost-based alignment methode op een grotere schaal en met een andere vergelijkingsbasis vereist verder onderzoek. Een grote uitdaging ligt in de alignment berekening. Onze alignments worden gestuurd in de richting van het te allen tijde vermijden van een model move, maar dit oefent een serieuze invloed uit op de alignment berekening en bijgevolg ook op de alignment vergelijking tussen de cost-based en context-aware alignment methode.

Referenties

1. Arya Adriansyah, Jorge Munoz-Gama, Josep Carmona, Boudewijn F. van Dongen, and Wil MP van der Aalst. Alignment based precision checking. In *Business Process Management Workshops: BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers 10*, pages 137–149. Springer.
2. Arya Adriansyah, Boudewijn F. van Dongen, and Wil MP van der Aalst. Memory-efficient alignment of observed and modeled behavior. 3:1–44. BPM Center Report.
3. Massimiliano de Leoni, Jorge Munoz-Gama, Josep Carmona, and Wil MP van der Aalst. Decomposing alignment-based conformance checking of data-aware process models. In *On the Move to Meaningful Internet Systems: OTM 2014 Conferences: Confederated International Conferences: CoopIS, and ODBASE 2014, Amantea, Italy, October 27-31, 2014, Proceedings*, pages 3–20. Springer.
4. Massimiliano De Leoni and Wil MP Van Der Aalst. Aligning event logs and process models for multi-perspective conformance checking: An approach based on integer linear programming. In *Business Process Management: 11th International Conference, BPM 2013, Beijing, China, August 26-30, 2013. Proceedings*, pages 113–129. Springer.

5. Massimiliano De Leoni and Wil MP Van Der Aalst. Data-aware process mining: discovering decisions in processes using alignments. In *Proceedings of the 28th annual ACM symposium on applied computing*, pages 1454–1461.
6. Massimiliano De Leoni, Wil MP Van Der Aalst, and Boudewijn F. Van Dongen. Data-and resource-aware conformance checking of business processes. In *Business Information Systems: 15th International Conference, BIS 2012, Vilnius, Lithuania, May 21-23, 2012. Proceedings 15*, pages 48–59. Springer.
7. Sebastian Dunzer, Matthias Stierle, Martin Matzner, and Stephan Baier. Conformance checking: a state-of-the-art literature review. In *Proceedings of the 11th international conference on subject-oriented business process management*, pages 1–10.
8. Marie Koorneef, Andreas Solti, Henrik Leopold, and Hajo A. Reijers. Automatic root cause identification using most probable alignments. In *Business Process Management Workshops: BPM 2017 International Workshops, Barcelona, Spain, September 10-11, 2017, Revised Papers 15*, pages 204–215. Springer.
9. Sander JJ Leemans, Dirk Fahland, and Wil MP Van Der Aalst. Discovering block-structured process models from event logs containing infrequent behaviour. In *Business Process Management Workshops: BPM 2013 International Workshops, Beijing, China, August 26, 2013, Revised Papers 11*, pages 66–78. Springer.
10. Felix Mannhardt, Massimiliano De Leoni, Hajo A. Reijers, and Wil MP Van Der Aalst. Balanced multi-perspective checking of process conformance. 98:407–437. *Computing*.
11. Anne Rozinat and Wil MP Van der Aalst. Conformance checking of processes based on monitoring real behavior. 33(1):64–95. *Information Systems*.
12. Anne Rozinat and Wil MP van der Aalst. Decision mining in ProM. 4102:420–425. *Business process management*.
13. Wil MP van der Aalst and Josep Carmona. *Process mining handbook*. Springer Nature.
14. Matthias Weidlich, Artem Polyvyanyy, Nirmal Desai, Jan Mendling, and Mathias Weske. Process compliance analysis based on behavioural profiles. 36(7):1009–1025. *Information Systems*.