BEDRIJFSECONOMISCHE WETENSCHAPPEN

2010
2011

master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica: informatie- en communicatietechnologie

Masterproef

Een process mining case study

Promotor :
Prof. dr. Koenraad VANHOOF

Niels Vandenreyt

Masterproef voorgedragen tot het bekomen van de graad van master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica , afstudeerrichting informatie- en communicatietechnologie

universiteit hasselt
UNIVERSITEIT VAN DE TOEKOMST

Universiteit Hasselt | Campus Diepenbeek | Agoralaan Gebouw D | BE-3590 Diepenbeek
Universiteit Hasselt | Campus Hasselt | Martelarenlaan 42 | BE-3500 Hasselt

universiteit hasselt
UNIVERSITEIT VAN DE TOEKOMST

2010
2011

BEDRIJFSECONOMISCHE WETENSCHAPPEN

master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica: informatie- en communicatietechnologie

Masterproef

Een process mining case study

Promotor :
Prof. dr. Koenraad VANHOOF

Niels Vandenreyt

Masterproef voorgedragen tot het bekomen van de graad van master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica , afstudeerrichting informatie- en communicatietechnologie

universiteit hasselt
UNIVERSITEIT VAN DE TOEKOMST

Woord vooraf

Deze masterproef vorm het einde van mijn masteropleiding Handelsingenieur in de Beleidsinformatica met als afstudeerrichting ICT aan de Universiteit Hasselt te Diepenbeek. Dankzij de hulp en steun van een aantal personen ben ik er in geslaagd om dit project tot een goed einde te brengen.

Graag zou ik mijn promotor, Prof. dr. Koen Vanhoof, willen bedanken voor zijn deskundige begeleiding, aanbevelingen en hulp bij het tot stand brengen van deze masterproef.

Daarnaast zou ik ook de heer George Sammour willen bedanken voor zijn hulp bij het aanleveren van de gebruikte dataset. Verder wil ik ook mijn broer Jens Vandenreyt bedanken voor zijn hulp bij het kleine stukje programmeerwerk.

Tenslotte wil ik nog mijn familie bedanken voor de steun die ze mij hebben gegeven doorheen mijn opleiding.

Samenvatting

Informatiesystemen nemen in de huidige bedrijfswereld een prominente plaats in bij de dagelijkse bedrijfsvoering. Ze ondersteunen de aaneenschakeling van activiteiten die uitgevoerd moeten worden om waarde te creëren voor klanten. De meeste van deze systemen zijn ondernemingsbrede informatiesystemen die geprogrammeerd zijn zodat ze het complete proces beheersen en taken aan de verschillende betrokkenen kunnen uitdelen. Omwille van de complexiteit van deze process-aware informatiesystemen zijn er echter weinig gebruikers die een duidelijk en volledig beeld van het complete proces hebben. Business Process Management (BPM) en meer in het bijzonder Business Process Analysis (BPA) zijn belangrijke hulpmiddelen om de complexiteit van de systemen te vatten en te beheersen. Een onderdeel van BPA is process mining. De idee bij process mining is om informatie over processen te vergaren uit event logs die worden bijgehouden door process-aware informatiesystemen. Event logs zijn lijsten waarin alle handelingen die worden uitgevoerd op een bepaald bedrijfsobject in het systeem worden geregistreerd. Om succesvol informatie uit zo een event log te halen werd het ProM-framework ontworpen. Dit framework accepteert echter enkel event logs in een universeel formaat, XES. Aangezien ieder informatiesysteem event logs op een systeemspecifieke manier registreert werd het XESame-framework ontwikkeld om de conversie naar XES uit te voeren. In deze thesis wordt een systeemspecifieke event log met behulp van XESame omgevormd tot een event log in het XES-formaat en klaargestoomd om process mining op toe te passen. Tegenwoordig blijven process-aware informatiesystemen niet enkel meer beperkt tot de bedrijfsomgeving maar vinden ze ook ingang in het dagelijkse leven. Ziekenhuizen, de overheid, onderzoeksinstellingen e.d. kunnen allemaal profiteren van de implementatie van zulke systemen. De data die gebruikt wordt in de case study in deze thesis is afkomstig van een mobiliteitsonderzoek aan de UHasselt. Maar alvorens de conversie van de event log uitgevoerd wordt, informeert deze thesis aan de hand van een literatuurstudie over de oorsprong, het doel en de werking van process

mining en worden de verschillende aspecten besproken die een invloed hebben op de definitie van de conversie. Vervolgens wordt een scenario uitgewerkt dat als process mining kader zal dienen om de conversiedefinitie te bepalen. Na het uitvoeren van de conversie zal de verkregen event log geanalyseerd worden en getest om de werking van XESame te valideren.

Inhoudsopgave

Deel I: Introductie

Hoofdstuk 1: Onderzoeksplan

1.1 Inleiding

Het gebruik van IT is in de huidige bedrijfswereld niet meer weg te denken. De introductie van IT zorgde voor een drastische wijziging in de bedrijfsvoering van ondernemingen. In een eerste fase werden steeds efficiëntere IT-toepassingen ontwikkeld om de verwerking van alledaagse transacties te automatiseren. Vandaag gebruiken vele bedrijven geïntegreerde ondernemingsbrede informatiesystemen om bedrijfsprocessen te coördineren [1].

Het bekendste voorbeeld van deze systemen is Enterprise Resource Planning. ERP-systemen zorgen voor de afhandeling van bijna alle subprocessen van de bedrijfsvoering. Aan de hand van Workflow Managementtechnieken ondersteunen ze zowel het financieel, human resource, customer relationship en supply chain management van de onderneming [2]. Het systeem, een Process-Aware Information System (PAIS), moet zich bijgevolg bewust zijn van het totale bedrijfsproces om taken te kunnen delegeren aan groepen van gebruikers uit de verschillende functionele divisies van het bedrijf. Alle transacties die worden uitgevoerd worden bijgehouden in een soort log of geschiedenis [1,3].

De complexe aard van deze ondernemingsbrede systemen zorgt ervoor dat weinig gebruikers een compleet en gedetailleerd overzicht hebben van de werking van het totale bedrijfsproces. Maar zelfs als het bedrijfsproces duidelijk gedocumenteerd is wijkt de werkelijke uitvoering dikwijls af van het gemodelleerde proces [2]. Deze kloof kan problemen opleveren bij de noodzakelijke monitoring van het proces die wordt opgelegd door de wetgever en de aandeelhouders. Zij eisen een adequate corporate governance structuur met transparante bedrijfsprocessen en de constante evaluatie

ervan. Bovendien is er ook een voortdurende druk om de bedrijfsprocessen te verbeteren om competitief te blijven in de huidige concurrentiële markten. Het is dan ook aannemelijk om bij het wegwerken van deze kloof te starten met een grondige analyse van het bestaande bedrijfsproces [1,3].

Business process mining heeft o.a. als doel om via reverse engineering een procesmodel op te bouwen dat enkel gebaseerd is op de informatie die vervat zit in de event log van het informatiesysteem. Een event log geeft een overzicht van alle handelingen die gebeuren op een specifiek business object plus wanneer en door wie deze handelingen werden uitgevoerd. Andere process mining analyse technieken gebruiken deze logs om controle, data, organisatie en sociale structuren te ontdekken in het informatiesysteem. Om de implementatie van al deze verschillende technieken te bundelen is het ProM framework ontwikkeld [1,3].

Process mining technieken, en dus ook het ProM-framework, kunnen enkel succesvol worden uitgevoerd als er een zuivere event log voor handen is. Die event log moet zijn opgesteld in een formaat waar het framework mee overweg kan. Het ProM-framework ondersteunt zowel het MXML als het XES event log formaat. Maar hoewel veel systemen event logs produceren in één of andere vorm, bevatten deze event logs vaak niet al de vereiste data en gebruiken de meeste systemen hun eigen event log formaat. Dat wil echter niet zeggen dat de vereiste data, omdat ze niet onmiddellijk zichtbaar is in de systeemspecifieke event log, niet aanwezig is in de dataopslag van het informatiesysteem. Events die gebeuren voor specifieke business objecten, en bij uitbreiding dus ook event logs van het hele proces, kunnen worden gereproduceerd uit de dataomgeving van het systeem [1].

1.2 Probleemstelling

Zoals reeds uitgelegd in de inleiding is de event log die nodig is voor process mining niet altijd beschikbaar in het vereiste formaat. De vereiste data is echter steeds nog onzichtbaar aanwezig in de dataopslag van het systeem zodat het mogelijk is om event logs te reproduceren in het juiste formaat.

Het omzetten van ruwe data naar event logs bestaat natuurlijk al langer en grote stappen zijn reeds genomen voor de automatisering ervan met de ontwikkeling van verschillende importframeworks zoals het ProM Import Framework. Dit programmeerframework ondersteunt het converteren van data naar het MXML event log formaat aan de hand van plug-ins die geschreven worden in Java. Het framework is reeds zeer nuttig gebleken in het converteren van logs uit echte bedrijfssystemen naar MXML event logs, maar het heeft ook enkele beperkingen. Eén beperking is de assumptie dat het formaat van de inputdata al een soort van event log is. De mogelijkheid dat de inputdata verspreid kan zijn opgeslagen over enkele gerelateerde tabbellen is niet in rekening gebracht. Bovendien wordt process mining voornamelijk uitgevoerd door business analisten met voldoende kennis in het domein van de bedrijfsprocessen. Zij beschikken dikwijls echter niet over de benodigde programmeerkennis om nieuwe conversies te definiëren volgens de structuur van het framework [1].

Om deze problemen op te lossen ontwikkelde ing. J. Buijs van de Technische Universiteit Eindhoven een nieuw conversie framework en een applicatieprototype XESMa in het kader van zijn master thesis. Het doel van het project was om een applicatie te ontwikkelen die business analisten in staat stelt om conversies te definiëren en uit te voeren met zo weinig programmeerkennis als mogelijk. De applicatie kan data in tabelformaat inlezen en event logs creëren in het XES formaat en onder bepaalde restricties ook in het MXML formaat [1].

Aangezien de XESMa applicatie (2010) nog maar een prototype is, zijn er tot nu toe maar relatief weinig case studies uitgevoerd om de validiteit van de applicatie te testen. Buijs voerde zelf enkele conversies uit in zijn master thesis maar het is interessant en noodzakelijk om de werking van de applicatie te valideren aan de hand van enkele onafhankelijke testcases.

1.3 Centrale onderzoeksvraag

Het doel van dit onderzoek is het vermogen van de XESMa-applicatie om adequaat conversies te definiëren en uit te voeren te beoordelen.

De onderzoeksvraag luidt dan ook:

Hoe kan men aan de hand van een onafhankelijke case study de werking van het XESMa-framework valideren?

1.4 Deelvragen

Het antwoord op de centrale onderzoeksvraag kan men onderbouwen met het onderzoeken van de volgende deelvragen:

- *Wat zijn de verschillende aspecten waar rekening mee moet worden gehouden bij het definiëren van een conversiedefinitie en hoe beïnvloeden deze aspecten de conversiedefinitie?*
- *Hoe kan men deze aspecten gebruiken om conversiescenario's op te stellen om de validiteit van het framework te testen?*
- *Hoe kan men de validiteit van de resultaten van de uitgevoerde conversiedefinities meten?*

1.5 Opbouw van het onderzoek

Om het doel van het project te bereiken en de onderzoek- en deelvragen te beantwoorden zal het onderzoek worden gevoerd in meerdere fasen.

In een eerste fase zal er een verkennende literatuurstudie worden uitgevoerd. Deze literatuurstudie zal een theoretisch kader worden waarin verschillende relevante concepten worden onderzocht. Vervolgens zal er onderzocht worden welke aspecten een invloed hebben op de definitie van de conversie en hoe ze de conversiedefinitie juist beïnvloeden. Er wordt ook onderzocht op welke manier de uitgevoerde conversies kunnen worden beoordeeld om het framework te valideren.

In de volgende fase zal er een experimenteel onderzoek worden gevoerd waarin verschillende scenario's voor conversiedefinities worden opgesteld. Hierbij wordt rekening gehouden met de relevante conversie aspecten die gevonden zijn in de verkennende literatuurstudie. Na het opstellen van deze scenario's worden de verschillende conversies uitgevoerd.

De laatste fase is de analysefase waarin de uitgevoerde conversies worden beoordeeld aan de hand van de maatstaven die gevonden zijn in de verkennende literatuurstudie. Er wordt een conclusie van het experimenteel onderzoek geformuleerd om de centrale onderzoeksvraag en de deelvragen te beantwoorden.

Deel II: Literatuurstudie

Hoofdstuk 2: Process Mining

De idee bij process mining is om informatie over processen te puren uit event logs die worden bijgehouden door process-aware informatiesystemen [4]. In deze verwoording komen verschillende begrippen voor die zullen worden uitgelegd in het vervolg van dit hoofdstuk. Eerst zal worden beschreven waar process mining zich situeert binnen de brede omgeving van het informatiemanagement. Verder zal er worden besproken wat process mining juist is en hoe het werkt. Vervolgens stellen we het ProM framework en de XESame tool voor.

2.1 Situering

Business process mining is sterk gerelateerd aan Business Process Management (BPM). In [8,9] wordt BPM gedefinieerd als volgt:

> *"BPM supports business processes by using methods, techniques, and software to design, enact, control, and analyze operational processes."*

BPM is dus een collectie van methodes, technieken en software om het bedrijfsproces te ondersteunen door operationele processen te ontwerpen en te modelleren, deze modellen te implementeren in informatiesystemen en de uitvoering van de processen te monitoren en analyseren om zo de modellen te verbeteren.

Business Process Management systemen worden door vele mensen beschouwd als een uitbreiding van Process-aware Informatiesystemen (PAIS) [8]. Een process-aware informatiesysteem is een specifiek type informatiesysteem. In [4] kan worden gevonden dat S. Alter een informatiesysteem definieert als volgt:

"An information system is a particular type of work system that uses information technology to capture, transmit, store, retrieve, manipulate, or display information, thereby supporting one or more other work systems."

Een informatiesysteem gebruikt dus informatietechnologie om één of meerdere werksystemen, d.i. een systeem waarin menselijke deelnemers bedrijfsprocessen uitvoeren om producten te produceren voor interne klanten aan de hand van informatie, technologie en andere bronnen, te ondersteunen. Een informatiesysteem bestond oorspronkelijk dan ook uit verschillende lagen met hardware en software die samen een portfolio van werksystemen ondersteunden [4].

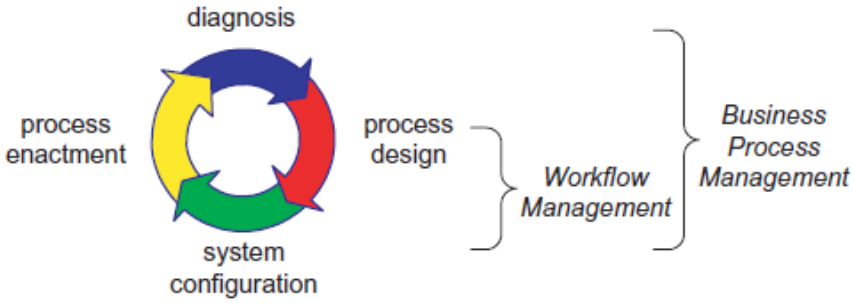
De laatste decennia zijn er echter verschillende trends merkbaar bij informatiesystemen [4]. Een eerste trend is de verschuiving van het programmeren van afzonderlijke applicaties voor de verschillende werksystemen naar de integratie van verschillende applicaties. Informatiesystemen worden dus steeds meer geassembleerd uit bestaande applicaties dia worden aangepast zodat ze een geïntegreerd geheel vormen. De tweede trend is de evolutie van een datageoriënteerde zienswijze naar een procesgeoriënteerde zienswijze. De focus van IT was aanvankelijk gericht op het opslaan, ophalen en voorstellen van data. De logica van de bedrijfsprocessen zat bijgevolg verspreid over de verschillende softwareapplicaties. Met de opkomst van managementtrends zoals BPM en Business Process Reengineering (BPR) verschoof de nadruk echter naar de processen. Een derde trend is de evolutie van een nauwgezet gepland design naar een design via het steeds herontwerpen en verbeteren van bestaande modellen.

Het resultaat van deze trends is de opkomst van een steeds groeiend aantal process-aware informatiesystemen. In [4] wordt een PAIS omschreven als:

"A software system that manages and executes operational processes involving people, applications, and/or information sources on the basis of process models."

De idee bij een PAIS is om informatiesystemen te bouwen die steunen op voorgedefinieerde procesmodellen. Idealiter wordt het procesbeheer afgezonderd in een aparte component van het informatiesysteem. Hierin wordt het bedrijfsproces gedefinieerd als een model dat kan worden opgevraagd, bekeken en aangepast zodat de processen niet star moeten worden gecodeerd in de applicatie. Deze soort generieke process-aware informatiesystemen maken gebruik van workflow technologie. Dit wil zeggen dat de software enkel de middelen aanreikt om de modellen te definiëren. De business analist moet vervolgens de operationele processen, bedrijfseenheden en gebruikers van het werksysteem opstellen zodat deze specificaties kunnen worden uitgevoerd door het informatiesysteem [4,5,6,7]. Het formeel analyseren, beschrijven en gebruiken van standaard workflowmodellen verbetert de kwaliteit van de processen en de flexibiliteit van de onderneming. Bovendien leveren de modellen een overzicht van het gehele bedrijfsproces zodat bedrijven eenvoudig de bestaande processen kunnen aanpassen of nieuwe subprocessen kunnen toevoegen indien wijzigingen in de bedrijfsomgeving dit opdringen [7].

Het zijn echter niet enkel deze zuivere Workflow Management Systemen (WfMS) die als PAIS kwalificeren. Er zijn ook systemen die op een lossere manier van workflow technologie gebruik maken. Voorbeelden hiervan zijn: Enterprise Resource Planning (ERP), Customer Relationship Management (CRM) software en Supply Chain Management (SCM) systemen. Ook dit zijn transactiesystemen die de verschillende taken van het gedefinieerde model delegeren aan de gebruikers van het systeem. Deze systemen zijn ook gebaseerd op modellen om de processen te begeleiden, maar de procesmodellen worden niet steeds volledig opgedrongen zodat de gebruikers bepaalde vrijheden hebben om ervan af te wijken [4].



Figuur 1: BPM levenscyclus [8]

Figuur 1 toont de hierboven beschreven relatie tussen Business Process Management en Workflow Management door gebruik te maken van de BPM levenscyclus. De verschillende fasen waarin de operationele processen worden ondersteund worden hierin weergegeven. In de designfase worden de processen ontworpen. Tijdens de configuratiefase worden de modellen geïmplementeerd door het PAIS te configureren. Na de configuratie start de uitvoering van de operationele processen met behulp van het zopas geconfigureerd informatiesysteem. De focus van het PAIS richt zich op deze drie fases.

In de vierde fase, de diagnosefase worden de operationele processen geanalyseerd om problemen te ontdekken en worden er verbeteringen gezocht om die problemen op te lossen. De oplossingen vormen vervolgens de basis voor het herontwerpen van het procesmodel, zodat de cyclus van vooraan kan herbeginnen. Dit deel van BPM wordt onderzocht in het subdomein Business Process Analysis (BPA) [6,8].

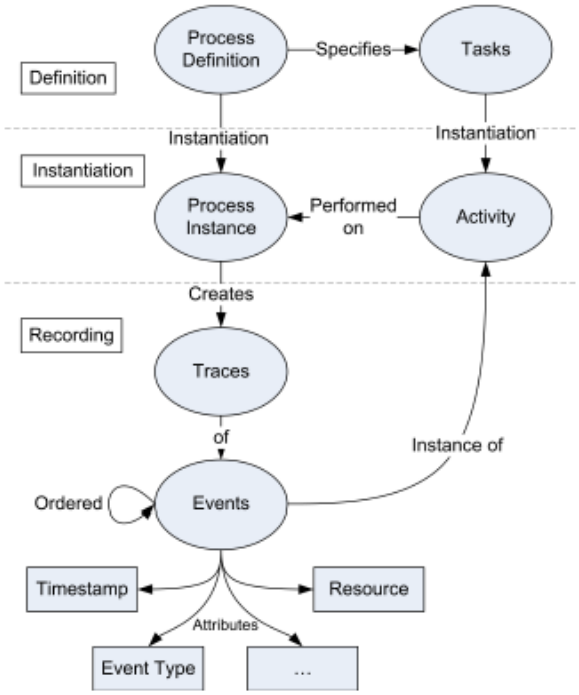
Business process mining bevindt zich in de vierde fase van de BPM levenscyclus, de diagnosefase.

2.2 Event Logs

Zoals reeds vermeld in de vorige sectie maakt een PAIS gebruik van workflow technologie. Veel van deze bedrijfsinformatiesystemen zijn transactiesystemen die alle relevante handelingen en

transacties die plaatsvinden in het systeem bewaren in een gestructureerde vorm. ERP-systemen loggen bijvoorbeeld alle transacties in verband met de aankoop en verkoop van goederen, CRM-systemen loggen dan weer alle interactie die plaatsvindt met klanten. Al deze gestructureerde lijsten zijn voorbeelden van event logs [4,6].

Figuur 2 toont de algemene structuur van een event log en de relatie ervan met de procesdefinitie. Event logs bestaan uit een opsomming van alle gebeurtenissen, of events, die zich hebben voorgedaan in het systeem. Ieder event verwijst naar één procesinstantie waarop één activiteit werd uitgevoerd. De procesinstantie is het onderwerp of voorwerp dat wordt behandeld in het systeem. Dit kan bijvoorbeeld een klantenorder zijn, of een patiënt in een ziekenhuis. De activiteit is een handeling die wordt uitgevoerd op de procesinstantie. Activiteiten voeren de taken uit die zijn gedefinieerd in de procesdefinitie [3,4,5,10,11].



Figuur 2: Event log structuur [1]

Bovenop de procesinstantie en de activiteit, die absoluut vereist zijn, kan een event nog extra informatie bevatten zoals bijvoorbeeld het tijdstip van de uitvoering, het type event of de opdrachtgever van de handeling.

Tabel 1 geeft een voorbeeld van een event log weer. Deze event log bevat 19 events die handelen over 5 verschillende procesinstanties (*case* 1 tot 5). Op deze procesinstanties worden één of meer activiteiten uitgevoerd. Er zijn 5 verschillende activiteiten (*activity* A tot E). Ieder event bevat bovendien ook nog de naam van de opdrachtgever en het tijdstip van uitvoering van de activiteit [3].

Tabel 1: Een event log [3]

case id	activity id	originator	timestamp
case 1	activity A	John	9-3-2004:15.01
case 2	activity A	John	9-3-2004:15.12
case 3	activity A	Sue	9-3-2004:16.03
case 3	activity B	Carol	9-3-2004:16.07
case 1	activity B	Mike	9-3-2004:18.25
case 1	activity C	John	10-3-2004:9.23
case 2	activity C	Mike	10-3-2004:10.34
case 4	activity A	Sue	10-3-2004:10.35
case 2	activity B	John	10-3-2004:12.34
case 2	activity D	Pete	10-3-2004:12.50
case 5	activity A	Sue	10-3-2004:13.05
case 4	activity C	Carol	11-3-2004:10.12
case 1	activity D	Pete	11-3-2004:10.14
case 3	activity C	Sue	11-3-2004:10.44
case 3	activity D	Pete	11-3-2004:11.03
case 4	activity B	Sue	14-3-2004:11.18
case 5	activity E	Clare	17-3-2004:12.22
case 5	activity D	Clare	18-3-2004:14.34
case 4	activity D	Pete	19-3-2004:15.56

Zoals reeds getoond werd in Figuur 2, blijven event logs niet beperkt tot enkel de informatie die in dit voorbeeld gegeven wordt. Allerlei soorten systeemspecifieke data kan worden opgenomen in een event log. Dit is dan ook direct één van de grootste uitdagingen in process mining. Ieder informatiesysteem heeft een eigen interne datastructuur en event logs worden niet altijd expliciet

weergegeven. Event logs zijn echter onontbeerlijk voor process mining, dus is het noodzakelijk om logs voor te stellen op een gestandaardiseerde manier [12].

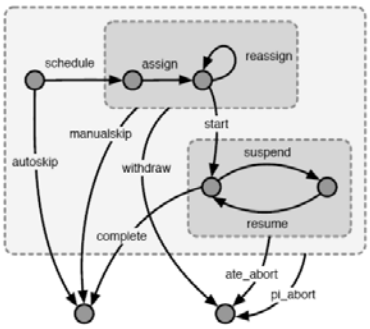
2.2.1 MXML

In 2005 introduceerden Van Dongen en Van der Aalst [12] een XML formaat voor het gestandaardiseerd opslaan van event logs. Het doel van dit formaat is om event logs zo te kunnen voorstellen dat de algoritmes van process mining zich kunnen baseren op een algemeen inputformaat.

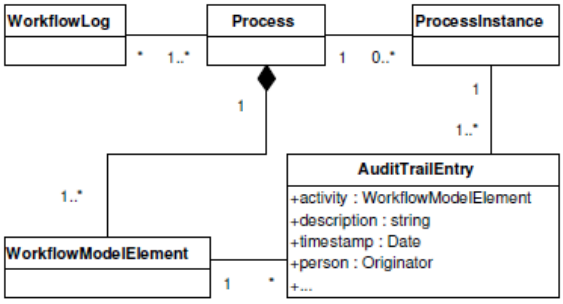
Alvorens een mining data metamodel op te stellen, stelden Van Dongen en Van der Aalst enkele vereisten op waaraan de data moest voldoen. Er moet namelijk een minimale hoeveelheid informatie beschikbaar zijn om process mining uit te voeren. De meeste vereisten zijn hierboven reeds ter sprake gekomen:

- Iedere procesinstantie hoort bij een specifiek proces.
- Ieder event verwijst naar één specifieke procesinstantie.
- Ieder event verwijst naar één specifieke activiteit.
- Ieder event is atomair. Events hebben geen tijdsduur, enkel een tijdstip van uitvoering.
- Ieder event moet een beschrijving geven van het event type.

Deze laatste vereiste komt voort uit het feit dat events geen tijdsduur aanduiden, maar activiteiten toch een tijdsduur hebben. Door in het event type de status van de activiteit aan te geven, bijvoorbeeld 'gestart' of 'gereed', kan de tijdsduur toch worden bepaald. De verschillende statussen die een activiteit gedurende haar levensloop kan aannemen worden weergegeven in het transactiemodel in Figuur 3.



Figuur 3: Transactiemodel [12]



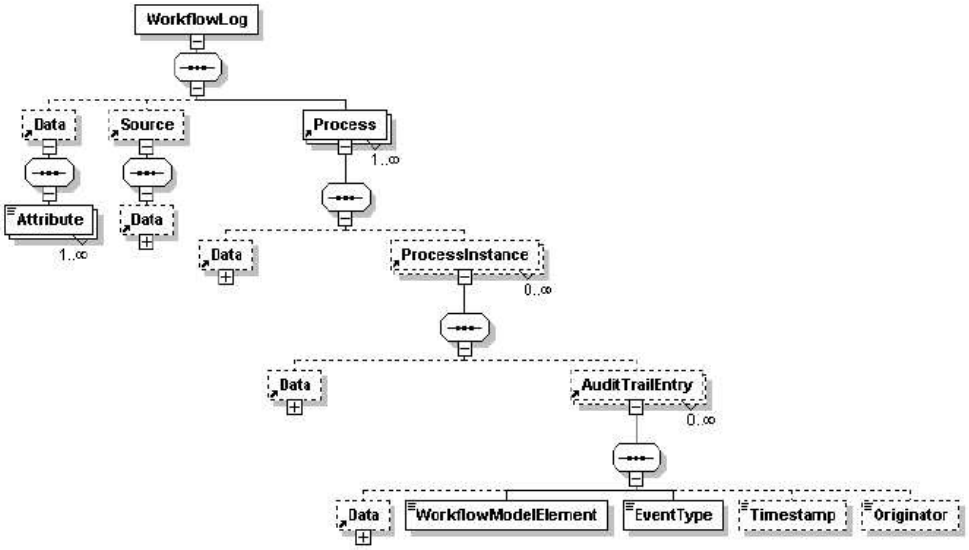
Figuur 4: UML klassendiagram metamodel [12]

Vertrekkende vanuit deze vereisten, stelden Van Dongen en Van der Aalst vervolgens een UML klassendiagram op van het metamodel (Figuur 4). Zoals kan worden afgeleid uit dit metamodel verwijst ieder event *(AuditTrailEntry)* naar één activiteit *(WorkflowModelElement)* en één procesinstantie *(ProcessInstance)*. Bovendien bevat een event een event type *(description)* en een tijdstip van uitvoering *(timestamp)* [12].

Na het opstellen van een metamodel hebben ze de XML structuur van het MXML-formaat vastgelegd in de schemadefinitie. In Figuur 5 kan men vinden dat een event log *(WorkflowLog)* steeds bestaat uit minstens één proces, vastgelegd in het element 'Process'. Een proces kan vervolgens verschillende procesinstanties bevatten die op hun beurt verschillende events kunnen bevatten. Ten opzichte van het metamodel zijn er echter twee aanpassingen uitgevoerd. Ten eerste is er een element 'Data' beschikbaar op ieder niveau. Dit element maakt het mogelijk om extra informatie op te slaan in tekstvorm en bevat steeds minstens één attribuut. De tweede aanpassing is het toevoegen van het element 'Source' als *child element* onder het niveau van de event log. Dit element kan gebruikt worden om informatie op te slaan over de herkomst van de event log [12].

Het MXML formaat is zeer nuttig gebleken als een gestandaardiseerde manier om event logs op te slaan voor het gebruik bij process mining [10,12]. Er zijn echter weinig informatiesystemen die event logs automatisch bijhouden in dit formaat. Bovendien zijn er gedurende het gebruik van het formaat verschillende problemen opgedoken. Eén belangrijk probleem komt voort uit het feit dat de extra

informatie die vervat zit in het element 'Attribute' enkel kan worden opgeslagen in zuivere tekstvorm.



Figuur 5: MXML schemadefinitie [10]

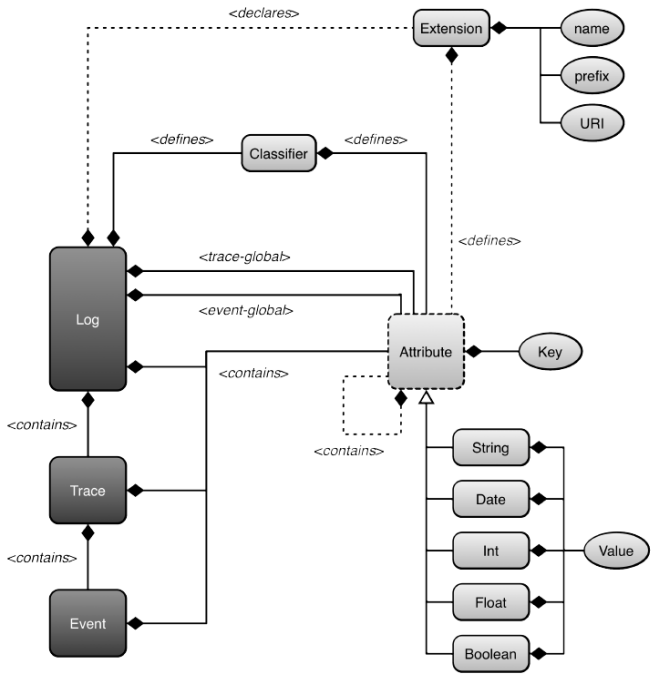
De informatie in dit element heeft dus geen semantische betekenis en kan moeilijk worden gebruikt door de process mining algoritmen. Een ander probleem is de benaming die is gegeven aan de verschillende concepten, zodat enkel strikt gestructureerde processen dit formaat konden gebruiken [1].

Om deze problemen op te lossen en een standaard te maken die door verschillende informatiesystemen rechtstreeks kan worden gebruikt werd een nieuwe standaard ontwikkeld door het IEEE Task Force Process Mining [1]. Dit nieuwe formaat heet XES, wat staat voor eXtensible Event Stream.

2.2.2 XES

In 2009 werd de definitie van het XES event log formaat versie 1.0 uitgebracht [13]. Deze definitie is gebouwd rond drie elementen: *log*-, *trace*- en *event*-objecten. Het log-element bevindt zich op het hoogste niveau en omvat de hele event log. Een log element bevat verschillende trace-elementen. Ieder trace-element is gerelateerd aan één specifieke instantie. Dit is het voorwerp of onderwerp dat wordt behandeld in het systeem. Het Nederlandstalige woord voor *trace* is spoor. In het spoor van een procesinstantie worden dus alle handelingen bijgehouden die op de respectievelijke instantie werden uitgevoerd, de zogenaamde events [1].

Geen van deze objecten bevatten echter zelf informatie, ze bepalen enkel de structuur van het document. Daarom kan ieder object één of meer attributen hebben waarin de informatie over het object vervat zit. Ieder attribuut bevat standaard een *key*, waarin de naam van het attribuut wordt weergegeven, en een *value*, waarin de waarde wordt weergegeven. Een attribuut kan van de vorm *string*, *date*, *integer*, *float* en *boolean* zijn. Het metamodel van XES wordt weergegeven in Figuur 6.



Figuur 6: XES metamodel [1]

Aangezien XES geen specifieke lijst met attributen definieert, is de betekenis van deze attributen niet ambigu. Dit probleem kan worden opgelost aan de hand van het concept 'extensions'. In [1] wordt zo een extensie verklaard als: *"a set of attributes on any level of the XES log hierarchy and in doing so provides points of reference for interpreting these attributes."* De extensiedefinities worden in de event log meegegeven in het log element. XES heeft 5 standaardextensies waarin reeds enkele attributen gedefinieerd zijn. Deze worden, samen met het elementniveau waarvoor het attribuut kan worden gebruikt, weergegeven in Tabel 2.

Tabel 2: XES standaardextensies [1]

Extension	Key	Type	Attribute Level	Description
Concept	name	string	log, trace, event	Generally understood name.
Concept	instance	string	event	Identifier of the activity whose execution generated the event.
Lifecycle	model	string	log	The transactional model used for the lifecycle transition for all events in the log.
Lifecycle	transition	string	event	The lifecycle transition represented by each event.
Organizational	resource	string	event	The name, or identifier, of the resource having triggered the event.
Organizational	role	string	event	The role of the resource having triggered the event, within the organizational structure.
Organizational	group	string	event	The group within the organizational structure, of which the resource having triggered the event is a member.
Time	timestamp	date	event	The date and time, at which the event has occurred.
Semantic	modelReference	string	log, trace, event, meta	Reference to model concepts in an ontology.

De extensie 'concept' definieert een attribuut 'name' dat gebruikt wordt om het log-, trace- of event-element een naam te geven. Ook wordt er een attribuut 'instance' bepaald op het event niveau om de activiteit van het event te identificeren en te koppelen aan de data uit de databron. In de 'lifecycle' extensie wordt het attribuut 'transition' bepaald. Dit attribuut geeft de status van de

activiteit in het event weer, zoals bijvoorbeeld gedefinieerd in het transactiemodel in Figuur 3. In het log element kan worden gedefinieerd welk model gebruikt wordt. De 'organizational'-extensie definieert drie verschillende attributen voor op het niveau 'event'. Het 'resource'-attribuut bewaart de naam van de uitvoerder van het event, 'role' bepaalt welke rol deze uitvoerder vertolkt in de organisatie en 'group' definieert de groep binnen de organisatiestructuur waarvan de uitvoerder lid is. De 'time'-extensie bepaalt een 'timestamp'-attribuut waarin het tijdstip kan worden bewaard waarop het event zich heeft voorgedaan. De 'semantic'-extensie tenslotte definieert het 'modelReference'-attribuut dat refereert naar een concept van een model in een externe ontologie [1,13].

2.3 Process mining: een overzicht

Het doel van process mining is om informatie over processen te halen uit event logs [5]. De term informatie is natuurlijk een breed begrip. Daarom worden er drie verschillende process mining perspectieven onderscheiden [4,11,12]:

- *Process* perspectief
- *Organizational* perspectief
- *Case* perspectief

Het procesperspectief richt zich op de volgorde van de activiteiten die uitgevoerd worden in het proces. Het doel van dit perspectief is om een passend model samen te stellen waarin alle mogelijke paden van activiteiten in vervat zitten. Process mining via het proces perspectief gebeurt door de vraag te stellen: "Hoe worden procesinstanties verwerkt van bij hun ontstaan tot ze worden afgesloten?" [4,11,15].

In het organisatieperspectief is het de bedoeling om te onderzoeken welke uitvoerders betrokken zijn bij de verschillende activiteiten en welke relatie deze uitvoerders onderling hebben. Algoritmes die binnen dit perspectief werkzaam zijn proberen om personen in groepen en rollen onder te verdelen naargelang de taak die iedere persoon uitvoert. Een andere mogelijkheid is om sociale netwerken te bouwen van de werknemers aan de hand van de verschillende taken die worden uitgevoerd. Dit perspectief stelt de vraag: "Wie voert welke taken uit en hoe verhouden de uitvoerders zich onderling?" [4,14].

Het laatste perspectief is het *case* perspectief. Dit perspectief richt zich op de eigenschappen van de verschillende procesinstanties. De eigenschappen van procesinstanties kunnen namelijk een invloed hebben op de manier waarop of door wie de procesinstantie in kwestie wordt behandeld. Hier wordt de vraag gesteld: "Wat heeft een invloed op behandeling van specifieke procesinstanties?" [4,14].

Orthogonaal met de hierboven besproken perspectieven kan men zich bij proces mining richten op één van de volgende twee nadrukken [4]:

- *Logical*
- *Performance*

Bij een nadruk op het logische worden event logs onderzocht met als doel een soort volgorde of classificatie in het onderzochte perspectief te vinden. Bij de nadruk op *performance* wordt de prestatie van de gekozen materie onderzocht. Terwijl de logische nadruk een kwalitatief onderzoek is, is de performance nadruk eerder een kwantitatief onderzoek.

In Tabel 3 worden enkele voorbeelden gegeven van onderzoekseigenschappen die kunnen worden onderzocht.

Tabel 3: Voorbeelden onderzoeksperspectieven en focussen [4]

Perspective	Examples of logical properties	Examples of performance properties
Process perspective	Activity A is always followed by B; activities C and D may be executed in parallel.	The average processing time of activity A is 35 minutes; activity A is executed for 80% of the cases.
Organizational perspective	John and Mary are on the same team; Pete is the manager of department D.	John handles on average 30 cases per day; Mary and Pete work together on 50% of the cases.
Case perspective	Cases of more than 5000 euros are handled by John; activity A is only executed for private customers.	80% of cases of more than 5000 euros are handled within 2 days; the average flow time of cases handled by John and Mary is 2 weeks.

2.4 In de praktijk

Sinds de opkomst van process mining worden er steeds nieuwe algoritmen ontwikkeld om informatie over één van de perspectieven van processen te verkrijgen. Naast het ontwikkelen van een process mining algoritme werd vaak ook nog een softwaretool ontworpen om het algoritme te implementeren in een bruikbare softwareapplicatie.

In 2002 en 2003 werden aan de Technische Universiteit Eindhoven de softwareapplicaties EMiT en Little Thumb ontwikkeld [16,17]. Deze twee applicaties zijn gebaseerd op het alfa-algoritme, een process mining algoritme dat kan worden geclassificeerd in het procesperspectief van process mining. Het alfa-algoritme probeert aan de hand van de inputdata in de event log een procesmodel te ontdekken en stelt dit model vervolgens voor in de vorm van een Petri-net [4].

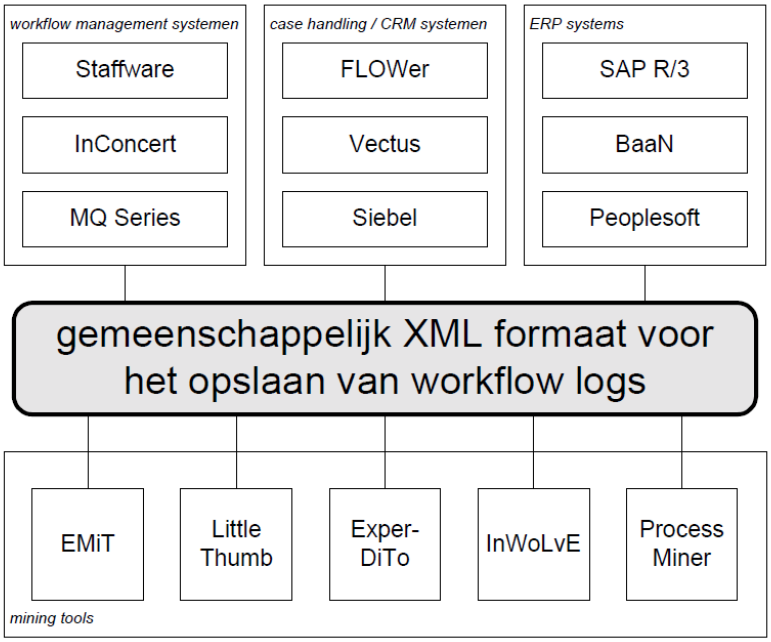
In 2004 werd vervolgens een andere tool, MiSon, ontworpen om het organisatieperspectief van process mining te onderzoeken [18,19]. Deze tool is gebaseerd op concepten die afkomstig zijn van Social Network Analysis (SNA). De applicatie probeert op basis van een event log sociogrammen of

matrices te construeren die de sociale netwerken binnen het systeem van de processen moeten voorstellen.

Het voordeel van deze aanpak, waarbij ieder algoritme een eigen softwareapplicatie heeft, is dat de transitie van de ruwe outputdata uit een PAIS naar een gestroomlijnde event log die als basis moet dienen voor process mining kan worden uitgevoerd door de applicatie zelf. De applicatie is geprogrammeerd om uit de output van enkele bekende types PAIS, die als input dient voor de applicatie, de juiste data te halen en om te vormen tot een event log met alle benodigde data die het algoritme vereist. Iedere applicatie weet dus welke data het nodig heeft en waar deze data zich bevindt in de input.

De verschillende formaten en verschillende technieken voor process mining leidden echter tot een complexe situatie die de ontwikkeling van meer praktische toepassingen in de weg stond. De ontwikkeling van een gemeenschappelijk XML-formaat (MXML en later XES) voor het opslaan van event logs in 2005 was een eerste stap in de harmonisering van het process mining landschap [20]. In Figuur 7 wordt getoond hoe het systeemonafhankelijke XML-formaat operationele informatiesystemen en *mining tools* ontkoppelt zodat onnodige conversielagen voorkomen worden.

De tweede fase in deze harmonisering was de ontwikkeling van een allesomvattend process mining framework dat de verschillende algoritmen kan samenbrengen in één applicatie. Deze applicatie kwam er in 2005 en werd ontwikkeld aan de Technische Universiteit Eindhoven onder de naam ProM [10].



Figuur 7: Het process mining landschap met MXML of XES [20]

2.5 Het ProM framework

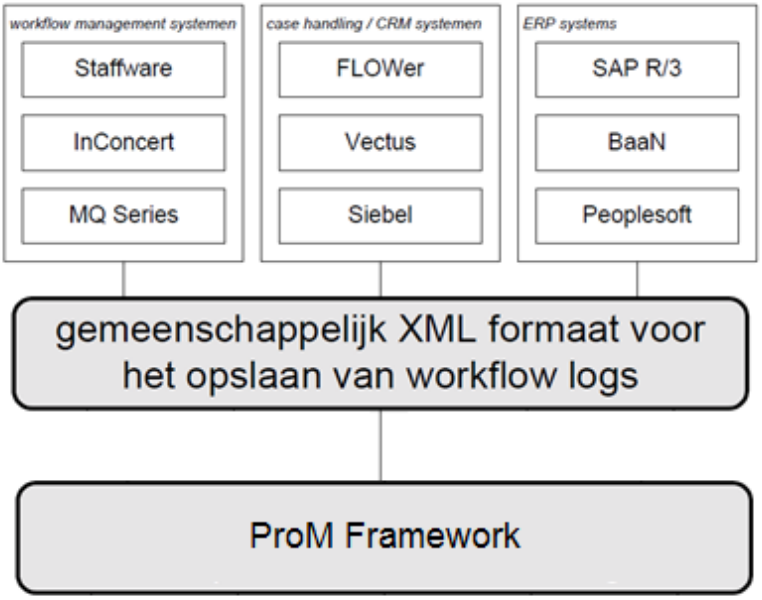
Het ProM framework is een generiek en flexibel open-source framework dat verschillende process mining algoritmen samenbrengt in één applicatie. Het framework is flexibel in verband met het input- en outputformaat en bovendien ondersteunt het onderzoekers door een uitgebreide basis te leveren om nieuwe algoritmen te implementeren. ProM kan event logs in zowel het MXML als het XES formaat inlezen. Dit alles gebeurt aan de hand van een overzichtelijke user interface [1,10].

Het framework is opgebouwd uit verschillende plug-ins. Er zijn 5 verschillende soorten plug-ins [10]:

- *Mining plug-ins*
- *Export plug-ins*
- *Import plug-ins*
- *Analysis plug-ins*
- *Conversion plug-ins*

In de mining plug-ins worden mining algoritmes geïmplementeerd om de feitelijk process mining uit te voeren. Export plug-ins leveren functionaliteiten om objecten, die het resultaat zijn van een mining algoritme, op te slaan. Via de import plug-ins kunnen event logs of geëxporteerde objecten worden ingeladen. De analyse plug-ins implementeren tools voor de analyse van een process mining resultaat. Conversion plug-ins ten slotte implementeren methodes om data te converteren van formaat [10].

Figuur 8 geeft het process mining landschap weer na de ontwikkeling van het ProM framework.



Figuur 8: Het process mining landschap met ProM

In het volgende deel van deze sectie worden twee dataconversie tools besproken: het ProM Import Framework en XESame.

2.5.1 ProM Import Framework

Het ProM Import Framework werd ontwikkeld om event log data te converteren naar het MXML event log formaat. Zo werd het mogelijk om logs van real-life systemen te gebruiken voor process mining. In dit framework wordt een programmeer framework aangeboden om de gebruiker te ondersteunen bij het definiëren en uitvoeren van de data conversie via een makkelijk te gebruiken user interface. Het framework wordt gevormd door een uitbreidbare verzameling van plug-ins die geprogrammeerd worden voor de verschillende dataformaten [1].

Het ProM Import Framework heeft echter enkele beperkingen. Omdat het framework steeds kan worden uitgebreid door het programmeren van nieuwe conversies, moeten de personen die een nieuwe conversie willen implementeren de structuur van het framework kennen en kunnen programmeren in Java. Ook de veronderstelling dat het inputformaat reeds een soort van event log is beperkt de bruikbaarheid van dit framework [1].

2.5.2 XESame

XESame werd ontwikkeld door ing. J.C.A.M. Buijs in het kader van zijn master thesis [1]. Hierin ontwikkelde hij een prototype, toen nog XESMa geheten, dat twee belangrijke functies ondersteunde. Het prototype moest de gebruiker ondersteunen in het definiëren van een conversie van data in een databasestructuur naar het XES event log formaat. Als tweede functie moest het deze conversie ook kunnen uitvoeren [1].

Om deze twee functies te ondersteunen werden enkele functionele en niet-functionele vereisten geformuleerd waarop kon gefocust worden. De eerste vereiste is connectiviteit. De applicatie moet in staat zijn om de meest gebruikelijke types van databronnen in te lezen. De meeste bedrijfsinformatiesystemen gebruiken momenteel relationele databases om data op te slaan. Omwille van het grote aantal *libraries* dat beschikbaar is om data te verkrijgen uit deze databases,

kan iedere relationele database gebruikt worden als inputformaat voor de applicatie. De tweede vereiste is dat de applicatie in staat moet zijn om event logs te creëren in het XES formaat aangezien dit formaat flexibeler is dan het MXML formaat (zie sectie 2.2). Een derde vereiste is dat de gebruiker de conversie zeer precies moet kunnen definiëren zonder dat de gebruiker veel programmeerkennis moet hebben. Een laatste belangrijke vereiste is de gebruiksvriendelijkheid van de applicatie. Hiervoor moet de hoeveelheid programmeerwerk zo klein mogelijk gehouden worden. Daarom wordt de query taal Structured Query Language (SQL) gebruikt voor het creëren van de conversiedefinitie. Deze taal wordt in vele Database Management Systemen (DBMS) gebruikt en heeft een veel simpelere syntax dan echte programmeertalen [1].

Hoofdstuk 3: Conversiebeslissingen

In de literatuurstudie werd reeds vermeld dat de opkomst van een gemeenschappelijk XML-formaat voor het opslaan van event logs en de ontwikkeling van het ProM-framework voor een harmonisering van het process mining landschap heeft gezorgd. Event logs in het MXML- of XES-formaat kunnen tegenwoordig eenvoudig worden ingelezen in het ProM-framework zodat er verschillende process mining algoritmen op de data kunnen worden losgelaten. Het converteren van ruwe outputdata uit een PAIS naar een bruikbare event log in dit formaat is echter geen lichtzinnige taak. Omwille van het feit dat de verschillende algoritmen stuk voor stuk andere data nodig hebben, moeten er bepaalde keuzes en afwegingen gemaakt worden bij het samenstellen van de event log. Bovendien komt het regelmatig voor dat een process mining team het proces binnen eenzelfde project vanuit meerdere perspectieven wil onderzoeken. Zo kan het zijn dat er meerdere conversies moeten worden uitgevoerd om verschillende event logs te creëren van dezelfde databron. In dit hoofdstuk worden de aspecten waarmee rekening gehouden moet worden bij het definiëren van een conversie besproken.

Een eerste beslissing die genomen moet worden in een process mining project is het bepalen van het doel van het project. Vervolgens is het belangrijk om het juiste onderwerp of voorwerp te kiezen dat dienst gaat doen als de procesinstantie of *trace* van het process mining project. Eens er beslist is van welke objecten de eventhistoriek bijgehouden gaat worden in de log, moet er beslist worden welke events juist in deze historiek vervat moeten zitten. De specifieke eigenschappen die bijgehouden moeten worden in de event log van zowel de log-, trace- als eventobjecten moeten vervolgens worden gedefinieerd als attributen. Tenslotte wordt in dit hoofdstuk nog het probleem van con- en divergentie in event logs besproken.

3.1 Doel, scope en focus

De beslissing over de data die in een event log vervat moet zitten hangt voornamelijk af van het doel dat het process mining team voor ogen heeft bij de start van het project. Zoals in sectie 2.4 reeds werd uitgelegd heeft de ontwikkeling van XES en MXML ervoor gezorgd dat event logs niet meer in een applicatie- of algoritmespecifiek formaat moesten worden opgeslagen. Een algemeen XML-formaat zorgde ervoor dat de onderzoeksgroepen zich konden richten op de kern van hun onderzoek, namelijk nieuwe en betere process mining algoritmen vinden, in plaats van waardevolle energie te verspillen aan perifere problemen zoals specifieke datamodellen.

Het voordeel van de vroegere werkwijze was echter dat specifieke data-elementen door de applicatie automatisch geselecteerd werden uit de databron zodat alle benodigde data voor het specifieke algoritme steeds aanwezig was in de event log. Dit voordeel valt weg bij MXML en XES. Hierdoor wordt het belangrijk om op voorhand het doel van het project vast te leggen zodat de juiste data kan worden geselecteerd uit de databron. Nauw verbonden met de beslissing over het doel van een project is de keuze voor het te onderzoeken perspectief en de focus waarop gericht gaat worden. Bij een process mining project waar gekozen is om het procesperspectief te onderzoeken bijvoorbeeld, is het waarschijnlijk voldoende om een event log te definiëren die bestaat uit events met enkel de elementaire datacombinatie (object en activiteit). Sommige algoritmen in dit perspectief kunnen natuurlijk ook nood hebben aan extra data zoals het tijdstip van uitvoering van de verschillende activiteiten. De event log voor een project vanuit het organisatieperspectief moet echter zeker ook de oorsprong of de uitvoerder van de verschillende activiteiten bevatten. Event logs voor een project dat gevoerd wordt vanuit het *case* perspectief zullen dan weer meer attributen bevatten waarin specifieke eigenschappen van de procesinstanties zelf worden beschreven.

Een andere belangrijke beslissing die genomen moet worden na het bepalen van het doel van het project is de scope van het onderzoek. De scope wordt gedeeltelijk bepaald door het doel van het project en bepaalt het onderwerp van het onderzoek. Het geeft aan welk deel van het totale proces onderzocht moet worden [1].

Een derde beslissing die genomen moet worden is de focus van het project. Het doel van het onderzoek spitst zich vaak toe bepaalde activiteiten in het proces [1]. In de focus wordt vastgelegd op welke activiteiten het onderzoek zich moet richten.

3.2 Selectie van het trace-object

Zoals in sectie 2.2 reeds besproken is, bestaan event logs minimaal uit de combinatie van een procesinstantie en een activiteit van het proces die werd uitgevoerd op deze procesinstantie. In secties 2.2.1 en 2.2.2 kan worden teruggevonden dat de structuur van zowel MXML als XES zo bepaald is dat alle events die van toepassing zijn op één procesinstantie bijgehouden worden in een trace of spoor van de specifieke instantie. De event log bestaat dus uit een verzameling van eventhistorieken voor iedere afzonderlijke procesinstantie.

Eens het doel en perspectief van het onderzoek bepaald is, de scope vastgelegd is en er bepaald is op welke activiteiten er gefocust gaat worden, kan er bepaald worden welk onderwerp of voorwerp van het systeem er als procesinstantie voor de event log gaat dienen. In het XES-formaat wordt dit onderwerp vastgelegd als het trace-object [13]. Kandidaat onder- of voorwerpen die als trace-object kunnen dienen zijn vaak bedrijfsobjecten die in het systeem worden behandeld. Voorbeelden van zulke objecten zijn personen, machines, bestellingen of facturen. Alle objecten waarop activiteiten uitgevoerd worden zijn in principe potentiële kandidaten om als trace te dienen. De selectie van het precieze trace-object wordt bepaald aan de hand van de scope.

Het is belangrijk om op te merken dat in één event log slechts één type procesinstantie gebruikt kan worden. Het kan niet zijn dat bepaalde events in een event log handelen over bijvoorbeeld een persoon, terwijl andere events gerelateerd zijn aan een machine. Een trace kan enkel events bevatten die gerelateerd zijn aan één enkel type object [1].

3.3 Selectie van de events

De volgende stap in het selectieproces is de keuze van de events die in de event log inbegrepen moeten zitten. De keuze voor bepaalde events wordt bepaald door de focus van het project. Een process mining project richt zich meestal op enkele specifieke activiteiten in het algemene proces. Het is dus belangrijk om zeker de events die gevormd worden door deze activiteiten in de event log op te nemen. Events van activiteiten van het proces die buiten de focus van het project vallen zijn niet essentieel en kunnen eventueel zelfs uit de event log gelaten worden [1].

Omdat bij process mining ieder event in de event log evenwaardig wordt beoordeeld, is het belangrijk om voor de hele event log een uniform detailniveau voor de events te kiezen. Niet alle activiteiten van een proces worden door een informatiesysteem immers even uitgebreid behandeld of vastgelegd. Sommige activiteiten kunnen bestaan uit verschillende subtaken die moeten worden uitgevoerd om de hoofdactiviteit af te ronden, terwijl andere activiteiten minder uitgebreid zijn uitgewerkt. Het is dan de taak van de onderzoeker om een oplossing te zoeken door in de conversiefase van het onderzoek bepaalde activiteiten te groeperen tot er een uniform detailniveau ontstaat, of door er in de analysefase rekening mee te houden dat het detailniveau kan verschillen van activiteit tot activiteit [1].

Een ander belangrijk aspect waarmee rekening gehouden moet worden bij het opstellen van het detailniveau, is het feit dat events atomair zijn. Ze hebben geen tijdsduur, maar enkel een tijdstip waarop ze zijn uitgevoerd. Het is belangrijk om bij het opstellen van een event log te bepalen of er

rekening moet gehouden worden met de tijdsduur van de verschillende uitgevoerde activiteiten. Dit hangt af van het doel van het onderzoek en het perspectief dat gekozen is. Het kan bijvoorbeeld de bedoeling zijn om de wachttijden voor het starten van activiteiten of de tijd die er nodig is om activiteiten uit te voeren te onderzoeken. Dit kan door activiteiten op te splitsen in twee of meer events waarin de status van de activiteit samen met het tijdstip wordt opgeslagen. Het verschil tussen het tijdstip van het event van een activiteit met status 'start' en het tijdstip van het event van dezelfde activiteit met status 'complete' geeft bijvoorbeeld de verwerkingstijd van die activiteit weer. Event logs voor projecten waar de tijdsduur van activiteiten geen rol in speelt zullen meestal voldoende hebben aan één enkele atomaire weergave van de verschillende activiteiten. Deze events zullen meestal het status 'gereed' en het tijdstip van het einde van de activiteit dragen [1].

3.4 Selectie van de attributen

De laatste belangrijke beslissing die genomen moet worden bij het definiëren van een conversie is het selecteren van de attributen die in de event log moeten worden opgenomen. Ook deze beslissing hangt voor een deel af van het doel van het onderzoek. Als er te weinig attributen opgenomen worden beperkt dit het aantal verschillende analyses en algoritmen die kunnen worden uitgevoerd. Als er te veel attributen worden gedefinieerd dan wordt de event log onnodig groot en dat zou het moeilijk maken om de event log in sommige tools in te laden. Een handige leidraad voor het selecteren van de attributen zijn de voorgedefinieerde extensies met verschillende standaardattributen die gebruikt kunnen worden voor het log-, trace- en eventniveau. Deze attributen zijn terug te vinden in Tabel 2.

Het log-element van een event log is het hoogste niveau. Het omvat alle data die in de event log terug te vinden is. Attributen op het logniveau hebben voornamelijk een beschrijvende functie. Het is eerst en vooral belangrijk om een event log een onderscheidende naam te geven zodat de log

gemakkelijk te klasseren en terug te vinden is. Dit kan door gebruik te maken van het standaarattribuut 'name' uit de concept-extensie. Bovendien is het nodig om in de logattributen voldoende te beschrijven waar de event log over handelt. Dit kan aan de hand van attributen die de naam van het proces, de databron, de organisatie e.d. beschrijven. Het is ook steeds handig om contactinformatie van de persoon die de conversie uitvoert en het process mining project in de event log op te nemen. Met het standaardattribuut 'model' uit de lifecycle-extensie kan worden aangegeven welk transactiemodel er gebruikt wordt voor de events. Het transactiemodel dat standaard gebruikt wordt voor XES is weergegeven in Figuur 3 [1].

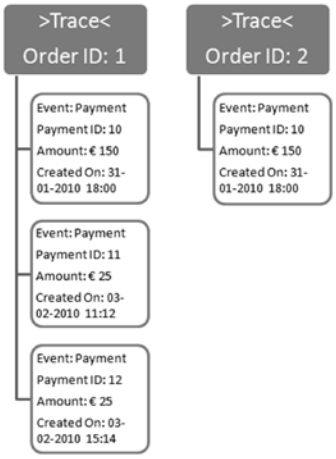
De attributen die bij een trace-object gedefinieerd worden zijn, op de naam van het object na, voornamelijk specifieke attributen die extra informatie over het onderwerp of voorwerp van het process mining project geven. Zulke attributen zijn zeer systeemspecifiek en zijn dikwijls niet noodzakelijk om process mining algoritmen te kunnen toepassen op de event log [1].

De attributen die bepaald worden bij de events daarentegen zijn voornamelijk attributen die uit één van de standaardextensies komen. Deze attributen werden reeds beschreven in sectie 2.2.2 en een overzicht is terug te vinden in Tabel 2.

3.5 Convergentie en divergentie

Zoals hierboven reeds beschreven werd zou ieder afzonderlijk event moeten refereren naar één procesinstantie. Dit is echter niet steeds het geval. Neem nu het geval van een event log in bijvoorbeeld een orderverwerkingssysteem waar als trace-object gekozen is voor de verschillende orders. Één activiteit die op deze orders uitgevoerd moet worden, is de betaling ervan. Maar als een klant verschillende orders heeft openstaan, kan het gebeuren dat de klant de schuld van deze verschillende orders met één betaling aflost. In dat geval zou een betaling met hetzelfde bedrag, dezelfde datum en hetzelfde referentienummer opduiken in de historiek van verschillende trace-

objecten zoals wordt weergegeven in Figuur 9. In deze figuur kan worden gevonden dat betaling met referentienummer 10 opduikt voor zowel het order met referentienummer 1 als nummer 2. Dit fenomeen wordt convergentie genoemd en kan problemen opleveren bij het uitvoeren van sommige process mining algoritmen [1]. Het lijkt immers dat de betaling meerdere keren uitgevoerd werd op hetzelfde ogenblik, zodat de timing en de oorsprong van de betalingen in de event log niet geen getrouw beeld meer zouden geven van de werkelijkheid en bovendien het betaalde totaal in de afrekening niet zou kloppen. Het probleem van convergentie is dan ook aanzienlijk bij het uitvoeren van de vroegste en striktste process mining algoritmen zoals het alfa-algoritme, dat ontwikkeld werd om uit te voeren op 100-procent betrouwbare, maar aldus artificiële event logs [22].



Figuur 9: Convergentie [1]

Een ander probleem dat kan optreden is divergentie [1]. Dit fenomeen is, zoals de naam laat uitschijnen, gerelateerd aan convergentie en gebeurt wanneer het tegengestelde ervan zich voordoet. In [1] wordt divergentie als volgt gedefinieerd:

> *"For one process instance the same activity is performed multiple times."*

Om bij het voorbeeld van een orderverwerkingssysteem te blijven, is dit het geval wanneer een schuld betaald wordt in meerdere keren zoals het geval is bij het order met referentienummer 1 in Figuur 9. Ook bij dit probleem zijn het vooral de eerdere process mining algoritmen zoals het alfa-algoritme dat moeilijkheden hebben om logs met deze gevallen te behandelen.

Het probleem van convergentie en divergentie kan niet altijd vermeden worden. Maar door de opkomst van nieuwe en lossere process mining algoritmen kan process mining in zulke gevallen toch worden uitgevoerd. In de case study die verder in deze thesis behandeld wordt zal het vooral het probleem van divergentie zijn dat optreedt. Meer daarover in hoofdstuk 4.

Deel III: Experimenteel onderzoek

Hoofdstuk 4: Case study

Om de werking van de conversietool XESame te testen en te valideren wordt in deze volgende sectie de conversie van een dataset met real-life data naar een event log gedefinieerd. De data in dit voorbeeld is afkomstig van een mobiliteitsonderzoek dat gevoerd wordt aan de Universiteit Hasselt. De case study is opgebouwd uit verschillende fasen. In de eerste fase wordt de oorsprong van de dataset besproken, het te onderzoeken process mining probleem geschetst en het doel van het project uitgelegd. Vervolgens wordt de dataset onderzocht en klaargemaakt voor de conversie. In de volgende fase wordt de conversiedefinitie opgesteld en vervolgens wordt de conversie uitgevoerd.

4.1 Probleemstelling

Zoals hierboven reeds vermeld, is de data uit deze case study afkomstig uit een mobiliteitsonderzoek. De onderzoekers van dat project proberen aan de hand van real-life data over de activiteiten en verplaatsingen van een groep testpersonen mobiliteitsmodellen op te stellen om toekomstige verkeersstromen in kaart te brengen. De data wordt ingegeven in een specifieke softwareapplicatie, het FEATHERS-framework, zodat er modellen kunnen worden opgesteld. Deze modellen leveren voorspellingen op over de activiteiten en verplaatsingen van deze personen. Die data wordt vervolgens vergeleken met de originele gegevens om de modellen te verfijnen en op punt te zetten. Het is namelijk zo dat een dag zoals hier gedefinieerd wordt één lange aaneenschakeling van activiteiten en verplaatsingen moet zijn. Maar bij de voorspelde waarden is dit niet steeds het geval. Daarom is het belangrijk om deze data weer te geven in een duidelijk gestructureerd formaat en de data te onderzoeken op zulke fouten.

Hoewel het hier niet om een conventioneel process mining project gaat is de keuze voor het XES-formaat als dataopslagmethode toch begrijpelijk. Het project heeft namelijk veel eigenschappen van process mining. Zo wordt bijvoorbeeld het proces onderzocht van de verschillende activiteiten die een procesinstantie, in dit geval de verschillende testpersonen van het onderzoek, doorloopt. Het verschil met een gewoon process mining project is dat het informatiesysteem waaruit de data komt geen directe invloed heeft op het te bestuderen proces. Dit informatiesysteem behoort niet tot het algemene systeem waarin het proces vervat zit, het is niet meer dan een opslagplaats waarin data die handelt over een ander proces in opgeslagen wordt.

4.2 Dataset

In deze sectie wordt de dataset met de voorspelde waarden onderzocht en besproken om een duidelijk beeld te krijgen van de inhoud. De dataset bestaat uit een verzameling van 6000 testpersonen. Deze personen voeren samen in totaal 24204 activiteiten uit en ze verplaatsen zich 18166 keer. De dataset bevat bovendien nog andere data over de personen, activiteiten en verplaatsingen.

Over de verschillende testpersonen wordt bijvoorbeeld nog bijgehouden tot welke leeftijdscategorie ze behoren en het geslacht van de persoon in kwestie. Er bestaan 5 leeftijdscategorieën: 0 tot 34 jaar, 35 tot 54 jaar, 55 tot 64 jaar, 65 tot 74 jaar en groter of gelijk aan 75 jaar. Ook wordt er bijgehouden of de persoon een rijbewijs heeft en of de persoon voltijds, halftijds of niet werkt. Er wordt ook extra informatie bijgehouden over de verschillend activiteiten. Eerst en vooral wordt er voor iedere activiteit natuurlijk geregistreerd welke persoon de respectievelijke activiteit uitvoert. Er wordt ook bijgehouden wat de activiteit inhoudt. Daarom worden de activiteiten geclassificeerd in 9 groepen activiteiten. Ten slotte wordt voor iedere activiteit het begintijdstip en de tijdsduur van de activiteit geregistreerd. De verplaatsingen hebben dezelfde structuur als de activiteiten. Er wordt

bijgehouden welke persoon een verplaatsing onderneemt, hoe de verplaatsing ondernomen wordt en het begintijdstip en de tijdsduur van de verplaatsing.

De dataset werd oorspronkelijk geleverd in de vorm van een Microsoft Excel document. Het document is onderverdeeld in vier tabbladen: 'Predicted_Persons' waarin de verschillende testpersonen worden bijgehouden, 'Predicted_Activities' dat de verschillende uitgevoerde activiteiten bevat, 'Predicted_Journeys' met de verplaatsingen en 'Data classes' waarin de verschillende attributen van de tabellen en de datacategorieën worden uitgelegd.

4.3 Preprocessing

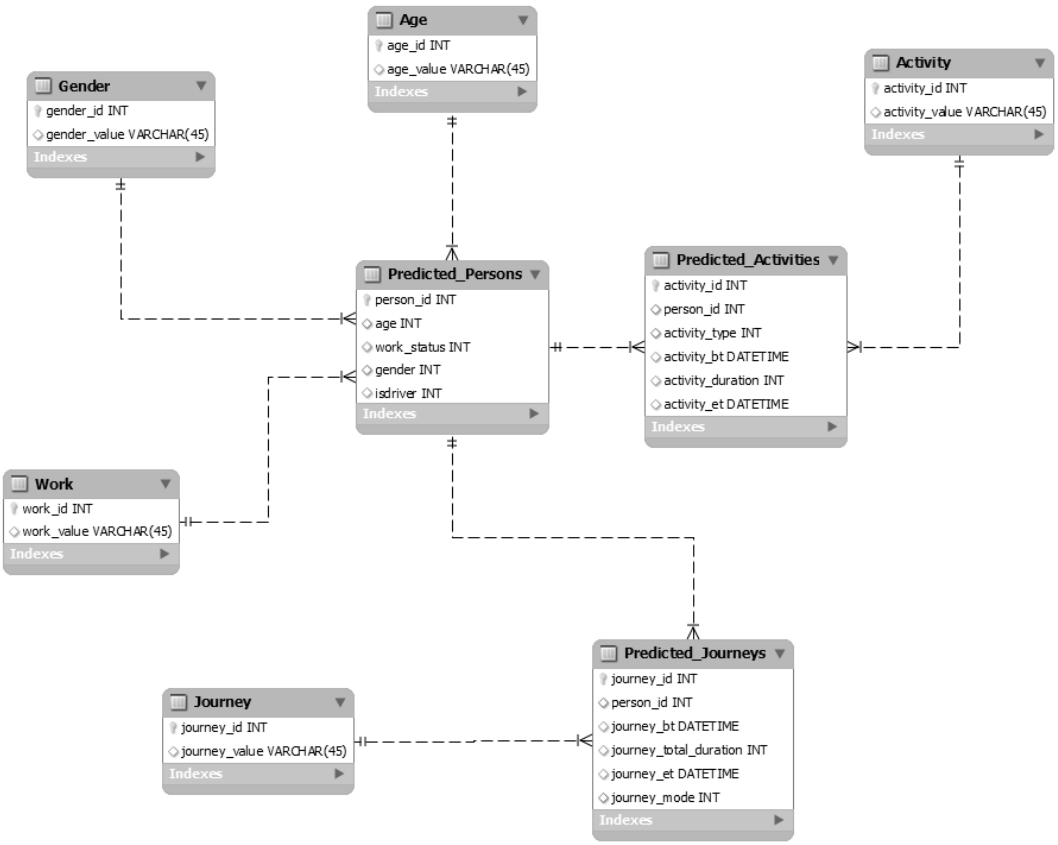
In deze voorbereidende fase wordt de dataset gebruiksklaar gemaakt voor de verdere uitvoering van het process mining project. Deze eerste stap in het conversieproces is zeer belangrijk aangezien de dataset en de individuele data-attributen het juiste formaat moeten hebben om ingelezen te kunnen worden in XESame.

Eerst en vooral wil dit zeggen dat de dataset genormaliseerd moet worden en opgeslagen in een formaat waarmee de applicatie overweg kan. Het doel van databasenormalisatie is om steeds herhaalde data in een aparte tabel op te slaan zodat het dubbel opslaan ervan vermeden wordt. Er zijn voor deze dataset in dat aspect reeds grote stappen ondernomen door de verschillende datacategorieën op te splitsen en uit te werken in het aparte tabblad 'Data classes'. Dit tabblad wordt weergegeven in Tabel 4.

Tabel 4: Tabblad 'Data classes'

	A	B
1		
2	Age	0: <35, 1: 35<55, 2: 55- <65, 3: 65-<75, 4:>75 years
3	work_status	0: no work, 1: part time, 2: full time work
4	gender	0: male, 1: Female
5	IsDriv	0: Don't drive, 1: Can drive
6		
7	activity_type	0:Being at home , 1: Work , 3:Bring/Get , 4-5: Shopping ,6:Services ,7:Social visits ,8:Leisure ,9: Touring ,10:Other
8	journey_mode	1: Car, 3: Slow , 4: Public transport, 6: Car passenger
9		

De volgende stap die dus nog moet worden genomen in het normalisatieproces is de verschillende datacategorieën die weergegeven zijn in dit tabblad op te splitsen in aparte tabellen. De genormaliseerde dataset zal er vervolgens uitzien zoals in het ER-diagram in Figuur 10



Figuur 10: Entiteit relatie diagram van de genormaliseerde dataset

Uit deze figuur kan men afleiden dat de verschillende datacategorieën zijn opgesplitst in de aparte tabellen 'Gender', 'Age', 'Activity', 'Work' en 'Journey'. Deze tabellen bevatten dezelfde informatie als het tabblad 'Data classes' in de oorspronkelijk dataset, maar de data is gestructureerd op een manier die erkend is bij databasenormalisatie. Een voorbeeld van deze structuur is in Tabel 5 weergegeven voor de tabel 'Activity'. De unieke numerieke waarde van het attribuut 'activity_type' in de tabel 'Predicted_Activities' staat gelijk aan de numerieke waarde in het veld 'activity_id' in de tabel 'Activity'. Deze numerieke waarde verwijst naar de corresponderende waarde in het attribuut 'activity_value' van dezelfde tabel. Zo zijn de verschillende waarden die het attribuut 'activity_type' in de tabel 'Predicted_Activities' kan aannemen op voorhand bekend en kunnen er geen fouten optreden bij het invoeren van de data.

Tabel 5: Tabel 'Activity'

	A	B
1	activity_id	activity_value
2	0	Being at home
3	1	Work
4	3	Bring/Get
5	4	Shopping
6	6	Services
7	7	Social visits
8	8	Leisure
9	9	Touring
10	10	Other

De volgende stap in de preprocessingfase is ervoor zorgen dat de dataset in een voor XESame aanvaardbaar formaat wordt opgeslagen. Zoals in [1] gelezen kan worden, aanvaardt XESame data in de vorm van relationele databases. De genormaliseerde dataset zoals in Figuur 10 heeft de structuur van zo een relationele database. De enige ontbrekende stap is deze dataset ook op te slaan als zo een database. In [1] kan er ook gevonden worden dat er verschillende *libraries* beschikbaar zijn om *comma-seperated-value* (CSV) bestanden te behandelen alsof het relationele databases zijn. ODBC, voluit *Open Database Connectivity,* is zo een standaard toegankelijkheidsmethode voor databases.

Omdat het makkelijk is om Excel documenten om te zetten naar het csv-formaat, is er dan ook voor gekozen om dit formaat te gebruiken. De omzetting van het Excel-formaat 'XLS' naar het CSV-formaat gebeurt door ieder tabblad afzonderlijk in Excel op te slaan als CSV-bestand. Het resultaat is dat de zeven tabellen die te zien zijn in Figuur 10 nu opgeslagen zijn als zeven afzonderlijk CSV-bestanden met ieder hun respectievelijke naam. Deze zeven tabellen moeten vervolgens voor XESame als een relationele database beschikbaar gemaakt worden via een ODBC-stuurprogramma voor tekstbestanden zoals CSV-documenten.

In de laatste stap van de preprocessingfase moet er voor worden gezorgd dat de individuele attributen van de verschillende tabellen in het voor XESame vereiste dataformaat opgeslagen zijn. Dit wil zeggen dat numerieke waarden ook een numerieke vorm hebben, tekstwaarden ook een tekstuele vorm hebben en tijdstippen weergegeven worden in het door XESame aanvaarde *datetime*-formaat. Omdat waarden in een CSV-bestand geen metadata meedragen over het formaat waarin iedere waarde wordt opgeslagen, vormen er zich geen problemen voor de numerieke en tekstuele waarden. Dit wil echter ook zeggen dat er voor de tijdstippen ook geen enkele indicatie over het formaat wordt meegedragen in het bestand, zodat XESame enkel aan de specifieke vorm die deze tijdstippen aannemen kan aflezen dat het om een waarde in het datatime-formaat gaat.

In Tabel 6 kan een fragment van de tabel 'Predicted_Activities' worden gevonden. In het attribuut 'activity_bt' wordt het begintijdstip van iedere activiteit weergegeven voor de verschillende testpersonen. Zoals in de tabel kan worden afgelezen, worden de tijdstippen in deze dataset op een ongewone manier voorgesteld. Een tijdstip wordt in het gebruikte formaat steeds weergegeven als een numerieke waarde van minimaal drie en maximaal vier cijfers. De laatste twee cijfers van deze getallen geven het minutengedeelte van het tijdstip aan, terwijl de cijfers die voor deze laatste twee cijfers staan het uurgedeelte van het tijdstip aangeven. Zo begon bijvoorbeeld de activiteit met

identificatienummer 0 om 3:00u en activiteit met nummer 1 om 13:16u. Dit probleem dook vervolgens ook op in de tabel 'Predicted_Journeys'.

Tabel 6: Tabel 'Predicted_Activities' voor datatransformatie

	A	B	C	D	E
1	Activity_id	person_id	activity_type	activity_bt	activity_duration
2	0	0	0	300	611
3	1	0	1	1316	93
4	3	0	0	1454	238
5	5	0	8	1901	225
6	7	0	0	2255	245
7	9	1	0	300	754
8	10	1	3	1543	5
9	12	1	0	1557	43
10	14	1	3	1657	14
11	16	1	0	1728	572
12	18	2	0	300	265
13	19	2	1	744	508

Om deze tijdstippen vervolgens te converteren naar het datetime-formaat werd er een klein, zelfgeschreven C++ programma gebruikt. Dit programma las de data van de tabellen in het CSV-formaat lijn na lijn in. Na het inlezen van een lijn splitste het de drie- of viercijfer getallen van de tijdstippen op in de twee delen, namelijk het minutengedeelte en het uurgedeelte, zodat het tijdgedeelte kan worden opgeslagen in de juiste vorm. Zoals de naam van het formaat, *datetime*, reeds doet vermoeden bestaat dit formaat ook uit een datumgedeelte. Omdat er geen data over de datum in de dataset meegeleverd wordt, is er voor gekozen om een standaardwaarde voor het datumgedeelte te gebruiken. Deze assumptie is aanneembaar door de aard van de data in het mobiliteitsonderzoek. In dit onderzoek wordt namelijk enkel data bijgehouden over de activiteiten en verplaatsingen van de testpersonen op één specifieke dag. Alle deze activiteiten vinden dus plaats op dezelfde dag zodat één standaardwaarde voor de datum gerechtvaardigd is. De gekozen waarde, 1 januari 1970, is geïnspireerd door de *UNIX time* die berekend wordt door de seconden bij te tellen

die verstreken zijn sinds deze datum. Zoals reeds vermeld werd is dit slechts een waarde zonder specifieke betekenis, iedere andere datum zou gekozen kunnen worden om als standaardwaarde te dienen.

Een ander probleem dat optreedt bij de tijdsweergave in de tabellen 'Predicted_Activities' en 'Predicted_Journeys' is dat er geen eindtijdstip voor de respectievelijke activiteiten en verplaatsingen wordt bijgehouden. Er wordt enkel een tijdsduur opgeslagen in het attribuut 'activity_duration'. De definitie van een event die in deze thesis gebruikt wordt schrijft echter voor dat events atomair zijn en geen tijdsduur hebben, maar enkel een begin- en eindtijdstip. Het eindtijdstip van de verschillende activiteiten en verplaatsingen kan echter gemakkelijk berekend en opgeslagen worden aan de hand van het begintijdstip en de tijdsduur die reeds beschikbaar zijn en een eenvoudige Excel-bewerking. Het resultaat van deze transformaties kan worden teruggevonden in Tabel 7. Fragmenten van alle tabellen uit het ER-diagram kunnen ook worden teruggevonden in de bijlagen.

Tabel 7: Tabel 'Predicted_Activities' na datatransformatie

	A	B	C	D	E	F
1	Activity_id	person_id	activity_ty	activity_bt	activity_duration	activity_et
2	0	0	0	1/01/1970 3:00	611	1/01/1970 13:11
3	1	0	1	1/01/1970 13:16	93	1/01/1970 14:49
4	3	0	0	1/01/1970 14:54	238	1/01/1970 18:52
5	5	0	8	1/01/1970 19:01	225	1/01/1970 22:46
6	7	0	0	1/01/1970 22:55	245	2/01/1970 3:00
7	9	1	0	1/01/1970 3:00	754	1/01/1970 15:34
8	10	1	3	1/01/1970 15:43	5	1/01/1970 15:48
9	12	1	0	1/01/1970 15:57	43	1/01/1970 16:40
10	14	1	3	1/01/1970 16:57	14	1/01/1970 17:11
11	16	1	0	1/01/1970 17:28	572	2/01/1970 3:00
12	18	2	0	1/01/1970 3:00	265	1/01/1970 7:25
13	19	2	1	1/01/1970 7:44	508	1/01/1970 16:12

Nu de dataset genormaliseerd is, alle data-attributen de juiste vorm hebben en de dataset in het CSV-formaat opgeslagen is, zijn alle puzzelstukken verzameld om aan de conversiedefinitie te beginnen.

4.4 Conversiedefinitie

Zoals in hoofdstuk 3 reeds besproken is, is het belangrijk om voor aanvang van de conversiedefinitie het doel, de scope en de focus van het process mining project vast te leggen. Deze beslissingen zullen tijdens het verloop van de conversie fungeren als een leidraad en ze zullen helpen bij het maken van alle verdere keuzes die opduiken bij het definiëren van de event log.

Uit de probleemstelling van deze case study in sectie 4.1 werd reeds duidelijk dat het doel van het process mining project is om te controleren of er geen fouten geslopen zijn in de data over de activiteiten en verplaatsingen van personen die voorspeld worden door de mobiliteitsmodellen. Volgens de assumpties die gemaakt werden bij het mobiliteitsonderzoek, bestaat een dag namelijk uit een lange aaneenschakeling van zulke activiteiten en verplaatsingen zonder onderbrekingen. Ieder ogenblik van de dag moet toegekend zijn aan een van die activiteiten of verplaatsingen.

De omschrijving van het doel in de vorige paragraaf levert ook een goede omschrijving van de scope en de focus van het process mining project. Het is duidelijk dat het onderwerp van het proces de verschillende testpersonen zijn waarover mobiliteitsinformatie bijgehouden wordt. Voor de focus van het onderzoek te bepalen moet er worden gekeken naar die activiteiten waarop het onderzoek zich gaat richten. De specifieke activiteiten die in de event log moeten vervat zitten zijn dus alle voorspelde activiteiten die de testpersonen ondernemen plus de verplaatsingen die hen van en naar deze activiteiten brengen.

Een moeilijke en verregaande stap die vervolgens in het proces van de conversiedefinitie genomen moet worden en die ook teruggevonden kan worden in sectie 2.3, is het bepalen van het perspectief waaruit het process mining project benaderd zal worden en of er een kwalitatieve *(logical)* dan wel een kwantitatieve *(performance)* nadruk op dit perspectief zal liggen. Deze beslissingen zijn een vertaling van de meer theoretische keuze van het doel van het onderzoek naar de eerder praktische

invulling ervan. Het is deze vertaling naar een praktische uitwerking om het probleem op te lossen die de hoeksteen van het verdere onderzoek gaat vormen. Omdat de oplossing van process mining problemen niet altijd voor de hand ligt, vergt het vaak een ernstige denkoefening die besproken wordt in de volgende paragraaf.

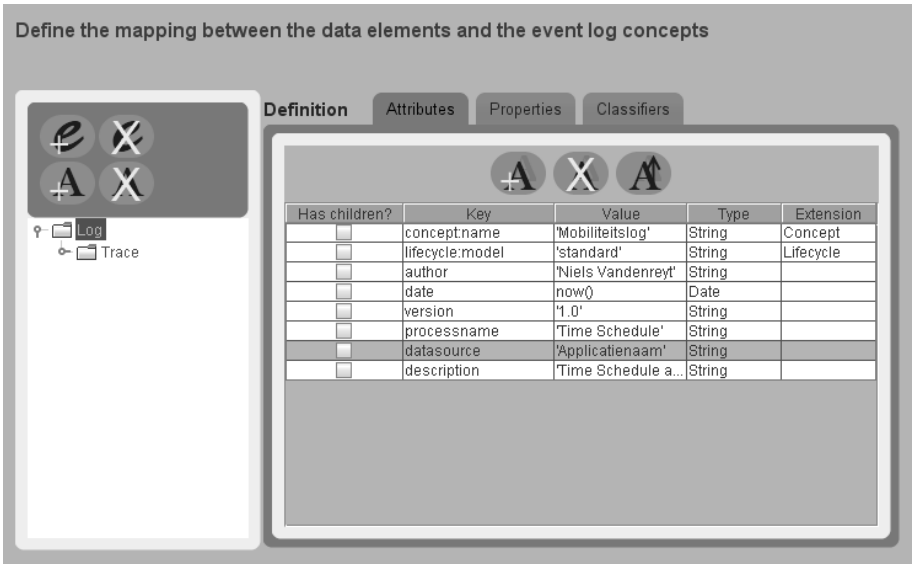
Het doel van het mobiliteitsproject is een model op te stellen dat de tijdsbesteding van testpersonen voorspelt. Het probleem dat kan optreden in de data die voorspeld wordt is dat er gaten zitten in de tijdsbesteding van de testpersonen. Het proces dat iedere testpersoon doorloopt, van de eerste tot de laatste activiteit van de dag, kan met andere woorden op sommige momenten door een onnauwkeurigheid van het mobiliteitsmodel onderbroken worden. Het mag dan al wel duidelijk zijn dat het proces centraal staat en er dus vanuit het procesperspectief naar het process mining project gekeken kan worden, de nadruk vergt toch nog enige verdere uitleg. Ook al is het zo dat het mobiliteitsonderzoek geïnteresseerd is in het vinden van een kwalitatief model om de tijdsbesteding te voorspellen, deze process mining case study is dat niet. Het enige waarin men hier geïnteresseerd is, is om fouten in de voorspellingen op te sporen. Dat kan als er naar dit project gekeken wordt met een kwantitatieve nadruk die zich richt op de prestaties van het proces. Het is namelijk zo dat een onderbreking die in het proces voor de aanvang van een nieuwe activiteit of verplaatsing optreedt, gezien kan worden als *waiting time* die plaatsvindt in het proces. Een kwantitatief prestatieonderzoek naar de wachttijden die optreden in het proces zal onthullen waar er fouten zijn geslopen in de voorspellingen. Het onderzoek zal dus gevoerd worden vanuit een procesperspectief met de nadruk op de prestaties van het proces.

Nu al deze voorafgaande beslissingen genomen zijn, is het gemakkelijk om te bepalen welk onderwerp van het proces er gekozen kan worden als procesinstantie. Deze procesinstantie zal fungeren als het trace-object waarvan alle events die de procesinstantie doorloopt geregistreerd

worden in de event log. Omwille van de definitie van de scope voor het process mining onderzoek is het voor de hand liggend dat de verschillende testpersonen gekozen worden als de trace-objecten voor de event log. De keuze voor de voorspelde activiteiten en verplaatsingen als de events voor de log wordt ingegeven door de definitie van de focus.

Een volgende belangrijke fase is de selectie van de attributen voor de log-, trace- en eventobjecten. In deze attributen worden de eigenschappen van de verschillende objecten opgeslagen en ze vormen een belangrijke bron van informatie.

Zoals in sectie 3.4 reeds besproken werd, vervullen de attributen op het logniveau een omschrijvende functie van de event log. Voor het logniveau werd er eerst gekozen voor twee standaardattributen die gedefinieerd zijn in de extensies. De event log krijgt een duidelijke naam via het attribuut 'name' uit de extensie 'concept'. In het attribuut 'transition' uit de extensie 'lifecycle' wordt er bepaald dat het standaard transactiemodel wordt gebruikt zoals weergegeven in Figuur 3. Verder worden er nog een reeks handgemaakt attributen bepaald voor het logniveau zoals de naam van het proces dat onderzocht wordt en de databron waarvan deze dataset afkomstig is. Bovendien wordt de naam van de auteur van de event log bijgehouden samen met de conversiedatum en de versie. Ten slotte wordt er een beschrijving gegeven van de gedefinieerde event log. De gekozen attributen zijn ook terug te vinden in Figuur 11.



Figuur 11: Specificatie van de logattributen

De volgende attributen die gedefinieerd moeten worden zijn de attributen van het trace-object. Attributen leveren extra informatie over de objecten die besproken worden. Het is echter belangrijk om op te merken dat in tegenstelling tot de attributen op het logniveau die slechts eenmaal in de event log opgeslagen moeten worden, de attributen die bepaald worden voor de trace-objecten bij iedere nieuwe procesinstantie in de event log moeten worden opgenomen. Het opnemen van attributen op dit niveau is dus een taak die weloverwogen moet worden aangezien de extra informatie die nieuwe attributen leveren voor het process mining project misschien niet in verhouding staat tot de verwerkingstijd die nodig is om de event log te genereren. Om dit aan te tonen wordt er in de volgende paragraaf een klein experiment beschreven.

In dit experiment worden drie korte dataconversies uitgevoerd met het XESame-framework. De eerste conversiedefinitie is weergegeven in Figuur 12 en is een zeer elementaire definitie waarin slechts één attribuut wordt gedefinieerd, namelijk de naam van het trace-object in de vorm van het identificatienummer van de testpersoon. In de volgende conversiedefinitie wordt hetzelfde attribuut gebruikt om de naam van het object te bepalen, maar worden er extra data-attributen bijgehouden

die informatie over de leeftijd, het werkstatus en het geslacht van de testpersoon opslaan. Deze conversiedefinitie is terug te vinden in Figuur 13. De laatste definitie is te zien in Figuur 14 en breidt de vorige conversiedefinitie uit door gebruik te maken van de *foreign keys* die bij de databasenormalisatie in de dataset zijn verwerkt. In plaats van de datacategorieën voor de drie extra attributen met hun corresponderende numerieke waarde aan te geven, wordt er ter vervanging het tekstuele label voor deze categorieën opgeslagen dat in de verschillende gekoppelde tabellen gevonden kan worden. Het is belangrijk om te vermelden dat er voor geen van de drie conversiedefinities events gedefinieerd zijn en dat de logattributen voor de drie definities gelijk zijn. De enige verschillen die optreden situeren zich dus op het niveau van de trace-objecten die hierboven besproken zijn.

Has children?	Key	Value	Type	Extensi...
☐	concept:name	'Person ' & person_id	String	Concept
☐	semantic:model...		String	Sema...

Figuur 12: Tracedefinitie 1

Has children?	Key	Value	Type	Extensi...
☐	concept:name	'Person ' & person_id	String	Concept
☐	semantic:model...		String	Sema...
☐	age	Age	String	
☐	work	work_status	String	
☐	gender	gender	String	

Figuur 13: Tracedefinitie 2

Has children?	Key	Value	Type	Extensi...
☐	concept:name	'Person ' & person_id	String	Concept
☐	semantic:model...		String	Sema...
☐	age	age.age_value	String	
☐	work	work.work_value	String	
☐	gender	gender.gender_value	String	

Figuur 14: Tracedefinitie 3

Vervolgens werden deze drie conversiedefinities ook daadwerkelijk uitgevoerd en de verkregen resultaten kunnen worden afgelezen in Tabel 8. In de resultaten kan gevonden worden dat door de

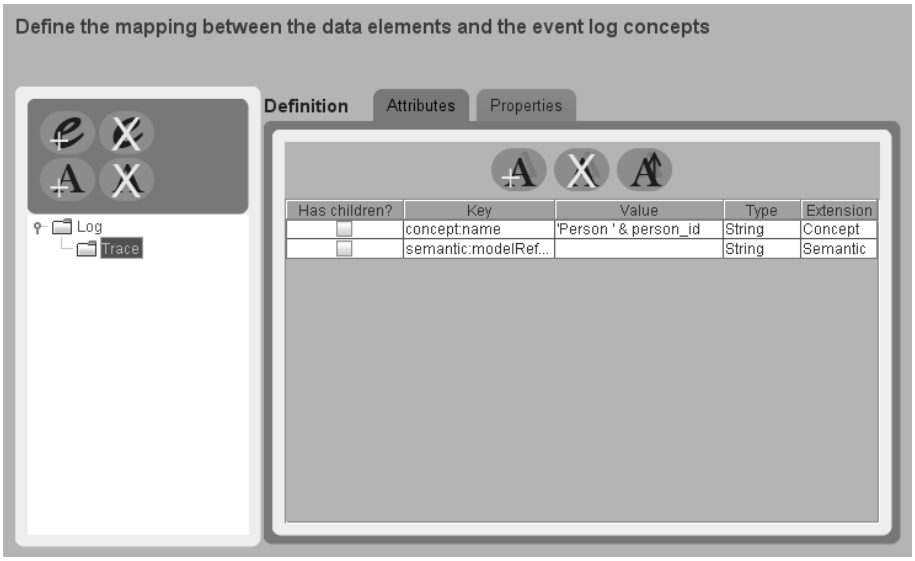
grote van de dataset de invloed van extra attributen groot is op zowel de conversietijd als de grootte van de event log. De invloed die het gebruik van de tekstuele waarden op de conversietijd en grootte heeft daarentegen is eerder beperkt.

Er kan dus worden besloten dat men voorzichtig moet zijn met het toevoegen van extra data-attributen bij die objecten waarvan vele instanties in de event log vervat zitten. Het gebruik van gekoppelde tabellen heeft echter een kleine invloed op de conversietijd en grootte van de event log en kan worden aangemoedigd indien dit de duidelijkheid van de event log ten goede komt.

Tabel 8: Resultaten

	Definitie 1	Definitie 2	Definitie 3
Conversietijd	1 min 6 s 916 ms	1 min 34 s 286 ms	1 min 34 s 826 ms
Event log grootte	411 KB	997 KB	1082 KB

Omdat het doel van dit process mining project enkel is om de foute voorspellingen eruit te halen en de dataset uit een behoorlijk groot aantal testpersonen bestaat, is het echter niet nodig noch wenselijk om veel extra data op te nemen in de trace-objecten. Daarom is er voor gekozen om de traces enkel te bepalen met het identificatienummer dat ze meekregen in de dataset. Zoals in Figuur 15 opgemerkt kan worden, worden deze identificatienummers steeds voorafgegaan met het woord 'Person'. Het trace-object van de persoon met identificatienummer nul zal worden opgeslagen als 'Person 0'. In het tabblad eigenschappen bij het definiëren van de trace-objecten moet vervolgens worden gespecificeerd in welke tabel de benodigde data teruggevonden kan worden.

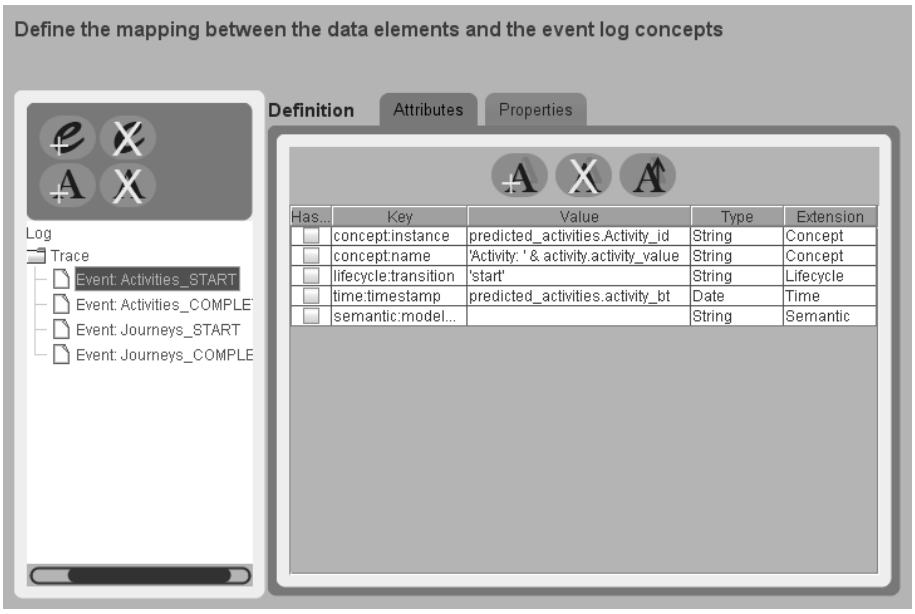


Figuur 15: Specificatie van de trace-attributen

De stap die volgt in het conversieproces is het definiëren van de verschillende events en hun bijhorende attributen. Ook voor het definiëren van deze attributen is voorzichtigheid geboden, aangezien het aantal events dat in de event log ingevoerd zal worden nog groter is dan het aantal testpersonen er in de dataset zit. Iedere testpersoon doorloopt dan ook meerdere activiteiten en verplaatsingen op een dag. Daarom is er ook voor gekozen om bij deze events het aantal attributen te beperken tot het minimum dat nodig zal zijn om het process mining project uit te kunnen voeren.

De eerste twee events die gedefinieerd worden zijn het begin en het einde van de verschillende activiteiten die de testpersonen ondernemen. De attributen die voor deze events gedefinieerd worden zijn terug te vinden in Figuur 16. Het eerste attribuut dat bepaald wordt is het attribuut 'instance'. In dit attribuut wordt het identificatienummer van iedere uitgevoerde activiteitsinstantie bijgehouden. Zo blijft er een link bestaan tussen de data in de event log en de originele data in de dataset om iedere activiteit in de event log te kunnen traceren. Dit identificatienummer is terug te vinden in de tabel 'Predicted_Activities'. In het volgende attribuut wordt de algemene naam voor de uitgevoerde activiteit geplaatst voorafgegaan met het woord 'Activity'. Deze naam wordt gehaald uit

de tabel 'Activity' waarin de verschillende categorieën van activiteiten opgeslagen zitten. De volgende twee attributen zijn afhankelijk van het type event dat geregistreerd wordt. Bij events die het begin van een activiteit aanduiden wordt het status 'start' aangegeven in het attribuut 'transition' en het begintijdstip van de activiteit in het attribuut 'timestamp'. In deze attributen wordt bij events die het einde van een activiteit aanduiden de waarde 'complete' en het eindtijdstip ingevuld.



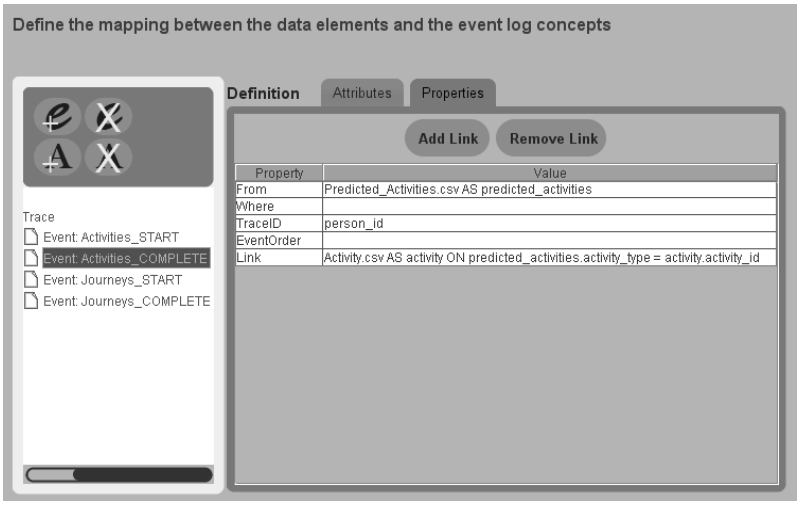
Figuur 16: Attributen van het startevent Activities

In Figuur 17 kunnen de eigenschappen van de attributen voor het event dat de activiteiten beschrijft worden teruggevonden. Hierin staat dat de data voor dit event in de eerste plaats afkomstig is uit de tabel 'Predicted_Activities'. Verder wordt deze tabel gekoppeld aan de tabel 'Activity' op basis van de *foreign key* die terug te vinden is in het ER-diagram in Figuur 10. Deze *key* verbindt de twee tabellen op basis van het attribuut 'activity_type' in de tabel 'Predicted_Activities' en het attribuut 'activity_id' in de tabel 'Activity'.

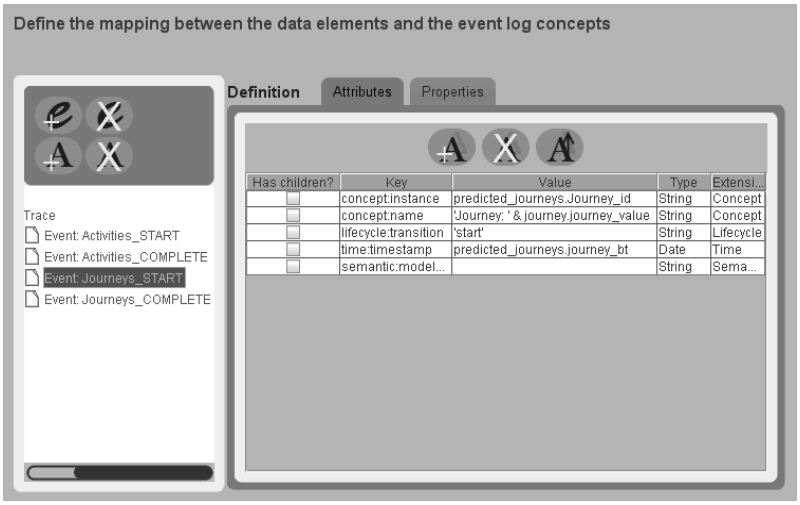
De andere twee events die gedefinieerd moeten worden zijn het begin en einde van de verplaatsingen die de testpersonen ondernemen. Omdat het belangrijk is om een uniform

detailniveau aan te houden voor de verschillende events, zoals reeds beschreven staat in sectie 3.3, hebben deze twee events een gelijkaardige structuur als de twee events die handelen over de activiteiten. Deze structuur kan teruggevonden worden in Figuur 18. Ook de eigenschappen van dit event in Figuur 19 hebben dezelfde structuur, op uitzondering van de *foreign key*. Deze *key* verbindt uiteraard de tabel 'Predicted_Activities' met de tabel 'Journey' op basis van het attribuut 'journey_mode' en is ook terug te vinden in Figuur 10.

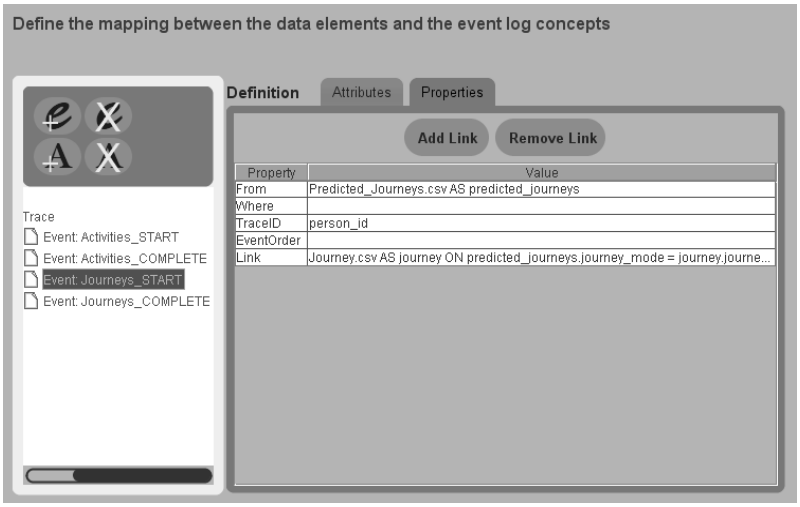
Nu alle events gedefinieerd zijn is de conversiedefinitie van de event log klaar om uitgevoerd te worden. Een volledig overzicht van de verschillende gedefinieerde attributen en events samen met de tabellen waarvan de data afkomstig is, is uitgewerkt in Figuur 20.



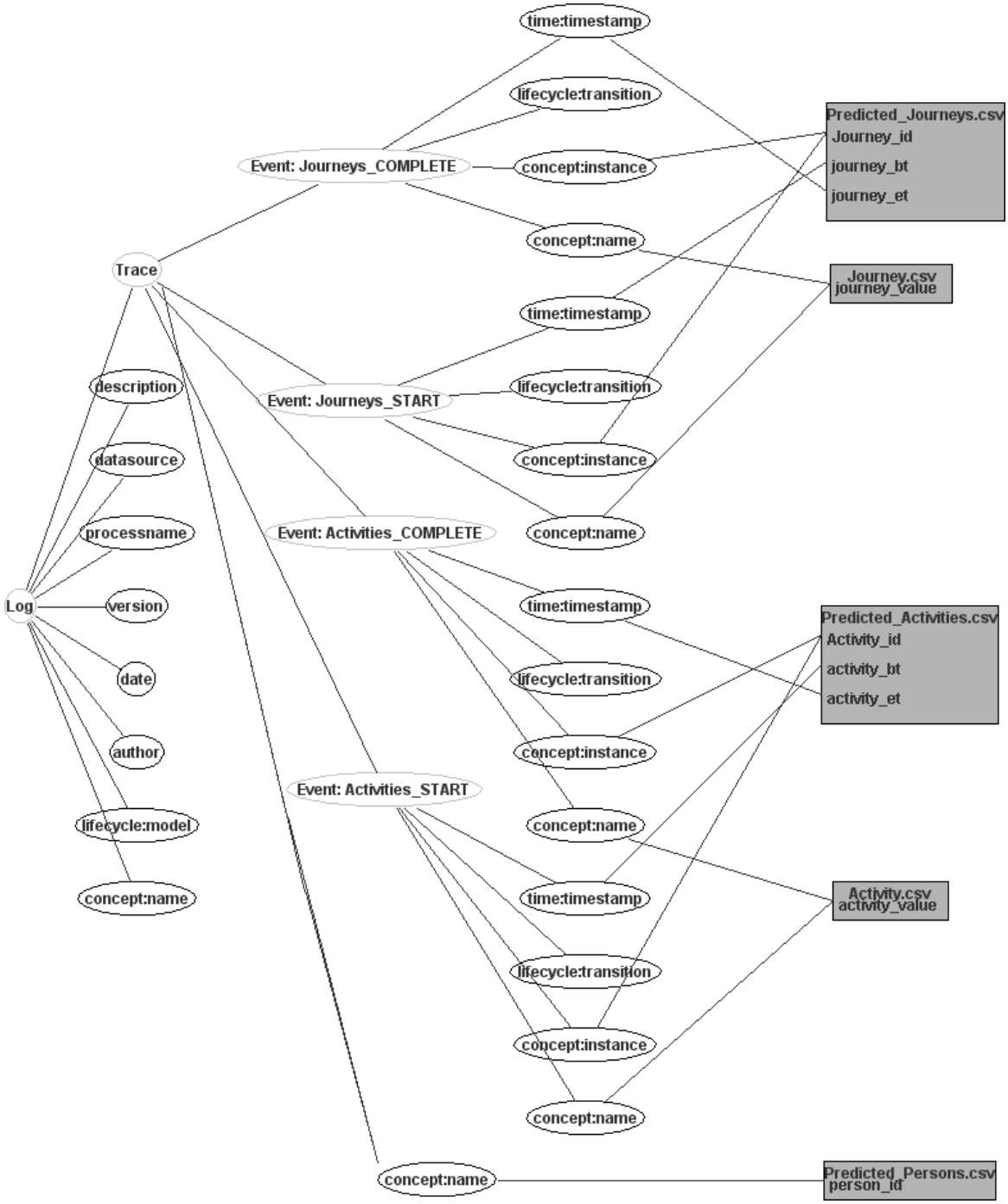
Figuur 17: Eigenschappen van het startevent Activities



Figuur 18: Attributen van het startevent Journeys



Figuur 19: Eigenschappen van het startevent Journeys

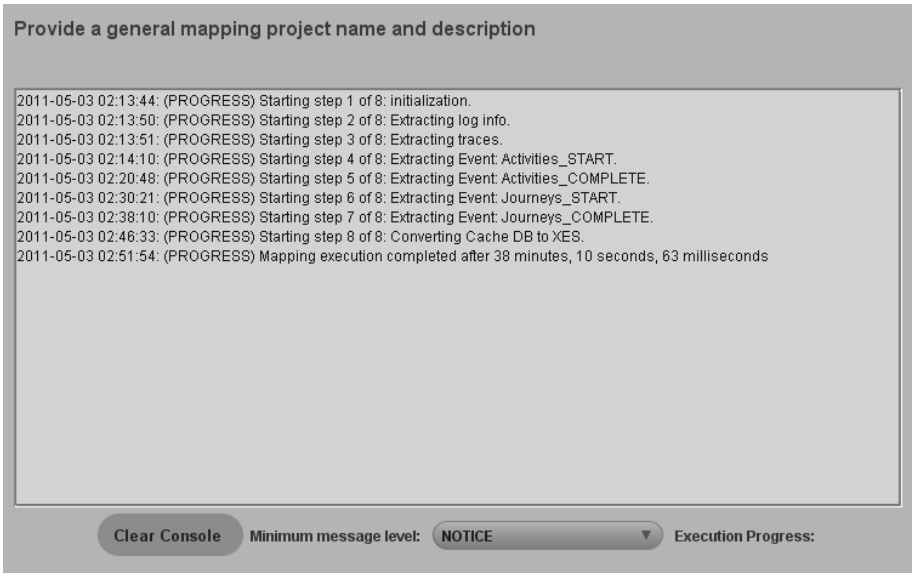


Figuur 20: Overzicht event log definitie

4.5 Conversie

In dit deel van de case study wordt de conversie van de dataset naar een event log in het XES-formaat uitgevoerd. De conversie wordt gedaan op basis van de hierboven beschreven conversiedefinitie. Omwille van de gekende problemen die optreden wanneer XESame gebruikt wordt in een 64-bit versie van Windows [1], is er voor gekozen om de conversie uit te voeren op een virtuele 32-bit Windows XP SP3. Dit virtueel platform wordt opgezet via Microsoft Virtual PC op een Windows 7 64-bit besturingssysteem. Het virtuele besturingssysteem wordt voorzien van 1024 MB geheugen RAM.

De conversie zelf duurde op dit platform 38 minuten, 10 seconden en 63 milliseconden zoals weergegeven wordt in Figuur 21. De resulterende event log is een XES-bestand met een grootte van 21481 KB en bestaat uit 526126 lijnen data. Een fragment van deze log is te zien in Figuur 22.



Figuur 21: Overzicht conversie

```
<trace>
    <string key="concept:name" value="Person 0"/>
    <event>
        <string key="concept:instance" value="0"/>
        <date key="time:timestamp" value="1970-01-01T03:00:00.000+01:00"/>
        <string key="lifecycle:transition" value="start"/>
        <string key="concept:name" value="Activity: Being at home"/>
    </event>
    <event>
        <string key="concept:instance" value="0"/>
        <date key="time:timestamp" value="1970-01-01T13:11:00.000+01:00"/>
        <string key="lifecycle:transition" value="complete"/>
        <string key="concept:name" value="Activity: Being at home"/>
    </event>
</trace>
```

Figuur 22: Fragment van de event log

In dit fragment zijn twee events te zien die uitgevoerd werden op het trace-object van de procesinstantie 'Person 0'. De events geven het begin en het einde aan van de activiteit 'Being at home'. De activiteit begint om 03:00u zoals aangegeven in het beginevent dat herkenbaar is door de waarde 'start' in het attribuut 'transition' en eindigt om 13:11u zoals aangegeven in het eindevent. Het is belangrijk om te vermelden dat de code in Figuur 22 handmatig aangepast is om de syntax van de event log correct weer te geven. Zoals in Tabel 7 kan worden afgelezen, onderneemt de testpersoon met het identificatienummer 0 immers vijf activiteiten gedurende de bijgehouden dag. Bovendien staat in de tabel 'Predicted_Journeys' dat deze persoon ook nog eens vier verplaatsingen doet zodat, indien deze activiteiten en verplaatsingen geconverteerd zouden worden naar begin- en eindevents, er achttien events voor dit trace-object zouden bestaan. Om het fragment overzichtelijk te houden werd er echter voor gekozen om slechts de eerste twee events weer te geven en het trace-object vervolgens handmatig af te sluiten.

Andere delen van de event log zijn terug te vinden in de bijlagen.

4.6 Analyse

In deze sectie zal onderzocht worden of de event log die hierboven gegenereerd werd bruikbaar is om process mining op toe te passen. Daarom zal de event log ingeladen worden in het ProM framework en het tabblad 'log summary' worden geraadpleegd. Dit overzicht kan enkel worden weergegeven als de event log de vereiste structuur heeft die van een XES-bestand verwacht wordt. Ook zullen er enkele process mining algoritmen op de event log worden toegepast om te kijken of ook zij met de event log overweg kunnen.

4.6.1 Onderzoek van de event log

Na het inlezen van de event log in ProM 6, is in Figuur 23 een overzicht weergegeven over de inhoud van de log. Aangezien de applicatie dit overzicht kan genereren, kan er worden afgeleid dat de structuur van de event log voldoet aan de vorm van een log in het XES-formaat. In het overzicht kan worden afgelezen dat de event log handelt over één proces zoals vereist is volgens de definitie van een event log die in deze thesis gebruikt wordt. Er wordt ook weergegeven dat er 6000 *cases* of procesinstanties als trace-object gedefinieerd zijn. Dit zijn alle testpersonen die in de tabel 'Predicted_Persons' vervat zitten. Het volgende kengetal dat afgelezen kan worden is het aantal events dat er in de log zitten, namelijk 84740. Ook dit kan worden nageteld door alle records in de tabellen 'Predicted_Activities' en 'Predicted_Journeys' op te tellen en vervolgens te verdubbelen. Iedere activiteit en verplaatsing wordt in deze event log immers opgesplitst en voorgesteld als twee events.

Figuur 23: ProM 6 Dashboard

Met het aantal *Event classes* wordt het aantal verschillende combinaties tussen de events en de transactiestatussen bedoeld. Het getal 26 wordt bekomen door de negen verschillende activiteitcategorieën op te tellen met de vier verschillende verplaatsingscategorieën. Dat getal moet vervolgens verdubbeld worden aangezien de events twee statussen kunnen aannemen, namelijk 'start' en 'complete'.

In de bovenste grafiek van Figuur 23 is weergegeven dat iedere procesinstantie in de event log minimaal uit 2 events bestaat. Dit is een logische vaststelling aangezien iedere activiteit of verplaatsing die een testpersoon onderneemt in de event log wordt opgesplitst in een begin- en eindevent. Zelfs voor personen die slechts één activiteit of verplaatsing uitvoeren zullen er dus steeds twee events in het bijhorende trace-object worden opgeslagen. Het maximale aantal events dat in een trace-object in deze event log is waargenomen is 74. Gemiddeld zitten er veertien events

per procesinstantie in de event log wat overeenkomt met gemiddeld zeven activiteiten of verplaatsing per testpersoon.

In de onderste grafiek wordt het aantal verschillende *event classes* per procesinstantie voorgesteld. Omdat het minimum, maximum en gemiddelde van deze grafiek verschilt van de bovenstaande grafiek, kan er worden afgeleid dat sommige *event classes* herhaald worden binnen hetzelfde trace-object. Dit fenomeen, divergentie genaamd, kan problemen opleveren voor sommige process mining algoritmen zoals reeds werd uitgelegd in sectie 3.5.

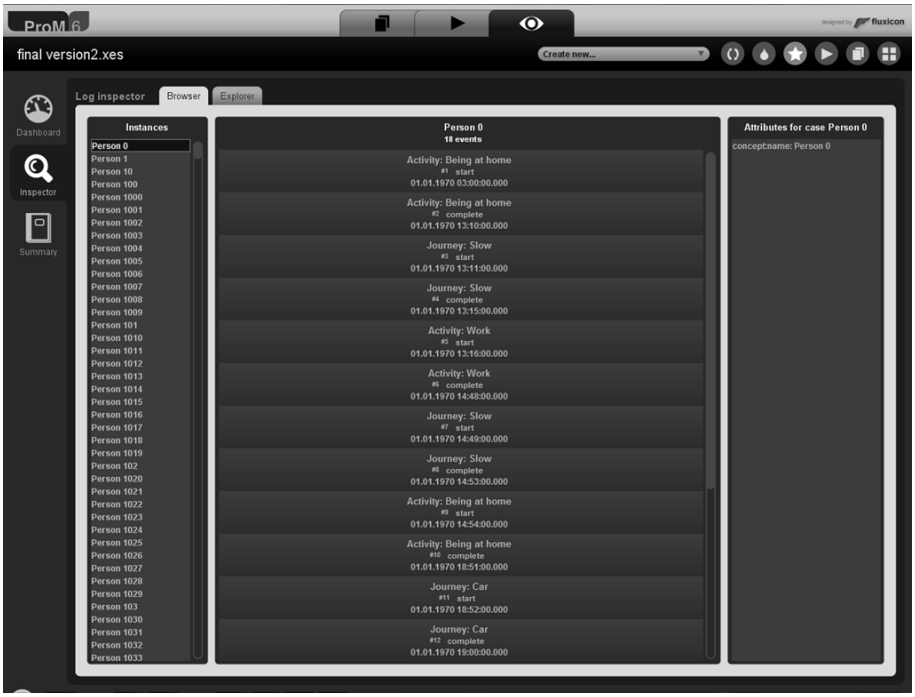
Figuur 24: ProM 6 tabblad Inspector

Figuur 24 geeft het tabblad 'Inspector' weer van de samenvatting van de event log. In dit tabblad worden alle events van de verschillende procesinstanties weergegeven in chronologische volgorde. Hier wordt duidelijk dat er een kleine fout in de inhoud van de event log is geslopen. Iedere activiteit of verplaatsing kan immers pas starten nadat de vorige afgerond is. In de volgorde die weergegeven wordt in Figuur 24 is dit niet het geval. Het vierde event bijvoorbeeld, dat de start van de activiteit

'Work' aangeeft, gebeurt volgens deze event log immers eerder dan het event dat het einde van de verplaatsing ervoor aangeeft. Dit zou problemen kunnen opleveren bij het uitvoeren van process mining algoritmen die een procesmodel opstellen.

De oorzaak van dit probleem ligt in het feit dat een persoon in dit proces nooit twee dingen tegelijkertijd kan doen. Volgens de data die gebruikt is voor het maken van deze event log is dit wel het geval. Om het probleem op te lossen moet er worden teruggegaan tot de preprocessingfase. De eindtijdstippen van de activiteiten en verplaatsing kunnen hier zo gedefinieerd worden dat ze één tijdseenheid vroeger eindigen.

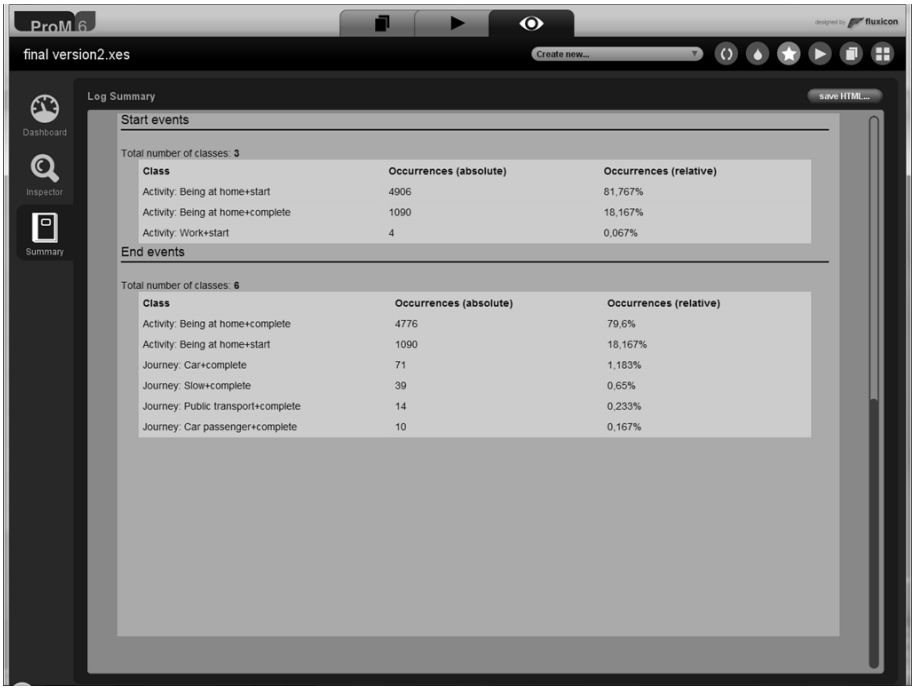
Het resultaat van deze ingreep is te zien in Figuur 25. Zoals af te lezen valt in deze figuur, wordt iedere activiteit of verplaatsing die gestart wordt eerst afgesloten alvorens een nieuwe begint.



Figuur 25: Samenvatting van de aangepaste event log

Een volgende controle van de event log kan gedaan worden aan de hand van de informatie op het tabblad 'Summary'. Dit tabblad wordt weergegeven in Figuur 26 en geeft onder andere de verschillende events weer die als eerste worden uitgevoerd voor een trace-object. De informatie die hieruit verkregen wordt is op zijn minst opmerkelijk. Het blijkt dat in de aangepaste versie van de event log 4906 van de 6000 *traces* het beginevent van de activititeit 'Being at home' als eerste event hebben. Van dit tabblad kan echter ook worden afgelezen dat 1090 van de 6000 *traces* het eindevent van die activiteit als eerste event hebben, wat natuurlijk onmogelijk is. Het eerste event van de dag moet uiteraard een event met als status 'start' hebben.

Opheldering voor dit probleem kan gevonden worden in de dataset. Het feit dat een activiteit met als status 'complete' zich heeft kunnen nestelen voor het startevent van diezelfde activiteit zou willen zeggen dat er activiteiten met een negatieve tijdsduur in de oorspronkelijke database zijn geslopen. Bij het berekenen van de eindtijdstippen voor die verschillende activiteiten zou dit ervoor gezorgd hebben dat deze tijdstippen vroeger zijn dan de begintijdstippen. Door de aanpassing die eerder in deze sectie besproken werd, namelijk het vervroegen van de eindevents met één tijdseenheid, zullen ook die activiteiten met een niet bestaande tijdsduur nu opduiken in de groep met foute events. Dit is echter niet erg, aangezien activiteiten met een niet bestaande tijdsduur ook niet realistisch zijn. Activiteiten worden immers uitgevoerd of niet uitgevoerd. Activiteiten die op papier bestaan, maar geen tijd in beslag nemen kunnen dan ook niet. Om het process mining project uit te kunnen voeren moet de dataset bijgevolg aangepast worden.



Figuur 26: ProM 6 tabblad Summary

Daarom wordt de dataset onderzocht op alle activiteiten met een negatieve of niet bestaande tijdsduur en worden deze activiteiten uit de dataset gefilterd. Ook activiteiten met een duur van langer dan één dag worden uit de dataset gefilterd, aangezien de assumptie was dat alle activiteiten van één dag moeten worden bijgehouden in de log. De tabel 'Predicted_Activities' werd vervolgens onderzocht op dezelfde soort van fouten maar deze kwamen niet voor in deze tabel.

In het tabblad 'Summary' van de log die gemaakt werd met deze aangepaste dataset in Figuur 27 kan worden gevonden dat alle eerste events nu het status 'start' hebben. Deze event log is bijgevolg klaar om enkele process mining algoritmen op toe te passen.

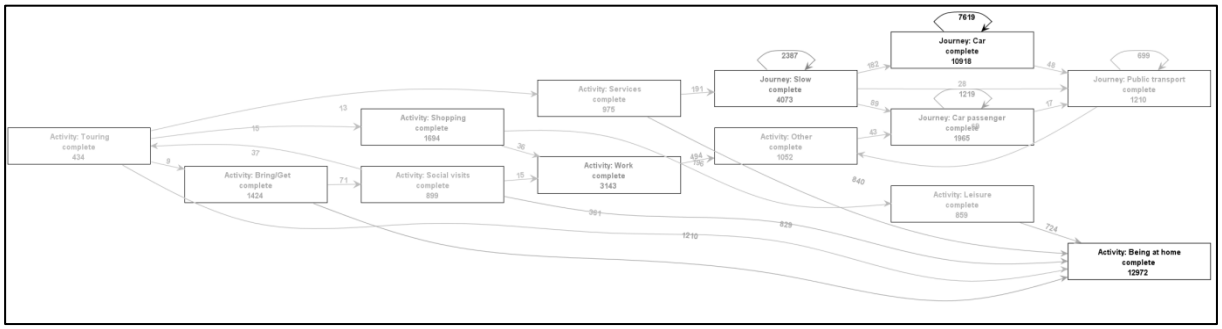
Figuur 27: Tabblad Summary van de aangepaste log

4.6.2 Process mining algoritmen

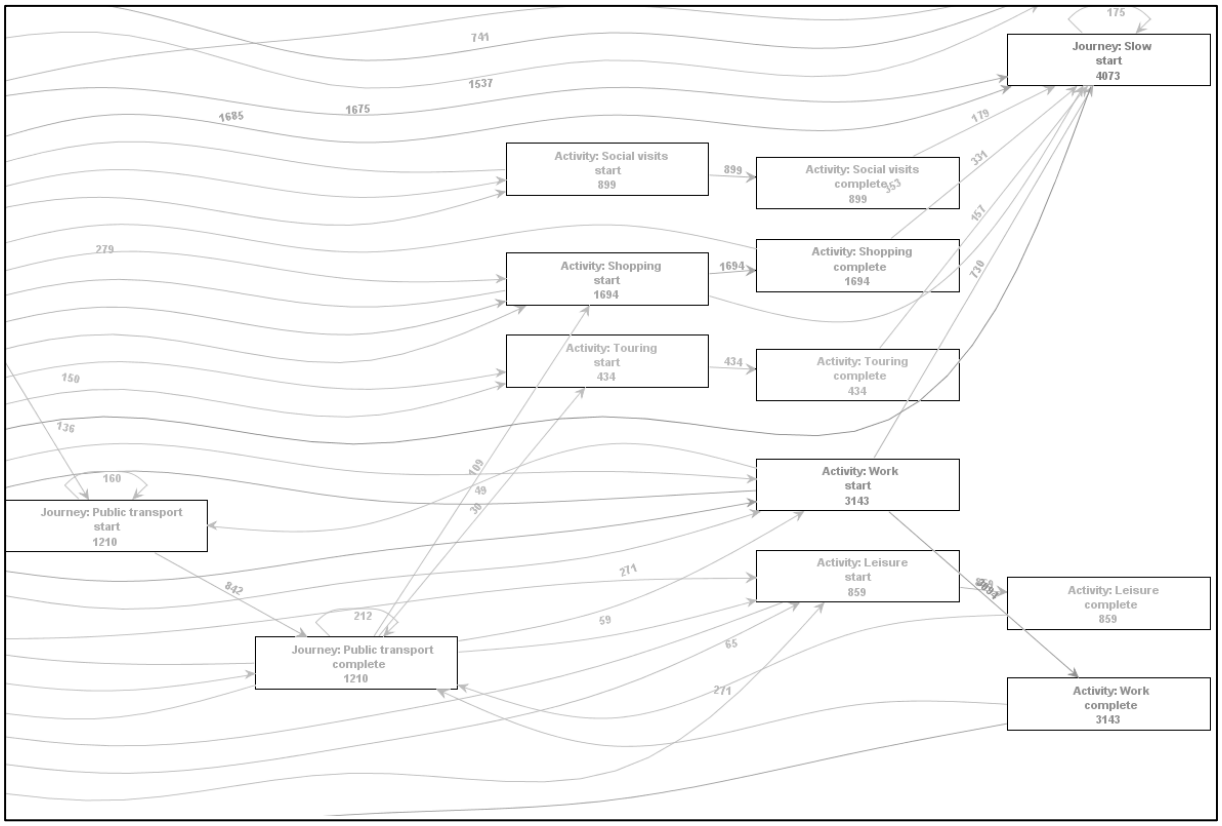
Een eerste algoritme dat wordt gebruik is de 'Simple Log Filter'. Met dit algoritme kunnen er bepaalde gedeeltes van de event log geselecteerd worden en andere delen weggelaten. Er wordt voor gekozen om een event log te selecteren waarin enkel de events met status 'complete' in vervat zitten. Een dergelijke event log is handiger om proces modellen op te stellen aangezien iedere activiteit en verplaatsing slechts eenmaal in de event log zit en er dus geen overbodige informatie in de modellen zal zitten. In Figuur 28 kan worden afgelezen dat de resulterende event log 41618 events telt. Dit is de helft van het oorspronkelijke aantal.

Figuur 28: Resultaat Simple Log Filter

De volgende twee figuren, Figuur 29 en Figuur 30, geven twee modellen weer die werden opgesteld aan de hand van het algoritme 'Heuristic Miner'. Het eerste model werd gebouwd op basis van de aangepaste, gefilterde log en het tweede model op basis van de oorspronkelijke. Het tweede model is veel ingewikkelder aangezien alle activiteiten en verplaatsingen er dubbel in verwerkt zijn. In deze figuren geldt dat hoe donkerder de tekst in de rechthoeken die de events voorstellen, hoe frequenter deze events in de event log aanwezig zijn.



Figuur 29: Heuristic model op basis van de gefilterde log

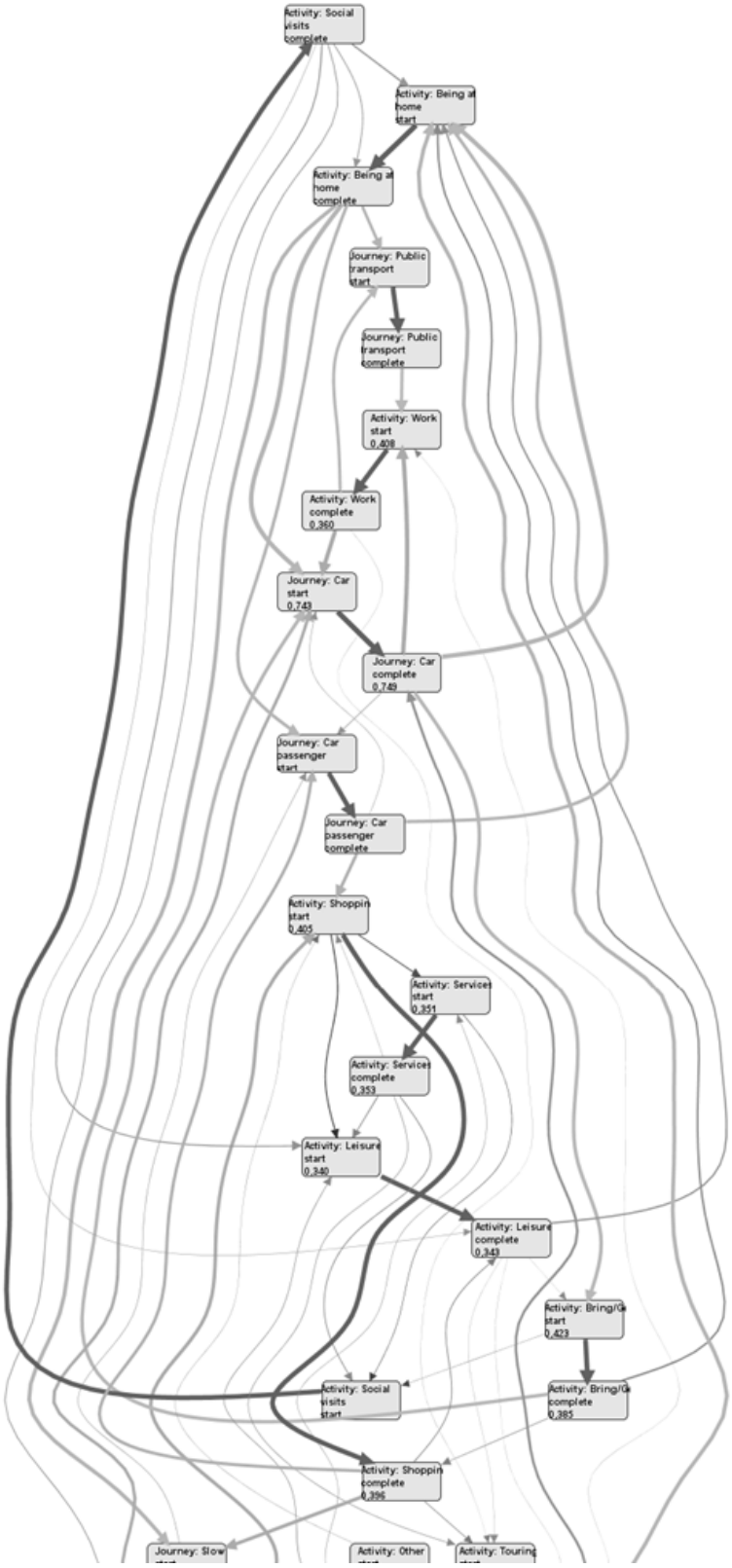


Figuur 30: Fragment van het Heuristic model op basis van de oorspronkelijke log

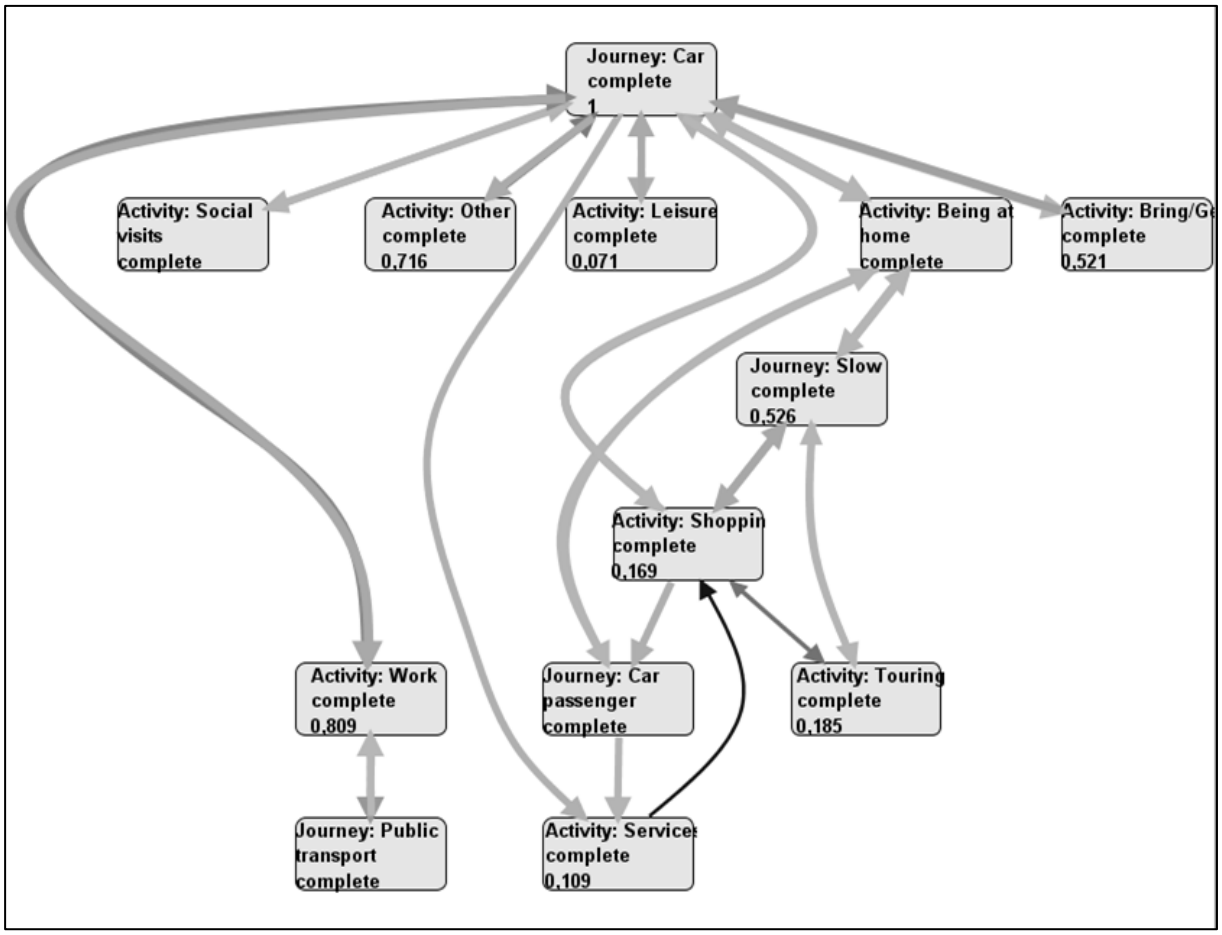
Het volgende algoritme dat werd uitgevoerd is het algoritme 'Fuzzy Miner'. Deze plug-in stelt een model op waarin de relaties tussen alle events, al dan niet geclusterd, worden weergegeven. Dit algoritme is losser gedefinieerd zodat het betere modellen kan opstellen bij event logs van ongestructureerde processen dan vroeger ontwikkelde algoritmen zoals het alfa-algoritme en de 'Heuristic Miner'. Deze algoritmen werden ontwikkeld voor kunstmatige event logs die moesten voldoen aan twee assumpties: ten eerste, alle events in de event log zijn betrouwbaar en hebben een overeenkomstige activiteit in het proces; en ten tweede, er bestaat een exact proces dat teruggevonden kan worden in de inhoud van de event log [22]. Het proces dat gedefinieerd is in de event log die in deze case study werd opgebouwd is echter een ongestructureerd proces. De tijdsbesteding van ieder testpersoon is namelijk verschillende zodat er geen algemene vaste

structuur in het proces gevonden kan worden. Het 'Fuzzy Miner'-algoritme is precies ontworpen voor dit soort processen en is in staat om een overzichtelijker model weer te geven dan de voorgaande algoritmen door eventueel sommige minder frequente of belangrijkere events weg te laten of samen te voegen. Dit algoritme werd uitgevoerd op de originele, ongefilterde event log en het bekomen resultaat is weergegeven in Figuur 31. De dikte van de pijlen tussen twee events geeft de frequentie aan van iedere afzonderlijke relatie. Het zal niet verbazen dat er tussen het beginevent en het eindevent van dezelfde activiteiten en verplaatsingen steeds een dikke pijl getekend is.

Vervolgens werd dit algoritme ook uitgevoerd op de gefilterde log. In Figuur 32 kan het resultaat van deze bewerking gevonden worden. Deze figuur geeft een duidelijker beeld van het ongestructureerde proces aangezien enkel de eindevents van de verschillende activiteiten en verplaatsingen erin opgenomen zijn. Het is bijvoorbeeld duidelijk dat de auto het vervoermiddel bij uitstek, en voor sommige activiteiten zelfs het enige, is dat gebruikt wordt om zich van en naar de meest uiteenlopende activiteiten te begeven. Het model geeft ook weer dat het openbaar vervoer de enige andere transportwijze is die gebruikt wordt om van en naar het werk te gaan. Deze voorbeelden tonen aan dat er wel degelijk informatie over dit ongestructureerd proces gevonden kan worden aan de hand van process mining.



Figuur 31: Procesmodel van ongefilterde log met Fuzzy miner plug-in



Figuur 32: Procesmodel van gefilterde log met Fuzzy miner plug-in

Deel IV: Conclusie

5. Besluit

In deze thesis werd een belangrijk deelproces van process mining onderzocht, namelijk het opstellen van een adequate event log die als input kan dienen voor het eigenlijk process mining project. Om de constructie van een event log die aan alle vereisten van het XES-formaat voldoet te vergemakkelijken, werd aan de Technische Universiteit Eindhoven het XESMa- of XESame-framework gemaakt door ing. J.C.A.M. Buijs. XESame laat de gebruiker toe om te bepalen welke data er in de event log moet worden opgenomen en transformeert deze data in een event log in de juiste vorm. Het doel van deze thesis was om de werking van dit framework uit te testen en te valideren. De centrale onderzoeksvraag luidde dan ook:

> *"Hoe kan men aan de hand van een onafhankelijke case study de werking van het XESMa-framework valideren?"*

Alvorens de case study uitgevoerd werd, werd de lezer aan de hand van een literatuurstudie geïnformeerd over de oorsprong, het doel, de werking en de vorm van process mining. Process mining werd gesitueerd als een onderdeel van business process management (BPM) en heeft als doel om informatie over processen van process-aware informatiesystemen (PAIS) te vergaren aan de hand van logs die bijgehouden worden over de bewerkingen en activiteiten die op de data in zo een systeem worden uitgevoerd. Er werd uitgelegd dat er verschillende perspectieven bestaan waaruit een process mining project kan worden ondernomen en dat de keuze voor een bepaald perspectief een invloed heeft op de data die vereist is in de event log. Vervolgens werden er, zoals in de eerste deelvraag gespecificeerd werd, andere aspecten onderzocht die een invloed hebben op de selectie

van de data voor in de log en werden de belangrijkste conversiebeslissingen die genomen moeten worden besproken. Deze deelvraag luidde immers:

> *"Wat zijn de verschillende aspecten waar rekening mee moet worden gehouden bij het definiëren van een conversiedefinitie en hoe beïnvloeden deze aspecten de conversiedefinitie?"*

Na de literatuurstudie werd er in de case study een probleemscenario geschetst om een event log te definiëren voor process mining. De verschillende fasen in het conversieproces, namelijk het dataonderzoek, de preprocessing, de conversiedefinitie en de eigenlijke conversie, werden vervolgens doorlopen om een volwaardige event log in het XES-formaat te bekomen en de tweede deelvraag,

> *"Hoe kan men deze aspecten gebruiken om conversiescenario's op te stellen om de validiteit van het framework te testen?"*

te beantwoorden.

Nu restte er enkel nog een manier om de verkregen event log op zijn bruikbaarheid te testen om zo het framework te valideren zoals in de laatste deelvraag werd gevraagd. Deze vraag werd beantwoord door de event log in het ProM-framework in te lezen en er verschillende process mining algoritmen op los te laten. Uit de verkregen modellen was het mogelijk om nieuwe informatie over de processen gemakkelijk te verkrijgen. De event log slaagde dus voor de test van de derde deelvraag.

Bijgevolg kan er besloten worden dat het doel van XESame, namelijk het mogelijk maken om een conversie van een databron naar een event log te definiëren en uit te voeren met zo weinig mogelijk programmeerervaring, bereikt is en de centraleonderzoeksvraag van deze thesis dan ook

beantwoord werd. Het is perfect mogelijk om een event log te definiëren en de conversie vervolgens uit te voeren met het XESame-framework.

Lijst van de geraadpleegde werken

1. J.C.A.M. Buijs, W.M.P. van der Aalst, H.M.W. Verbeek, G.H.L. Fletcher (2010). *Mapping Data Sources to XES in a Generic Way, Master Thesis.*
2. L. Wen, J. Wang, W.M.P. van der Aalst, B. Huang, J. Sun (2009). *A novel approach for process mining based on event types.* J Intell Inf Syst 32: 163–190.
3. W.M.P. van der Aalst, H.A. Reijers, A.J.M.M. Weijters, B.F. van Dongen, A.K. Alves de Medeiros, M. Song, H.M.W. Verbeek (2007). *Business process mining: An industrial application.* Information Systems 32: 713–732.
4. M. Dumas, W.M.P. van der Aalst, A.H.M. ter Hofstede (2005). *Process-Aware Information Systems: Bridging People and Software through Process Technology*, hoofdstuk 1, p3-20, hoofdstuk 10, p235-255. Wiley-Interscience, Hoboken, New Jersey, USA.
5. W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, A.J.M.M. Weijters (2003). *Workflow mining: A survey of issues and approaches*. Data & Knowledge Engineering 47: 237-267.
6. W.M.P. van der Aalst, A.J.M.M. Weijters (2004). *Process mining: a research agenda.* Computers in Industry 53: 231-244.
7. J. Caverlee, J. Bae, Q. Wu, L. Liu, C. Pu, W.B. Rouse (2007). *Workflow management for enterprise transformation.* Information Knowledge Systems Management 6: 61-80.
8. M. Weske, W.M.P. van der Aalst, H.M.W. Verbeek (2004). *Advances in business process management.* Data & Knowledge Engineering 50: 1-8.
9. M. Wang, H. Wang (2006). *From process logic to business logic - A cognitive approach to business process management.* Information & Management 43: 179-193.
10. B.F. van Dongen, A.K.A. de Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters, W.M.P. van der Aalst (2005). *The ProM framework: A new era in process mining tool support.*
11. W.M.P. van der Aalst (2005). *Business alignment: using process mining as a tool for Delta analysis and conformance testing.* Requirements Engineering 10: 198-211.
12. B.F. van Dongen, W.M.P. van der Aalst (2005). *A Meta Model for Process Mining Data.* Conference on Advanced Information Systems Engineering 161, Porto, Portugal.
13. C.W. Günther (2009). *Xes Standard Definition.* Fluxicon Process Laboratories, November 2009.

14. W.M.P. van der Aalst, H.A. Reijters, M. Song (2005). *Discovering Social Networks from Event Logs.* Computer Supported Cooperative Work 14: 549-593.
15. A.J.M.M. Weijters, W.M.P. van der Aalst (2001). *Process Mining: Discovering Workflow Models from Event-Based Data.* Proceedings of the ECAI Workshop on Knowledge Discovery and Spatial Data: 283-290.
16. W.M.P van der Aalst, B.F. van Dongen (2002). *Discovering Workflow Performance Models from Timed Logs.* Proceedings of the First International Conference on Engineering and Deployment of Cooperative Information Systems: 45-63.
17. A.J.M.M. Weijters, W.M.P. van der Aalst (2003). *Rediscovering Workflow Models from Event-Based Data using Little Thumb.* Integrated Computer-Aided Engineering 10: 151-162.
18. W.M.P. van der Aalst, M. Song (2004). *Mining Social Networks: Uncovering interaction patterns in business processes.* Computer Sciences 3080: 244-260.
19. W.M.P. van der Aalst, H.A. Reijeters, M. Song (2005). *Discovering Social Networks from Event Logs*. Computer Supported Cooperative Work 14: 549-593.
20. W.M.P van der Aalst, A.J.M.M. Weijters (2003). *Kleinduimpje in Workflowland.* Management & Informatie 11: 4.
21. C.W. Günther, W.M.P. van der Aalst (2007). *Fuzzy Mining - Adaptive Process Simplification Based on Multi-Perspective Metrics*, Proceedings of the 5th international conference on Business process management.

Bijlagen

	A	B	C	D	E	F
1	Activity_id	person_id	activity_ty	activity_bt	activity_duration	activity_et
2	0	0	0	1/01/1970 3:00:00	611	1/01/1970 13:10:59
3	1	0	1	1/01/1970 13:16:00	93	1/01/1970 14:48:59
4	3	0	0	1/01/1970 14:54:00	238	1/01/1970 18:51:59
5	5	0	8	1/01/1970 19:01:00	225	1/01/1970 22:45:59
6	7	0	0	1/01/1970 22:55:00	245	2/01/1970 2:59:59
7	9	1	0	1/01/1970 3:00:00	754	1/01/1970 15:33:59
8	10	1	3	1/01/1970 15:43:00	5	1/01/1970 15:47:59
9	12	1	0	1/01/1970 15:57:00	43	1/01/1970 16:39:59
10	14	1	3	1/01/1970 16:57:00	14	1/01/1970 17:10:59

Bijlage 1: Fragment 'Predicted_Activities'

	A	B	C	D	E	F
1	Journey_id	person_id	journey_bt	journey_total_duration	journey_et	journey_mode
2	2	0	1/01/1970 13:11	5	1/01/1970 13:15	3
3	4	0	1/01/1970 14:49	5	1/01/1970 14:53	3
4	6	0	1/01/1970 18:52	9	1/01/1970 19:00	1
5	8	0	1/01/1970 22:46	9	1/01/1970 22:54	1
6	11	1	1/01/1970 15:34	9	1/01/1970 15:42	1
7	13	1	1/01/1970 15:48	9	1/01/1970 15:56	1
8	15	1	1/01/1970 16:40	17	1/01/1970 16:56	1
9	17	1	1/01/1970 17:11	17	1/01/1970 17:27	1
10	20	2	1/01/1970 7:25	19	1/01/1970 7:43	1

Bijlage 2: Fragment 'Predicted_Journeys'

	A	B	C	D	E	F
1	person_id	household_id	Age	work_status	gender	IsDriv
2	0	0	1	2	2	1
3	1	1	3	0	1	1
4	2	2	3	2	1	1
5	3	3	2	2	1	1
6	4	4	2	2	1	1
7	5	5	2	2	1	1
8	6	6	2	2	1	0
9	7	7	1	0	1	1
10	8	8	2	2	2	1

Bijlage 3: Fragment 'Predicted_Persons'

	A	B
1	activity_id	activity_value
2	0	Being at home
3	1	Work
4	3	Bring/Get
5	4	Shopping
6	6	Services
7	7	Social visits
8	8	Leisure
9	9	Touring
10	10	Other

Bijlage 4: Fragment 'Activity'

	A	B
1	age_id	age_value
2	1	-35
3	2	35-55
4	3	55-65
5	4	65-75
6	5	75

Bijlage 5: Fragment 'Age'

	A	B
1	gender_id	gender_value
2	1	male
3	2	female

Bijlage 6: Fragment 'Gender'

	A	B
1	journey_id	journey_value
2	1	Car
3	3	Slow
4	4	Public transport
5	6	Car passenger

Bijlage 7: Fragment 'Journey'

	A	B
1	work_id	work_value
2	0	no work
3	1	part time
4	2	full time

Bijlage 8: Fragment 'Work'

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- This file has been generated with the OpenXES library. It conforms -->
<!-- to the XML serialization of the XES standard for log storage and -->
<!-- management. -->
<!-- XES standard version: 1.0 -->
<!-- OpenXES library version: 1.0RC7 -->
<!-- OpenXES is available from http://www.openxes.org/ -->
<log xes.version="1.0" xes.features="nested-attributes" openxes.version="1.0RC7" xmlns="http://www.xes-standard.org/">
    <extension name="Lifecycle" prefix="lifecycle" uri="http://www.xes-standard.org/lifecycle.xesext"/>
    <extension name="Organizational" prefix="org" uri="http://www.xes-standard.org/org.xesext"/>
    <extension name="Time" prefix="time" uri="http://www.xes-standard.org/time.xesext"/>
    <extension name="Concept" prefix="concept" uri="http://www.xes-standard.org/concept.xesext"/>
    <extension name="Semantic" prefix="semantic" uri="http://www.xes-standard.org/semantic.xesext"/>
    <global scope="trace">
        <string key="concept:name" value="UNKNOWN"/>
    </global>
    <global scope="event">
        <string key="concept:instance" value="UNKNOWN"/>
        <date key="time:timestamp" value="1970-01-01T00:00:00.000+01:00"/>
        <string key="lifecycle:transition" value="UNKNOWN"/>
        <string key="concept:name" value="UNKNOWN"/>
    </global>
    <classifier name="Activity classifier" keys="concept:name lifecycle:transition"/>
    <string key="author" value="Niels Vandenreyt"/>
    <string key="processname" value="Time Schedule"/>
    <string key="description" value="Time Schedule als proces met de testpersonen als trace-objecten en de voorspelde
    <string key="lifecycle:model" value="standard"/>
    <date key="date" value="2011-05-04T03:44:09.000+02:00"/>
    <string key="concept:name" value="Mobiliteitslog"/>
    <string key="datasource" value="Applicatienaam"/>
    <string key="version" value="1.0"/>
```

Bijlage 9: Log-element van de event log

```
<trace>
    <string key="concept:name" value="Person 0"/>
    <event>
        <string key="concept:instance" value="0"/>
        <date key="time:timestamp" value="1970-01-01T03:00:00.000+01:00"/>
        <string key="lifecycle:transition" value="start"/>
        <string key="concept:name" value="Activity: Being at home"/>
    </event>
    <event>
        <string key="concept:instance" value="0"/>
        <date key="time:timestamp" value="1970-01-01T13:11:00.000+01:00"/>
        <string key="lifecycle:transition" value="complete"/>
        <string key="concept:name" value="Activity: Being at home"/>
    </event>
</trace>
```

Bijlage 10: Trace-element van de event log

Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:
Een process mining case study

Richting: **master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica-informatie- en communicatietechnologie**
Jaar: **2011**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

Vandenreyt, Niels

Datum: **31/05/2011**