

2013•2014
FACULTEIT BEDRIJFSECONOMISCHE WETENSCHAPPEN
*master in de toegepaste economische wetenschappen:
handelsingenieur in de beleidsinformatica*

Masterproef

Het verbeteren van de beheerbaarheid van bedrijfsprocessen: een framework voor het middle management

Promotor :
Prof. dr. Koenraad VANHOOF

Jan Huskens

Proefschrift ingediend tot het behalen van de graad van master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica

2013•2014

FACULTEIT BEDRIJFSECONOMISCHE
WETENSCHAPPEN

*master in de toegepaste economische wetenschappen:
handelsingenieur in de beleidsinformatica*

Masterproef

Het verbeteren van de beheerbaarheid van
bedrijfsprocessen: een framework voor het middle
management

Promotor :
Prof. dr. Koenraad VANHOOF

Jan Huskens

*Proefschrift ingediend tot het behalen van de graad van master in de toegepaste
economische wetenschappen: handelsingenieur in de beleidsinformatica*

WOORD VOORAF

Deze masterproef vormt het sluitstuk van mijn opleiding Handelsingenieur in de beleidsinformatica aan de Universiteit Hasselt en handelt over het verbeteren van de beheerbaarheid van bedrijfsprocessen. Langs deze weg zou ik een aantal mensen willen bedanken die mij geholpen hebben dit werk tot een goed einde te brengen.

Ten eerste zou ik mijn promotor, prof. dr. Koen Vanhoof, willen bedanken voor zijn advies en begeleiding tijdens het tot stand komen van deze masterproef. Daarnaast ook een woord van dank aan de studenten die hun medewerking verleend hebben aan de gevalstudie die deel uitmaakt van deze masterproef. Zonder hen zou het niet mogelijk geweest zijn de theoretische concepten aan de praktijk te toetsen.

Jan Huskens

Peer, 27 mei 2014

SAMENVATTING

Elk bedrijf zet via bedrijfsprocessen, i.e. een geheel van geordende activiteiten, input om in output. Door steeds toenemende concurrentie en verhoogde kwaliteitseisen, is het belangrijk dat deze omzetting zo effectief en efficiënt mogelijk gebeurt. Om dit te verzekeren, is het belangrijk dat processen beheerbaar zijn. Daarom is er nood aan een framework dat het mogelijk maakt om de beheerbaarheid van processen te verbeteren. In deze masterproef worden drie deelaspecten van beheerbaarheid onderscheiden: controle, monitoring en sturing. Een beheerbaar proces wordt dus gezien als een proces dat controleerbaar, meetbaar en flexibel is. Voor elk van bovengenoemde deelaspecten wordt besproken wat het betreffende deelaspect juist inhoudt en hoe het verbeterd kan worden.

Een proces is controleerbaar wanneer het voorspelbaar is. Controle tracht de variantie in tijd, kosten en kwaliteit van het proces zoveel mogelijk te beperken. Er worden negen manieren besproken om de controle over een proces te verbeteren: het vereenvoudigen van processen, het elimineren van taken, het samenvoegen en opsplitsen van taken, het verplaatsen van taken, het gelijktijdig uitvoeren van taken, het toevoegen van controletaken, het minimaliseren van het aantal betrokken afdelingen, het optimaliseren van informatie-uitwisseling en het optimaliseren van de resources.

Hoewel bij het ontwerp van het proces zoveel mogelijk controle wordt ingebouwd om de variantie in tijd, kwaliteit en kosten te beperken, kan nooit alle variantie geëlimineerd worden. Om afwijkingen in het proces te kunnen vaststellen en er vervolgens op te reageren, is het belangrijk dat processen gemonitord kunnen worden. Dit kan door het formuleren van zogenaamde “Process Performance Indicators” (PPI’s). Om het ook mogelijk te maken om PPI’s in een procesmodel te integreren, wordt een uitbreiding van BPMN 2.0 besproken die dit toestaat.

Wanneer tijdens het monitoren van een proces bepaalde afwijkingen ten opzichte van het gewenste gedrag worden vastgesteld, is het belangrijk dat er ingegrepen kan worden. Daarom moeten processen kunnen omgaan met veranderingen zonder dat hiervoor het procesmodel moet worden aangepast. Het proces moet dus flexibel zijn. Het creëren van flexibiliteit door ontwerp, i.e. het door ontwerp mogelijk maken te reageren op veranderingen, kan met behulp van verschillende “workflow patterns”: “choice”, “iteration”, “interleaved routing”, “multiple instances activities” en “cancelation”. Het gebruik van “workflow patterns” om een proces flexibel te maken heeft echter een nadeel: met de flexibiliteit stijgt ook de complexiteit van het proces. Er wordt een methode besproken die dit probleem het hoofd biedt door toe te staan dat het procesmodel waar nodig wordt aangepast tijdens de uitvoering.

In een gevalstudie worden de hierboven beschreven theoretische concepten toegepast. Ze is opgebouwd als volgt: in een eerste fase wordt aan studenten gevraagd een aantal processen te modelleren, meer specifiek de processen voor een autoverhuurbedrijf. Er wordt hen niet gevraagd met bepaalde aspecten zoals controle, monitoring of sturing rekening te houden. In een tweede fase wordt aan de studenten het belang van deze concepten bij het modelleren van processen uitgelegd, en wordt hen vervolgens gevraagd hun procesmodellen aan te passen, rekening houdend met deze concepten. Aan de hand van de resultaten van de gevalstudie worden voor elk deelaspect van beheerbaarheid toepassingen besproken om het desbetreffende deelaspect te verbeteren en worden conclusies getrokken over de implementeerbaarheid van de verschillende deelaspecten.

INHOUDSOPGAVE

WOORD VOORAF

SAMENVATTING

INHOUDSOPGAVE

HOOFDSTUK 1: INLEIDING	9
1.1 PROBLEEMSTELLING	9
1.2 ONDERZOEKSVRAAG	10
1.2.1 Deelvraag 1: Wat zijn de deelaspecten van beheerbaarheid?	10
1.2.2 Deelvraag 2: Hoe kunnen de deelaspecten verbeterd worden?	10
1.2.3 Deelvraag 3: in welke mate zijn de verschillende verbeteringen implementeerbaar?	10
1.3 ONDERZOEKSMETHODE	11
1.3.1 Literatuurstudie	11
1.3.2 Case study	11
HOOFDSTUK 2: BEHEERBAARHEID	13
HOOFDSTUK 3: CONTROLE	15
3.1 EENVOUDIGE PROCESSEN	16
3.2 ELIMINEREN VAN TAKEN	18
3.3 SAMENVOEGEN EN OPSPLITSEN VAN TAKEN	20
3.4 VERPLAATSEN VAN TAKEN	21
3.5 GELIJKTijdige UITVOERING VAN TAKEN	22
3.6 TOEVOEGEN VAN CONTROLETAKEN	23
3.7 BRONOPTIMALISATIE	24
3.8 INFORMATIEOPTIMALISATIE	26
3.9 MINIMALISEREN VAN HET AANTAL BETROKKEN AFDELINGEN	27
HOOFDSTUK 4: MONITORING	29
4.1 PROCESS PERFORMANCE INDICATORS	30
4.2 EXTENDED BPMN	31
HOOFDSTUK 5: STURING	35
5.1 FLEXIBILITEIT DOOR ONTWERP: CONTROL FLOW PATTERNS	36
5.2 FLEXIBILITEIT DOOR ONTWERP: WORKFLOW ADAPTATION PATTERNS	40
HOOFDSTUK 6: GEVALSTUDIE EASY CAR	41
6.1 CONTEXT	42
6.2 OPTIMALISATIE VAN CONTROLE: TOEPASSING	43
6.2.1 Taakeliminatie	43
6.2.2 Samenvoegen en opsplitsen van taken	43
6.2.3 Verplaatsen van taken	44
6.2.4 Toevoegen van controletaken	44
6.2.5 Gelijktijdige uitvoering	44
6.2.6 Bronoptimalisatie	45
6.2.7 Informatieoptimalisatie	46
6.3 OPTIMALISATIE VAN CONTROLE: RESULTATEN	47
6.4 OPTIMALISATIE VAN MONITORING: TOEPASSING	48

6.4.1	PPI 1: Percentage wagen niet beschikbaar.....	48
6.4.2	PPI 2: Gemiddelde duur betaling.....	48
6.4.3	PPI 3: Percentage ontevreden klanten.....	49
6.4.4	PPI 4: Gemiddelde duur controle wagen.....	49
6.5	OPTIMALISATIE VAN MONITORING: RESULTATEN.....	50
6.6	OPTIMALISATIE VAN STURING: TOEPASSING.....	51
6.6.1	Keuze.....	51
6.6.2	Iteratie.....	51
6.6.3	Interleaved routing.....	51
6.6.4	Annulatie.....	52
6.7	OPTIMALISATIE VAN STURING: RESULTATEN.....	53
HOOFDSTUK 7: BESLUIT.....		55
LITERATUUR.....		57

HOOFDSTUK 1: INLEIDING

1.1 PROBLEEMSTELLING

Het beheren van bedrijfsprocessen behoort tot het domein van Business Process Management (BPM), een denkwijze die elk bedrijf ziet als een verzameling van bedrijfsprocessen die gemodelleerd, beheerd en verbeterd moeten worden (McCormack & Johnson, 2001). Business Process Management kan ook als volgt gedefinieerd worden: “Supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organisations, applications, documents, and other sources of information” (Weske, van der Aalst & Verbeek, 2004, p. 2). Business Process Management kan ook nog vanuit een ander perspectief bekeken worden: als een manier om de prestaties van een onderneming te verbeteren. Jeston en Nelis (2006) definiëren BPM als “the achievement of an organization’s objectives through the improvement, management and control of essential business processes” (p. 11). Business Process Management is dus een managementbenadering die draait om het beheren en optimaliseren van bedrijfsprocessen. Hierdoor kan de klantentevredenheid toenemen, kunnen kosten verlaagd worden en kunnen nieuwe producten en diensten aan een lagere kost ontwikkeld worden (Weske, 2007).

Een procesgeoriënteerde aanpak kan een bedrijf er dus niet alleen toe in staat stellen te besparen, maar kan er ook voor zorgen dat klanten meer waarde ervaren, wat op zijn beurt tot een hogere klantentevredenheid leidt. Dit bieden van superieure waarde aan klanten is essentieel om te kunnen concurreren in een globaliserende economie.

Aangezien het verwezenlijken van bedrijfsdoelstellingen staat of valt met effectieve en efficiënte bedrijfsprocessen, is het belangrijk dat deze processen goed beheerd worden. Om dit te bewerkstelligen is er nood aan een framework dat het mogelijk maakt om na te gaan hoe beheerbaar een proces is en hoe dit verbeterd kan worden. Meer specifiek moet worden nagegaan wat de deelaspecten van beheerbaarheid zijn en hoe deze verbeterd kunnen worden. Door het middle management te voorzien van een framework om de beheerbaarheid van een proces te verbeteren, zullen processen meer gecontroleerd verlopen en zal het management sneller kunnen ingrijpen wanneer een proces toch niet verloopt zoals gewenst. Zo zal het verwezenlijken van de bedrijfsdoelstellingen niet in gevaar komen.

1.2 ONDERZOEKSVRAAG

Zoals blijkt uit de probleemstelling, luidt de centrale onderzoeksvraag van deze masterproef: “Hoe kan de beheerbaarheid van bedrijfsprocessen verbeterd worden?”. Deze onderzoeksvraag laat zich opsplitsen in drie deelvragen: (1) “Wat zijn de deelaspecten van beheerbaarheid?” (2) “Hoe kunnen de deelaspecten verbeterd worden?” en (3) “In welke mate zijn de verschillende verbeteringen implementeerbaar?”.

1.2.1 DEELVRAAG 1: WAT ZIJN DE DEELASPECTEN VAN BEHEERBAARHEID?

Aangezien “beheerbaarheid” een ruim begrip is, moet er eerst worden afgebakend wat dit begrip juist inhoudt. Hiertoe wordt het begrip opgedeeld in drie deelaspecten, die vervolgens elk apart onderzocht worden. De volgende deelaspecten worden onderscheiden: controle, monitoring en sturing. Controle tracht het proces voorspelbaar te maken met betrekking tot tijd, kosten en kwaliteit. Monitoring heeft als doel het meten van bepaalde procesaspecten om eventuele afwijkingen van het gewenste procesgedrag vast te stellen, terwijl sturing de mogelijkheid is om te reageren op veranderingen zonder dat het procesmodel moet worden aangepast.

1.2.2 DEELVRAAG 2: HOE KUNNEN DE DEELASPECTEN VERBETERD WORDEN?

Zowel voor controle, monitoring als sturing moet onderzocht worden hoe deze deelaspecten van beheerbaarheid verbeterd kunnen worden.

1.2.3 DEELVRAAG 3: IN WELKE MATE ZIJN DE VERSCHILLENDE VERBETERINGEN IMPLEMENTEERBAAR?

Aan de hand van een gevalstudie zullen de mogelijke verbeteringen die volgen uit de literatuurstudie op concrete procesmodellen worden toegepast. Zo kan voor alle verbeteringen worden vastgesteld in welke mate ze ook effectief toepasbaar zijn.

1.3 ONDERZOEKSMETHODE

Om tot een antwoord op de centrale onderzoeksvraag en deelvragen te komen, zullen een literatuurstudie en een gevalstudie worden uitgevoerd.

1.3.1 LITERATUURSTUDIE

Voor zowel de eerste deelvraag (“Wat zijn de deelaspecten van beheerbaarheid?”), als de tweede (“Hoe kunnen de deelaspecten verbeterd worden?”) zal een antwoord gezocht worden aan de hand van een literatuurstudie. Hiervoor zullen wetenschappelijke artikels gezocht worden in wetenschappelijke databases, via Google Scholar en in de bibliotheek van de Universiteit Hasselt.

1.3.2 CASE STUDY

Het resultaat van de literatuurstudie zal aan de hand van een gevalstudie aan de praktijk getoetst worden. Concreet zullen voor elk afzonderlijk deelaspect van beheerbaarheid de mogelijke verbeteringen op een voorbeeld worden toegepast zodat er conclusies kunnen getrokken worden over de implementeerbaarheid van de verschillende verbeteringen.

Zoals blijkt uit de probleemstelling, vormen bedrijfsprocessen het hart van iedere organisatie. In een omgeving van steeds toenemende concurrentie, kunnen bedrijven hun doelstellingen slechts realiseren indien input effectief (i.e. door de juiste acties) en efficiënt (i.e. door de acties juist te doen) wordt omgezet in output. Deze transformatie van input naar output wordt verwezenlijkt door bedrijfsprocessen.

Gezien het grote aandeel van bedrijfsprocessen in het wel of niet realiseren van bedrijfsdoelstellingen, is het belangrijk dat deze processen de nodige aandacht krijgen en goed beheerd worden, wil men bedrijfsdoelstellingen ook effectief realiseren. “Beheren” is echter een vrij vage term. Het is de vertaling van het Engelse woord “to manage”, wat zoveel betekent als “de controle over iets bewaren”.

De betekenis van “to manage” in een proces-gerelateerde context kan worden afgeleid uit de definitie van Business Process Management (BPM): “Supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes, involving humans, organisations, applications, documents, and other sources of information” (Weske et al., 2004, p. 2). Het beheren van bedrijfsprocessen gaat dus over het gebruik van methodes en technieken om processen te ondersteunen, te controleren en te analyseren.

In deze masterproef onderscheiden we drie deelaspecten van beheerbaarheid: controle, monitoring en sturing. Het beheren van bedrijfsprocessen wordt dus gezien als het bewaren van de controle over het proces, het uitvoeren van metingen om na te gaan of het proces nog steeds onder controle is en het eventueel bijsturen van het proces indien bepaalde afwijkingen zich voordoen.

Voor elk van bovengenoemde deelaspecten zal aan de hand van een literatuurstudie worden nagegaan wat het deelaspect precies inhoudt en hoe het verbeterd kan worden. Vervolgens kunnen de verschillende deelaspecten worden samengebracht om de beheerbaarheid van een proces in zijn geheel te verbeteren.

Het eerste deelaspect van beheerbaarheid is controle. Vooraleer we dieper ingaan op dit aspect is het belangrijk om duidelijk te definiëren wat we juist verstaan onder deze term. We zouden “onder controle hebben” kunnen definiëren als “in bedwang hebben, beheersen”. Wanneer we deze definitie vertalen naar bedrijfsprocessen, willen we dus nagaan in welke mate het proces beheersbaar is. Intuïtief kunnen we stellen dat we willen onderzoeken in welke mate het middelmanagement de teugels van het proces in handen heeft.

Het onder controle houden van processen is belangrijk, want processen variëren. Elke eigenschap van een proces is een functie van een aantal variabelen:

$$Y = f(X_1, X_2, X_3, \dots, X_n)$$

met Y een eigenschap van een proces en n het aantal variabelen. $X_1, X_2, X_3, \dots, X_n$ kunnen zowel discreet als continu zijn, kunnen Y zowel direct als indirect beïnvloeden en verschillen onderling wat betreft de grootte van het effect dat ze uitoefenen op Y . Het aantal variabelen dat een invloed heeft op Y kan oneindig groot zijn, maar in de praktijk geldt veelal het Pareto-principe: een klein aantal variabelen is verantwoordelijk voor het grootste deel van de verandering in Y (Petrovich, 2000).

Procescontrole handelt over het beperken van de verandering in Y . Petrovich (2000) definieert procescontrole als “the ability to constrain variation and prevent nonconformance over time” (p. 3). De controle over processen kan dus verbeterd worden door veranderingen in processen aan te brengen die een invloed hebben op de oorzaak van de verandering met betrekking tot een bepaalde eigenschap van een proces, i.e. de variantie in Y kan worden beperkt door maatregelen die de veranderingen in $X_1, X_2, X_3, \dots, X_n$ onder controle houden.

Voor het opzet van deze masterproef wordt procescontrole gedefinieerd als: “voorspelbaarheid van een proces met betrekking tot tijd, kosten en kwaliteit”.

3.1 EENVOUDIGE PROCESSEN

Een eerste vereiste om een proces onder controle te houden, is dat het proces begrijpbaar is. Om dit na te gaan, zijn er zogenaamde complexiteitsmaatstaven ontwikkeld. Deze maatstaven vinden vaak hun oorsprong in de computerwetenschappen, waar ze dienst doen als middel om de complexiteit van software te bepalen. Op basis van de complexiteit kunnen dan voorspellingen gedaan worden over bijvoorbeeld het aantal fouten of onderhoudskosten.

Laue en Gruhn (2006) hebben onderzoek gedaan naar hoe complexiteitsmaatstaven voor software gebruikt kunnen worden om de complexiteit van procesmodellen te bepalen. Alvorens de complexiteit van een model na te gaan, is het echter belangrijk het begrip “complexiteit” duidelijk te definiëren. Laue en Gruhn (2006) definiëren “complexity metrics” als “measurements that can tell us whether a model is easy or difficult to understand” (p. 1).

Een eerste maatstaf om de complexiteit van software te bepalen die Laue en Gruhn (2006) bespreken is de “lines of code”, die de complexiteit bepaalt op basis van de grootte van een programma. Wanneer we het aantal activiteiten in een procesmodel als equivalent voor het aantal statements in een programma beschouwen, kan deze maatstaf gebruikt worden om de complexiteit van een procesmodel vast te stellen op basis van het aantal activiteiten in dat model.

We kunnen dus stellen dat hoe meer activiteiten een procesmodel bevat, hoe complexer het model is. Hoe groter de complexiteit van een model, des te moeilijker het te begrijpen is, hetgeen cruciaal is om de controle over het proces te bewaren. Wanneer we het aantal activiteiten in een procesmodel zouden verminderen, zou de complexiteit afnemen en de begrijpbaarheid toenemen, wat op zijn beurt dan weer een positief effect heeft op de controle over het proces. In sectie 6.2 wordt een methode beschreven om op basis van gelijkens tussen verschillende taken overbodige taken te elimineren, zodat de complexiteit van het proces afneemt en de controle erover toeneemt.

Intuïtief lijkt de “lines of code” een geschikte maatstaf om de complexiteit van een procesmodel vast te stellen. Dikwijls zal een procesmodel dat bestaat uit duizend activiteiten inderdaad complexer zijn dan een model met slechts vijftig activiteiten. Toch is dit niet altijd het geval. De maatstaf houdt namelijk geen rekening met de structuur van een proces. Een proces met duizend activiteiten en een eenvoudige structuur kan minder complex zijn dan een proces met vijftig activiteiten maar met een zeer complexe structuur.

Willen we de complexiteit van een procesmodel correct bepalen, moeten we dus ook rekening houden met de structuur van het model. Een maatstaf die de structuur van een procesmodel in rekening brengt is de control flow complexity of processes (CFC). Deze maatstaf telt het aantal verschillende paden die genomen kunnen worden in een proces. Aangezien keuze in een proces plaatsvindt bij een split, kan het aantal paden van een proces verkregen worden door aan elk soort split een bepaalde waarde toe te kennen en vervolgens de toegekende waardes op te tellen.

Aangezien bij een AND-split steeds alle uitgaande paden gevolgd moeten worden, levert een AND-split maar één pad op. Daarom verhoogt de CFC-maatstaf met 1 voor elke AND-split. Bij een XOR-split met n uitgaande paden, moet steeds één en slechts één pad gekozen worden. Daarom kunnen er steeds n verschillende paden gevolgd worden, en wordt voor elke XOR-split met n uitgaande paden de CFC-maatstaf met n verhoogd. Voor een OR-split met n uitgaande paden tot slot, wordt steeds minstens 1 en hoogstens n paden gekozen. Dit levert $2^n - 1$ mogelijke paden op en dus wordt de CFC-maatstaf met $2^n - 1$ verhoogt voor elke OR-split met n uitgaande paden.

Wanneer we ons op de CFC-maatstaf baseren om de complexiteit van een procesmodel te bepalen, kunnen we stellen dat de complexiteit verlaagd kan worden, de begrijpbaarheid verhoogd en op zijn beurt ook de controle verhoogd kan worden door het aantal splits in het model te beperken. Hierbij moeten we echter twee bemerkingen maken: (1) het beperken van het aantal splits gaat ten koste van de flexibiliteit van het proces, waarover later meer en (2) de CFC-maatstaf houdt enkel rekening met het aantal splits, niet met de structuur van de splits, hetgeen ook voor complexiteit kan zorgen.

Om ook de mate waarin de verschillende splits in elkaar verweven zijn in rekening te brengen, maken Laue en Gruhn (2006) gebruik van maatstaven voor het bepalen van softwarecomplexiteit die een indicatie geven over hoe software in elkaar verweven is: de “maximum nesting depth” en de “mean nesting depth”. Wanneer we deze maatstaven toepassen op procesmodellen, is de “nesting depth” van een bepaalde activiteit gelijk aan het aantal beslissingen die moeten genomen worden om die bepaalde activiteit te kunnen uitvoeren.

Om een meer accuraat beeld te krijgen van de complexiteit van een procesmodel, is het dus nuttig om zowel de CFC-maatstaf, die focust op het aantal splits in een model, als de nesting depth, die rekening houdt met de structuur van de splits, te berekenen.

Een laatste manier om de begrijpbaarheid van procesmodellen te verhogen, is het opsplitsen van het proces in modules. Henry en Kafura (1981) evalueren een module op basis van volgende maatstaf: $((fan - in) \cdot (fan - out))^2$. Hierbij is “fan-in” gelijk aan het aantal andere modules die de module oproepen en is “fan-out” gelijk aan het aantal andere modules die de module oproept. De waarde voor deze maatstaf zal hoog zijn wanneer zowel de waarde van fan-in als die van fan-out hoog zijn. Dit is het geval wanneer een module door veel andere modules opgeroepen wordt, maar zelf ook veel andere modules oproept. Aangezien dit patroon de module moeilijk bruikbaar maakt, moet bij een hoge waarde voor deze maatstaf nagegaan worden of de module herontworpen kan worden om de begrijpbaarheid van het model te verhogen.

3.2 ELIMINEREN VAN TAKEN

De controle over een proces kan verbeterd worden door onnodige taken te elimineren. Een taak kan als onnodig beschouwd worden wanneer ze vanuit een klantenperspectief geen waarde toevoegt. Zo zouden bijvoorbeeld controlerende taken, hoewel vaak zeer nuttig, vanuit een klantenperspectief als onnodig beschouwd kunnen worden (Reijers & Liman-Mansar, 2005).

Daarnaast kunnen taken ook overbodig zijn: wanneer twee taken binnen eenzelfde proces zeer gelijkend zijn, kan er één geëlimineerd worden. Om na te gaan hoe gelijkend twee taken zijn, hebben Castano, De Antonellis en Melchiori (1999) “similarity coefficients” ontwikkeld. Om deze coëfficiënten te berekenen, wordt gebruik gemaakt van zogenaamde “discriptors”. Hoewel weinig relevant vanuit een praktisch standpunt, wordt hieronder deze methode beschreven.

Een “activiteitdiscriptor” is een formele representatie van de karakteristieken van een activiteit. Een descriptor D_i van een proces A_i is een 4-tuple van karakteristieken: $D_i = \langle f_1, f_2, f_3, f_4 \rangle$ (Castano & De Antonellis, 1998).

Met

- f_1 de naam n_{OU_k} van de organisatorische eenheid OU_k waartoe A_i behoort
- f_2 de verzameling $IN(A_i)$ van de namen van de inputentiteiten van A_i
- f_3 de verzameling $OUT(A_i)$ van de namen van de outputentiteiten van A_i
- f_4 de verzameling $OP(A_i)$ van de operaties uitgevoerd door A_i . Operaties worden beschreven aan de hand van triplets, i.e. $OP(A_i) = \{O_{1i} = \langle n_{OP_{1i}}, CST_{1i}, CMS_{1i} \rangle, \dots, O_{qi} = \langle n_{OP_{qi}}, CST_{qi}, CMS_{qi} \rangle\}$

Met

1. $n_{OP_{ki}}$ de naam van een operatie
2. $CST_{ki} \subseteq IN(A_i)$ een verzameling namen van de verplichte entiteiten voor de uitvoering van $n_{OP_{ki}}$ (CST staat voor “constitutive”).
3. CMS_{ki} een verzameling namen van optionele activiteiten bij de uitvoering van $n_{OP_{ki}}$ (CMS staat voor “circumstantial”).

Castano & De Antonellis (1998) stellen dat door de verbanden te onderzoeken tussen de namen die in de activiteitendiscriptors staan, de gelijkenis tussen twee activiteiten kan worden nagegaan. Bij deze methode stelt zich echter het volgende probleem: twee activiteitendiscriptors kunnen dezelfde of semantisch gelijkende entiteiten en operaties hebben, die daarom niet noodzakelijk ook dezelfde naam hebben. Furnas, Landauer, Gomez en Dumais (1987) stellen dat de kans dat twee ontwikkelaars voor een gegeven entiteit dezelfde naam kiezen, tussen 7% en 18% ligt. Om dit probleem het hoofd te bieden, kan gebruik gemaakt worden van een semantisch woordenboek, waarin semantisch gelijkende entiteiten en operaties bij hetzelfde concept worden ondergebracht. Elk concept heeft dus een bijhorende cluster met namen van entiteiten en operaties die semantisch gelijk zijn aan dat concept.

Aan de hand van het semantisch woordenboek wordt de “affiniteit” tussen operatie- of entiteitsnamen van twee verschillende activiteitendiscriptors onderzocht. Twee namen hebben affiniteit wanneer ze duiden op dezelfde entiteit of operatie, of wanneer ze duiden op een semantisch gelijkende entiteit of operatie (Castano & De Antonellis, 1998).

In het semantisch woordenboek worden concepten met elkaar verbonden door hiërarchische links. Om de affiniteit te bepalen, wordt aan de hiërarchische links in het woordenboek een sterkte σ toegewezen. Twee namen hebben affiniteit als ze verwijzen naar een gemeenschappelijk concept in

het woordenboek, i.e. als er tussen deze namen een pad van lengte l bestaat met $l \geq 1$, genoteerd als $\overset{l}{\rightarrow}$. De affiniteit tussen de twee namen komt dan overeen met de sterkte van het pad dat hen verbindt. De naamaffiniteit-coëfficiënt van twee namen n_i en n_j , genoteerd als $NA(n_i, n_j)$, geeft de sterkte van het pad dat hen verbindt en wordt als volgt berekend (Castano & De Antonellis, 1998):

$$NA(n_i, n_j) = \begin{cases} 1 & \text{if } (n_i = n_j) \\ \tau^N(l, \sigma) & \text{als } (n_i \neq n_j) \wedge (n_i \overset{l}{\rightarrow} n_j), l \geq 1 \\ 0 & \text{elders} \end{cases}$$

De “entity-based similarity coefficient” van twee activiteiten met descriptors A_i en A_j , genoteerd als $ESim(A_i, A_j)$ wordt berekend door de input/output-entiteiten van de verschillende descriptors met elkaar te vergelijken (Castano et al., 1999):

$$ESim(A_i, A_j) = \frac{2 \cdot AF_{tot}(IN(A_i), IN(A_j))}{|IN(A_i)| + |IN(A_j)|} + \frac{2 \cdot AF_{tot}(OUT(A_i), OUT(A_j))}{|OUT(A_i)| + |OUT(A_j)|}$$

Waar $AF_{tot}(IN(A_i), IN(A_j))$ (respectievelijk $AF_{tot}(OUT(A_i), OUT(A_j))$) de totale affiniteit is tussen de verschillende paren input (respectievelijk output) entiteiten in A_i en A_j en $||$ de kardinaliteit van een verzameling weergeeft. AF_{tot} is de som van de affiniteit van alle input/output entiteitsparen die affiniteit hebben in het semantisch woordenboek.

De entity-based similarity coefficient kan waarden aannemen tussen 0 en 2. De waarde is 0 wanneer er zowel in de inputverzameling als outputverzameling geen paren met affiniteit zijn. De waarde is 2 wanneer elke input- en outputentiteit van A_i affiniteit heeft met een input- en outputentiteit van A_j en vice versa (Castano et al., 1999).

De “functionality-based similarity coefficient” van twee activiteiten met descriptors A_i en A_j , genoteerd als $FSim(A_i, A_j)$ wordt berekend door de operaties van de verschillende descriptors met elkaar te vergelijken (Castano et al., 1999):

$$FSim(A_i, A_j) = \frac{2 \cdot AF_{tot}(OP(A_i), OP(A_j))}{|OP(A_i)| + |OP(A_j)|}$$

Waar $AF_{tot}(OP(A_i), OP(A_j))$ de totale affiniteit is tussen de verschillende paren operaties die gelijkend zijn in A_i en A_j . Twee operaties zijn gelijkend als hun acties en hun constitutieve informatie-entiteiten affiniteit hebben in het semantisch woordenboek. De affiniteit tussen twee operaties wordt verkregen door de affiniteit van hun overeenkomstige elementen op te tellen (Castano et al., 1999).

De “functionality-based similarity coefficient” kan waarden aannemen tussen 0 en 3. De waarde is 0 wanneer er geen gelijkende operaties bestaan tussen A_i en A_j . De waarde is 3 wanneer elke operatie van A_i gelijkend is aan een operatie van A_j en vice versa (Castano et al., 1999).

3.3 SAMENVOEGEN EN OPSPLITSEN VAN TAKEN

Wanneer de grootte van een bepaalde taak niet optimaal is, kan dit een nadelig effect hebben op de controle over een proces. Wanneer twee taken samengevoegd worden tot één grotere taak, vermindert namelijk het aantal setups in het proces. De variantie in tijd zal dus afnemen. Reijers (2003) stelt daarnaast dat wanneer twee of meer taken gecombineerd worden, de tijd die een werknemer nodig heeft om vertrouwd te raken met de gecombineerde taak korter is dan de som van de tijd om vertrouwd te raken met de afzonderlijke taken. Bovendien komt het samenvoegen van twee of meerdere taken ook de kwaliteit ten goede.

Taken kunnen echter niet eindeloos worden samengevoegd. Volgens Reijers (2003) zullen er wanneer te veel taken worden samengevoegd twee nadelige effecten optreden: (1) de geleverde kwaliteit zal afnemen en (2) het proces zal inboeten aan flexibiliteit. Het afnemen van de kwaliteit is een gevolg van het feit dat wanneer een taak te groot wordt, ze moeilijker uit te voeren wordt en in extreme gevallen zelfs niet meer uit te voeren is. Het verlies aan flexibiliteit wordt verklaard doordat wanneer verschillende taken eenmaal samengevoegd zijn, de samengevoegde taak atomisch is. De taak zal steeds in zijn geheel dienen te worden uitgevoerd, waar het voordien mogelijk was om slechts een deel van die samengestelde taak uit te voeren.

Beide nadelige effecten kunnen verholpen worden door taken terug op te splitsen. Het is dus van belang een goede balans te vinden tussen het samenvoegen van kleine taken enerzijds, met als gevolg een verminderde variantie in tijd en een hogere kwaliteit, en het opsplitsen van te grote taken anderzijds, zodat taken niet te groot worden zodat de kwaliteit achteruit gaat en er te veel aan flexibiliteit moet worden ingeboet.

3.4 VERPLAATSEN VAN TAKEN

Wanneer we te maken hebben met een zogenaamd “knock-out process”, kan de variantie in tijd en kost beperkt worden door taken in het proces onderling van plaats te wisselen. Een knock-out process is een proces waarbij voor elke case een vooraf bepaald aantal taken dient te worden uitgevoerd. De uitvoering van een bepaalde taak stopt onmiddellijk indien er een reden wordt gevonden om de case te verwerpen. Enkel cases waarvoor alle taken succesvol worden afgerond, worden aanvaard (Van der Aalst, 2001).

Om de variantie in tijd en kosten te beperken, moet er voor gezorgd worden dat taken die het vroegtijdig beëindigen van een case veroorzaken, steeds zo vroeg mogelijk in het proces plaatsvinden. Wanneer we de taken van een proces in willekeurige volgorde zouden plaatsen, zou het mogelijk zijn dat een bepaalde taak met een grote kans om de case te verwerpen achteraan in het proces staat. Bij een proces waar elke case het hele proces van vooraan opnieuw moet doorlopen wanneer ze geweigerd wordt, en dit net zolang tot ze geaccepteerd wordt, zal er zo een grote variantie in tijd en kost ontstaan tussen cases die het proces zonder problemen doorlopen en cases die opnieuw gestart moeten worden. Het kan namelijk voorkomen dat problemen pas laat gedetecteerd worden, waardoor veel tijd en geld verloren gaat en zo het verschil in tijd en kost tussen de verschillende cases toeneemt.

Wanneer we taken echter rangschikken naar gelang de waarschijnlijkheid van vroegtijdige beëindiging en naargelang de bijhorende kost, waarbij taken met een hoge kans op vroegtijdige beëindiging en een lage kost vooraan komen en taken met een lage kans op vroegtijdige beëindiging en een hoge kost zo lang mogelijk worden uitgesteld, zal de variantie in tijd en kost afnemen. Cases die niet voldoen zullen vroeger verworpen worden dan wanneer de taken willekeurig gerangschikt zouden zijn, waardoor minder tijd en geld verloren gaat en het verschil in tijd en kost tussen de verschillende cases afneemt.

Een tweede manier om de variantie in tijd en kosten te beperken is het ordenen van taken op basis van locatie. Wanneer verschillende taken op een verschillende locatie worden uitgevoerd, kunnen ze zo gerangschikt worden dat de verplaatsingstijd tussen de taken geminimaliseerd wordt. Wanneer de locatie van activiteit B ver verwijderd is van de locatie van A en C, maar A en C wel dicht bij elkaar liggen, kan bijvoorbeeld de taakvolgorde gewijzigd worden van A, B, C naar A, C, B. Op deze manier worden twee lange verplaatsingen gereduceerd tot één korte en één lange verplaatsing, wat de variantie in tijd en kosten ten goede komt.

Een derde mogelijkheid bestaat erin de taken te ordenen op basis van taakinhoud. Door gelijkende taken na elkaar te zetten, kan het aantal setups beperkt worden, wat er voor zorgt dat de variantie in tijd en kosten van het proces afneemt.

3.5 GELIJKTIJDIGE UITVOERING VAN TAKEN

Soms worden in processen taken in sequentie gezet, zonder dat er beperkingen zijn die zulke ordening verantwoordt (Reijers, 2003). Wanneer taken sequentieel geordend worden, ontstaan er onvermijdelijk wachttijden. Een taak kan dan namelijk pas worden uitgevoerd nadat alle taken voorafgaand aan die taak voltooid zijn. Soms is het echter mogelijk om verschillende taken gelijktijdig oftewel “parallel” uit te voeren.

Het voordeel van taken in parallel te plaatsen in plaats van in sequentie, is dat wachttijden gereduceerd kunnen worden en dus ook de variantie in tijd zal beperkt worden. Een bepaalde taak kan namelijk al gelijktijdig met een andere taak worden uitgevoerd in plaats van erna. Van der Aalst (2001) somt vier voorwaarden op waaraan voldaan moet worden om taken succesvol in parallel te plaatsen: (1) de taken die gelijktijdig worden uitgevoerd zijn van dezelfde orde van grootte, (2) de kans dat een case verworpen wordt op basis van één van de taken in parallel is eerder klein, (3) er worden geen bronnen overbelast als gevolg van het gelijktijdig uitvoeren van de taken en (4) de tijd die nodig is voor synchronisatie is beperkt.

Wanneer twee of meer taken die in parallel staan van een verschillende orde grootte zijn, vermindert het effect van de gereduceerde wachttijd. Een taak die volgt op de taken in parallel kan namelijk pas starten wanneer de taken die gelijktijdig worden uitgevoerd beide voltooid zijn. Wanneer de ene taak bijvoorbeeld vier weken duurt en de andere twee dagen, zal het in parallel vier weken duren om beide taken uit te voeren en in sequentie vier weken en twee dagen. Het verschil is dan dus minimaal.

Indien van de taken die in parallel staan een bepaalde taak een grote kans heeft om ervoor te zorgen dat een case verworpen wordt, is het nadelig deze taken in parallel te plaatsen. Alle taken die gelijktijdig gestart zijn, maar die pas later zouden worden uitgevoerd indien de taken in sequentie stonden, leiden tot extra kosten omdat ze onnodig worden uitgevoerd.

Bij het plaatsen van taken in parallel, dient rekening gehouden te worden met de bronnen die gebruikt worden om de taken uit te voeren. Indien verschillende taken gebruik maken van dezelfde bron, kan overbelasting ontstaan, wat ten koste kan gaan van de geleverde kwaliteit.

Reijers en Liman-Mansar (2005) stellen dat er ook een nadeel verbonden is aan het uitvoeren van taken in parallel: het proces kan hierdoor complexer worden, wat op zijn beurt kan leiden tot meer fouten en minder kwaliteit.

Om ervoor te zorgen dat het gelijktijdig uitvoeren van taken effectief een gunstig effect heeft op de variantie in tijd en kosten van het proces, is het dus belangrijk dat voldaan wordt aan bovengenoemde voorwaarden, alsook dat de voordelen zorgvuldig tegen de nadelen worden afgewogen.

3.6 TOEVOEGEN VAN CONTROLETAKEN

Hoewel variantie in productkwaliteit onvermijdelijk is, moet kwaliteitsverschil toch zoveel mogelijk beperkt worden. Een manier om de geleverde kwaliteit onder controle te houden is het aan het proces toevoegen van controletaken. Aangezien het toevoegen van extra taken om controle te bevorderen in strijd is met het eerder vermelde elimineren van taken om een proces eenvoudiger te maken, moeten voor het toevoegen van een controletaak de bijhorende kosten en baten tegen elkaar worden afgewogen. Men dient zich dus af te vragen of de winst aan controle door het toevoegen van de taak groter is dan het verlies aan controle door de toegenomen complexiteit van het proces?

Volgens Reijers (2003) is het voordeel van controlerende taken tweeledig: enerzijds verhoogt het de kwaliteit van de output van het proces, anderzijds moeten minder taken opnieuw worden uitgevoerd omdat fouten en afwijkingen eerder worden opgespoord. Een voorbeeld hiervan is het toevoegen van een controletaak vlak na het aankopen van materialen. Indien niet gecontroleerd zou worden of alle materialen in de juiste aantallen aanwezig zijn, zou het proces starten en na verloop van tijd zou vastgesteld worden dat het proces niet voltooid kan worden als gevolg van het ontbreken van bepaalde benodigde materialen.

Naast tussentijdse controles, kunnen controletaken ook op het eind van het proces geplaatst worden, om zo een kwalitatief hoogstaand eindproduct te garanderen. Op die manier zal de klanttevredenheid toenemen en zullen klanten minder ondersteuning nodig hebben van de dienst na verkoop. Hierdoor zal de variantie in kosten ook afnemen.

Zoals reeds gezegd verhoogt het toevoegen van controletaken de complexiteit van het proces. Reijers (2003) vermeldt ook nog twee andere nadelen van het toevoegen van controlerende taken: het uitvoeren van controles kost tijd en middelen.

De impact van extra controletaken op de controleerbaarheid van het proces is dus niet eenvoudig te bepalen. Alles hangt af van de factoren met een positieve impact enerzijds, i.e. minder variantie in zowel tijd als kwaliteit en minder variantie in kosten als gevolg van een verhoogde klanttevredenheid, en de factoren met een negatieve impact anderzijds, i.e. een verhoogde procescomplexiteit en extra kosten.

Om een positieve impact op de controleerbaarheid van een proces te verkrijgen, is een grondige afweging tussen bovengenoemde factoren nodig, aangezien het verwijderen van controletaken met als doel de complexiteit van het proces te verlagen juist een omgekeerd effect kan hebben indien de taken die verwijderd worden cruciaal blijken om de variantie in tijd en kwaliteit te verminderen. Hetzelfde geldt voor het overmatig toevoegen van controlerende taken: de verhoogde complexiteit als gevolg van de extra taken en de extra kosten voor het uitvoeren van de controles kunnen het proces onvoorspelbaar maken in plaats van meer controleerbaar.

3.7 BRONOPTIMALISATIE

Om de controle over een proces te bewaren, is het belangrijk er voor te zorgen dat er een eenvoudige en duidelijke verantwoordelijkheidsstructuur is. Een proces kan namelijk pas gecontroleerd verlopen wanneer het duidelijk is wat van alle betrokkenen verwacht wordt. Om weer te geven wat voor elke activiteit van wie verwacht wordt, worden zogenaamde “RACI-matrices” gebruikt. Deze matrices geven voor elke activiteit weer wie de “responsible”-rol, de “accountable”-rol, de “consulted”-rol en de “informed”-rol vervult.

De responsible is de persoon die het werk uitvoert en de verantwoordelijkheid over de activiteit draagt totdat die beëindigd is. De accountable keurt het werk van de responsible goed en draagt nadien de verantwoordelijkheid over de activiteit. De consulted zijn de personen die naar hun mening gevraagd worden tijdens het uitvoeren van het werk. De informed-rol tot slot, wordt vervuld door de personen die up-to-date worden gehouden over de voortgang van de activiteit (Cabanillas, Resinas, & Ruiz-Cortés, 2011).

De RACI-matrix wordt opgebouwd aan de hand van deze vier rollen. De verschillende rijen in de matrix komen overeen met de activiteiten van een bepaald proces, de kolommen komen overeen met de werknemers die bij het proces betrokken zijn. Elke cel bevat één of meer van de vier rollen, zodat in een cel de verantwoordelijkheden van een bepaalde werknemer met betrekking tot een bepaalde activiteit kunnen worden afgelezen.

Wanneer voor een bepaald proces de RACI-matrix gegeven is, kan de verantwoordelijkheidsstructuur geanalyseerd worden. Wanneer dit niet het geval is, moet de RACI-matrix afgeleid worden uit het procesmodel alvorens het mogelijk is de verantwoordelijkheidsstructuur te analyseren.

Het afleiden van de matrix kan door de rijen van de matrix overeen te laten komen met de activiteiten in het proces, terwijl de kolommen bestaan uit de werknemers die bij het proces betrokken zijn. Om in de cellen van de matrix vervolgens de juiste rol te plaatsen, moet uit het proces kunnen afgeleid worden welke rol een bepaalde werknemer vervult voor een bepaalde activiteit. Dit kan aan de hand van zogenaamde “RACI-patronen”, indien aanwezig in het procesmodel (Cabanillas et al., 2011).

Wanneer we er bij een procesmodel in BPMN vanuit gaan dat de namen van de “lanes” overeenkomen met de verantwoordelijke voor alle activiteiten binnen die lane, dan kan de responsible voor een bepaalde activiteit worden afgeleid uit de naam van de lane waartoe die activiteit behoort. Hierdoor veronderstellen we impliciet dat steeds slechts één persoon de verantwoordelijkheid over een bepaalde activiteit draagt. De accountable voor een bepaalde activiteit is de werknemer in wiens lane we een sub proces terugvinden dat de goedkeuring van die bepaalde activiteit afhandelt. Een “request-response”-patroon tussen verschillende werknemers geeft aan wie de consulted is. De informed-rol tot slot herkennen we aan een patroon dat gelijkend is aan dat van de consulted-rol, maar hier vindt geen response plaats (Cabanillas et al., 2011).

Wanneer we van een bepaald proces de bijhorende RACI-matrix verkregen hebben, kan de verantwoordelijkheidsstructuur van het desbetreffende proces geanalyseerd worden. Een verticale analyse van de matrix geeft meer informatie over de verantwoordelijkheden van de verschillende betrokkenen, terwijl een horizontale analyse duidelijk maakt welke verantwoordelijkheden er zijn met betrekking tot een bepaalde activiteit.

Wanneer we verticaal in één bepaalde kolom veel de responsible-rol terugvinden, impliceert dit dat één bepaalde werknemer veel activiteiten moet uitvoeren en kan de vraag gesteld worden of deze werknemer niet te zwaar belast wordt om het proces gecontroleerd te laten verlopen. Er zou onderzocht kunnen worden of de verantwoordelijkheid over bepaalde activiteiten kan worden doorgeschoven naar werknemers met minder verantwoordelijkheden.

Een werknemer die bij verschillende activiteiten de accountable-rol vervult, behoort waarschijnlijk tot het management. Om te vermijden dat managers overbelast raken, kan nagegaan worden of het mogelijk is deze rol voor bepaalde activiteiten door te schuiven naar een lager niveau in de organisatie.

Wanneer we een kolom in zijn geheel bekijken, kunnen we nagaan of het patroon in die kolom overeenstemt met de persoonlijkheid en capaciteiten van de desbetreffende werknemer. Voor een gecontroleerd verloop van het proces is het immers belangrijk dat voor iedere werknemer de belasting niet te hoog is, rekening houdend met zijn of haar capaciteiten.

Wanneer horizontaal voor een bepaalde activiteit veel de responsible-rol terugkomt, bestaat het gevaar dat het proces moeilijk onder controle te houden is. Een mogelijke oplossing bestaat erin de activiteit op te splitsen in verschillende activiteiten en aan iedere activiteit minder responsables toe te wijzen. Wanneer echter niemand de responsible-rol voor een bepaalde activiteit toegewezen krijgt, kan het zijn dat de activiteit niet afgerond wordt, omdat alle betrokkenen zitten te wachten, het zij om de activiteit goed te keuren, het zij om advies te geven of informatie te ontvangen. Ook wanneer voor een bepaalde activiteit de accountable-rol niet toegewezen wordt, stelt er zich een probleem: de activiteit kan worden overgeslagen, aangezien het niet uitvoeren van de taak voor geen enkele betrokkene gevolgen heeft. Daarom is het belangrijk dat voor elke activiteit er één en slechts één werknemer de accountable-rol toegewezen krijgt.

Activiteiten waarvoor verschillende werknemers de consulted-rol vervullen, kunnen het proces aanzienlijk vertragen. In zo'n geval moeten de baten van het inwinnen van informatie bij meerdere werknemers worden afgewogen tegen de kosten die de vertraging van het proces met zich meebrengt. Ook wanneer er informatie wordt ingewonnen bij verschillende werknemers dient deze afweging gemaakt te worden. Wanneer de consulted-rol en de informed-rol echter zeer weinig voorkomen, kan dit erop wijzen dat er te weinig communicatie is tussen de betrokkenen, wat niet bevorderend is voor het onder controle houden van het proces.

3.8 INFORMATIEOPTIMALISATIE

Om een proces tot een goed einde te brengen, is het noodzakelijk dat alle betrokken partijen over de informatie beschikken die noodzakelijk is om hun taken te kunnen uitvoeren. De bron van dergelijke informatie kan zowel intern als extern zijn. Om te verzekeren dat het geproduceerde product of de geleverde dienst zo goed mogelijk aan de eisen van de klant voldoet of om de klant productondersteuning te bieden na zijn of haar aankoop, kan het noodzakelijk zijn om ook de klant in het proces te betrekken en er informatie mee uit te wisselen. Ook intern, tussen verschillende afdelingen, zal informatie uitgewisseld moeten worden. Zo zal bijvoorbeeld de accountant enkel een order correct kunnen registreren indien hij ook beschikt over de gegevens van de betrokken klant.

Bij het ontwerpen van een proces dient dus voldoende aandacht te worden besteed aan deze informatiestromen binnen het proces, zowel voor wat de interne informatiestromen tussen afdelingen als wat de externe informatiestromen met derden zoals klanten of partners betreft.

Hoewel communiceren met klanten vaak onvermijdelijk is, moet deze communicatie tot een minimum beperkt worden. Het uitwisselen van informatie kan namelijk zeer tijdrovend zijn. Reijers en Liman-Mansar (2005) stellen dat vooral wanneer informatie via e-mail verkregen dient te worden, aanzienlijke wachttijden kunnen ontstaan, wat de voorspelbaarheid van het proces niet ten goede komt.

Bij elk contact met een derde partij, hetzij een klant, hetzij een partner, bestaat bovendien de mogelijkheid dat fouten optreden. Hammer en Champy (1993) beschrijven een proces waarbij er door verschillende contacten met klanten voor facturatie doeleinden grote vertragingen ontstaan. Het reduceren van het aantal contacten kan dus de controle over het proces verhogen.

Reijers en Liman-Mansar (2005) stellen dat om het aantal uitwisselingen van informatie te beperken, het niet steeds noodzakelijk is bepaalde informatiestromen te schrappen. Dikwijls is het mogelijk om met een beperkte bijkomende kost verschillende informatiestromen samen te voegen. Dit kan er dan wel weer voor zorgen dat er grote hoeveelheden data gelijktijdig moeten worden verwerkt, wat ook weer extra kosten met zich meebrengt.

Het analyseren en optimaliseren van de uitwisseling van interne informatie tussen verschillende processen wordt in BPMN vergemakkelijkt door het gebruik van zogenaamde “pools”. Doordat elk proces zich in een aparte pool bevindt, moet ervoor gezorgd worden dat er voldoende communicatie tussen de pools is opdat het proces in elke pool optimaal kan verlopen. Zo kan bijvoorbeeld het registratieproces van een betaling slechts correct uitgevoerd worden indien er voldoende communicatie is met het productieproces.

3.9 MINIMALISEREN VAN HET AANTAL BETROKKEN AFDELINGEN

Wanneer bij een proces verschillende afdelingen betrokken zijn, kan dit het onder controle houden van het proces bemoeilijken. Reijers en Liman-Mansar (2005) vermelden vier voordelen als gevolg van het beperken van het aantal betrokken afdelingen bij een proces: (1) minder coördinatieproblemen, (2) minder conflicten, (3) een snellere respons ten opzichte van de klant en (4) minder wachtrijen.

Verschillen tussen afdelingen kunnen de coördinatie van het proces bemoeilijken, aangezien geen twee afdelingen hetzelfde zijn. Zo zal de manier van werken op de ene afdeling verschillen van die op de andere afdeling. Ook kan de communicatie tussen twee afdelingen moeizamer verlopen dan binnen één afdeling als gevolg van de fysieke afstand tussen de afdelingen.

Samenwerking tussen verschillende afdelingen kan ook een bron van conflict zijn. Verschillende afdelingen zullen er verschillende prioriteiten op nahouden, die mogelijk met elkaar botsen. Ook kunnen werknemers in de verleiding komen om taken door te schuiven naar een andere afdeling in plaats van ze zelf uit te voeren.

Wanneer problemen met coördinatie tussen afdelingen en onderlinge conflicten vermeden worden, kan tijd bespaard worden. De tijd die normaal verloren zou gaan door verschillende afdelingen in een proces te betrekken, kan nu gebruikt worden om taken uit te voeren, waardoor de responstijd van een order zal afnemen en de klant sneller bediend zal worden.

De tijd die verloren gaat aan diverse problemen als gevolg van de betrokkenheid van veel verschillende afdelingen kan vertragingen in het proces veroorzaken, met als gevolg dat wachtrijen kunnen ontstaan.

Reijers en Liman-Mansar (2005) stellen dat er ook nadelen verbonden zijn aan het verminderen van het aantal betrokken afdelingen bij een proces: (1) vermindering van expertise en (2) minder routinevorming.

Door minder afdelingen bij het proces te betrekken, kan geen gebruik gemaakt worden van bepaalde expertise die zich in een afdeling bevindt die geen deel uitmaakt van het proces. Mogelijk kan dit betekenen dat er aan kwaliteit moet worden ingeboet.

Een tweede nadeel van een proces over minder afdelingen te spreiden, is dat elke betrokken afdeling voor een ruimer takenpakket verantwoordelijk is. Hierdoor zal minder routinevorming plaatsvinden dan wanneer er meer taakspecialisatie zou zijn. Dit kan een negatief effect hebben op de tijd die nodig is om het proces te voltooien.

Hoewel bij het ontwerp van een proces zoveel mogelijk controle wordt ingebouwd om de variantie in tijd, kwaliteit en kosten te beperken, kan nooit alle variantie geëlimineerd worden. Om afwijkingen in het proces te kunnen vaststellen en er vervolgens op te reageren, is het belangrijk dat processen gemonitord worden. Het Amerikaanse onderzoeks- en adviesbureau voor informatietechnologie Gartner definieert Business Activity Monitoring (BAM) als “the processes and technologies that enhance situation awareness and enable analysis of critical business performance indicators based on real-time data. BAM is used to improve the speed and effectiveness of business operations by keeping track of what is happening and making issues visible quickly.” (Gartner, 2014)

Business Activity monitoring is dus het meten van procesaspecten. Aspecten die gemeten kunnen worden zijn bijvoorbeeld hoeveelheden, snelheden, fouten en doelstellingen (webMethods, 2006). Hoeveelheden zijn waardes en aantallen met betrekking tot verschillende procesaspecten. Het aantal events in een procesinstantie, de opbrengst van een proces en de kost van een proces zijn voorbeelden van hoeveelheden die gemeten kunnen worden voor een bepaald proces. Snelheden hebben betrekking op het tijdsaspect van processen. Voorbeelden van snelheden zijn wachttijden tussen activiteiten, de duur van een case of de tijd die nodig is voor het uitvoeren van een bepaalde activiteit. Hoeveelheden en snelheden geven belangrijke informatie over een proces, maar het is ook belangrijk om te weten wanneer er iets fout loopt, zodat de oorzaak van de fout gedetecteerd en verholpen kan worden. Het laatste aspect van monitoring is het meten van doelstellingen: de gebruiker kan zelf bepaalde doelstellingen formuleren in de vorm van “Process Performance Indicators” (PPI’s) die gemonitord dienen te worden.

4.1 PROCESS PERFORMANCE INDICATORS

De doelstellingen voor een bedrijfsproces kunnen uitgedrukt worden door middel van Process Performance Indicators (PPI's). PPI's zijn maatstaven die een eigenschap van een proces weerspiegelen. Door aan deze maatstaven doelwaarden te koppelen, kunnen de doelstellingen voor een bepaald proces met betrekking tot een bepaalde eigenschap gespecificeerd worden. Process Performance Indicators zijn dus eigenlijk "Key Performance Indicators" voor bedrijfsprocessen en maken het mogelijk de prestaties van een bedrijfsproces te evalueren.

Del-Rio-Ortega, Resinas en Ruiz-Cortés (2010) stellen dat om deze evaluatie te vergemakkelijken, het noodzakelijk is dat PPI's in de hele levenscyclus van een proces geïntegreerd worden: ze moeten samen met het proces gemodelleerd worden tijdens de ontwerpfase, en tijdens de configuratie-fase moet bepaald worden welke instrumenten nodig zijn om de maatstaven te berekenen. De PPI's kunnen dan berekend worden tijdens de uitvoering van het proces, wat het mogelijk maakt het proces te monitoren.

Om dergelijke integratie van PPI's in de levenscyclus van een proces mogelijk te maken, is het belangrijk dat PPI's formeel gedefinieerd worden. Daarom hebben Del-Rio-Ortega et al. (2010) een ontologie ontwikkeld die dit mogelijk maakt en die volgens hen drie grote voordelen heeft: (1) het verband tussen PPI's en het bedrijfsproces wordt expliciet gelegd, (2) het laat toe een brede waaier aan PPI's te definiëren, zoals PPI's die gerelateerd zijn aan data-objecten en (3) afhankelijkheden tussen procesmaatstaven en instantiemaatstaven kunnen direct uit de ontologie worden afgeleid.

Een Process Performance Indicator heeft een ID, een beschrijving, en een doel. Het is gekoppeld aan een proces en wordt gedefinieerd door de definitie van een maatstaf. Een maatstaf wordt bepaald door een naam en een schaal, die aangeeft welke waarden de maatstaf kan aannemen. Een maatstaf neemt verschillende waardes aan op verschillende tijdstippen (Del-Rio-Ortega et al., 2010).

Binnen de ontologie wordt een classificatie van maatstaven gemaakt op basis van twee criteria: (1) op basis van het aantal procesinstanties waar de maatstaf op berekend wordt en (2) op basis van de manier waarop de waarde van de maatstaf verkregen wordt. In het eerste geval onderscheidt men enerzijds instantiemaatstaven en anderzijds procesmaatstaven, afhankelijk van of de waarde berekend wordt op basis van één enkele procesinstantie dan wel door verschillende instanties te aggregeren. In het tweede geval wordt een onderscheid gemaakt tussen basismaatstaven, geaggregeerde maatstaven en afgeleide maatstaven (Del-Rio-Ortega et al., 2010).

Wanneer de waarde van de maatstaf verkregen wordt door meetmethoden toe te passen op één enkele procesinstantie, spreken we van een basismaatstaf. Voor het bepalen van de waarde van een basismaatstaf wordt dus geen gebruik gemaakt van een andere maatstaf. Een basismaatstaf kan tijd, een aantal of het al dan niet vervuld zijn van bepaalde voorwaarden meten (Del-Rio-Ortega et al., 2010)

Als de waarde van de maatstaf verkregen wordt door een aggregatiefunctie toe te passen op een aantal maatstaven, spreken we van een geaggregeerde maatstaf. Voorbeelden van aggregatiefuncties zijn minimum, maximum, aantal, gemiddelde en som (Del-Rio-Ortega et al., 2010).



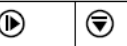

Wanneer de waarde van de maatstaf verkregen wordt door twee of meer maatstafdefinities te combineren met een wiskundige functie, spreken we van een afgeleide maatstaf. Afhankelijk van of de gebruikte maatstaven procesmaatstaven dan wel instantiemaatstaven zijn, spreken we van een afgeleide procesmaatstaf respectievelijk een afgeleide instantiemaatstaf (Del-Rio-Ortega et al., 2010).

4.2 EXTENDED BPMN

In de vorige paragraaf werd een ontologie besproken om PPI's te definiëren en om expliciet het verband te leggen tussen PPI's en de elementen van het proces waarop ze betrekking hebben. Er is dus wel een duidelijke link tussen PPI's en de elementen van het bedrijfsproces, maar het is niet mogelijk om de PPI's aan het grafische procesmodel toe te voegen.


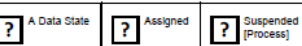
Om het mogelijk te maken BAM-concepten te integreren in procesmodellen, hebben Friedenstab, Janiesch, Matzner en Muller (2012) een uitbreiding op BPMN 2.0 ontwikkeld. Deze uitbreiding maakt gebruik van tien concepten: duur ("duration"), frequentie ("frequency"), sectie ("section"), basismaatstaf ("basic measure"), samengestelde basismaatstaf ("composed basic measure"), geaggregeerde maatstaf ("aggregated measure"), filter ("filter"), doeldefinitie ("target definition"), acties ("actions") en instrumentenbord ("dashboard").

Om op verschillende niveaus aan een proces gerelateerde tijdspannes te meten, wordt een duration-element gedefinieerd. Zo'n duration-element heeft steeds betrekking op iets dat tijd in beslag neemt. Dat kan een heel proces of een activiteit zijn. Om ook meer flexibele tijdspannes dan enkel activiteiten of hele processen te kunnen meten, is er het section-element. Het "Start marker"-element en het "End marker"-element geven het begin respectievelijk het einde van een process section aan (Friedenstab et al., 2012) (Figuur 0.1).

Element Type	Symbol
Duration	
Process Section	 A Process Section
Start Marker	
End Marker	


Figuur 0.1

Het frequency-element telt het aantal keer dat iets gebeurt en is altijd gekoppeld aan iets dat telbaar is. Dat kan een activiteit zijn of het voorkomen van een bepaalde status. In het eerste geval wordt geteld hoeveel keer een bepaalde activiteit tijdens een bepaalde procesinstantie wordt uitgevoerd, in het tweede geval wordt het voorkomen van een bepaalde status bij het uitvoeren van het proces geteld (Friedenstab et al., 2012) (Figuur 0.2).

Element Type	Symbol
Frequency	 A Frequency
State Occurrence	


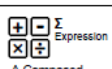
Figuur 0.2

Wanneer meerdere basic measures gecombineerd worden door middel van operatoren zoals optelling, aftrekking, vermenigvuldiging of deling, spreken we van een composed basic measure. Een voorbeeld is bijvoorbeeld de procentuele duur van een activiteit ten opzichte van de totale duur van het proces (Friedenstab et al., 2012) (Figuur 0.3).

Element Type	Symbol
Composed Basic Measure	 Expression A Composed Measure



Figuur 0.3

Een aggregated measure is een maatstaf van een procesinstantie die geaggregeerd wordt, en zo de link vormt tussen instantie en proces. Bij een aggregated measure hoort een “aggregation type”, dat bepaalt hoe de instantiemaatstaf geaggregeerd wordt. Voorbeelden van aggregatietypes zijn minimum, maximum, gemiddelde, som... (Friedenstab et al., 2012) (Figuur 0.4).

Element Type	Symbol
Aggregated Measure	 An Aggregated Measure
Composed Aggregated Measure	 Expression A Composed Aggregated Measure



Figuur 0.4

Om het mogelijk te maken bepaalde maatstaven te berekenen op basis van slechts een deel van de procesinstanties, worden filters gebruikt. Een filter kan twee componenten bevatten: een voorwaarde en/of een inputlimiet. Een voorwaarde laat procesinstanties die niet aan de voorwaarde voldoen buiten beschouwing, terwijl een inputlimiet het aantal procesinstanties dat in beschouwing wordt genomen beperkt. Een inputlimiet kan een kwantitatieve limiet (i.e. enkel een x aantal procesinstanties worden in beschouwing genomen) of een tijdslimiet zijn (i.e. enkel de instanties binnen een bepaald tijds kader worden in overweging genomen) (Friedenstab et al., 2012) (Figuur 0.5).

Element Type	Symbol
Filter	 Process Duration > 6 h  2 Days / 20 Instances A Filter




Figuur 0.5

De target definition maakt het mogelijk om doel- of drempelwaarden voor KPI's te specificeren. Elke maatstaf kan een bijhorende doeldefinitie hebben, die bestaat uit twee componenten: een specificatie van het eigenlijke doel en een optionele specificatie van mogelijke afwijkingen ten opzichte van dat doel (Friedenstab et al., 2012) (Figuur 0.6).

Element Type	Symbol
Target Definition	 > 90 %  Target: < 24 h Deviations: [24 h; 30 h] = Problematic > 30 h = Critical

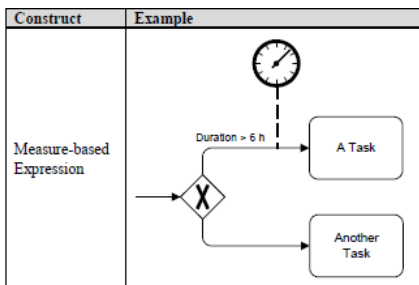
Figuur 0.6

Wanneer de waarde van een bepaalde maatstaf buiten het toegelaten bereik valt, moet het mogelijk zijn hier automatisch op te reageren. Daarom is aan een maatstaf een actie gekoppeld, de maatstaf vormt de basis voor het al dan niet uitvoeren van de bijhorende actie. Een actie kan de vorm aannemen van een “notification” of een “process operation”. Een notification informeert gebruikers en bestaat uit een tekstbericht en een lijst van ontvangers. Een process operator kan een “process trigger”, een “process abortion” of een “process suspension” zijn. Een process trigger start een nieuwe procesinstantie, een process abortion beëindigt een instantie en een process suspension pauzeert een instantie (Friedenstab et al., 2012) (Figuur 0.7).

Element Type	Symbol
Process Trigger	 [Deviation: Critical]
Process Abortion	
Process Suspension	


Figuur 0.7

Om het mogelijk te maken om op basis van een berekende maatstaf te bepalen welk pad in een proces gevolgd wordt, worden uitdrukkingen in BPMN uitgebreid en mogen ze maatstaven bevatten (Friedenstab et al., 2012) (Figuur 0.8).



Figuur 0.8

Om te specificeren welke maatstaven moeten worden weergegeven is er een dashboard-element. Een maatstaf kan bij geen of meerdere dashboards horen en een dashboard kan meerdere maatstaven bevatten (Friedenstab et al., 2012) (Figuur 0.9).

Element Type	Symbol
Dashboard	 A Dashboard

Figuur 0.9

Wanneer tijdens het monitoren van een proces bepaalde afwijkingen ten opzichte van het gewenste gedrag van het proces worden vastgesteld, is het belangrijk dat er ingegrepen kan worden. Daarom moeten processen kunnen omgaan met veranderingen zonder dat hiervoor het procesmodel moet worden aangepast. Het proces moet dus flexibel zijn. Schonenberg, Mans, Russell, Mulyar en Van der Aalst (2008) definiëren procesflexibiliteit als: “the ability to deal with both foreseen and unforeseen changes, by varying or adapting those parts of the business process that are affected by them, whilst retaining the essential format of those parts that are not impacted by the variations” (p. 2).

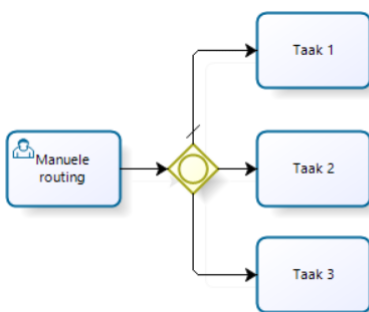
Flexibiliteit kan reeds in de ontwerpfase van het proces mogelijk gemaakt worden. Schonenberg et al. (2008) definiëren flexibiliteit door ontwerp als: “the ability to incorporate alternative execution paths within a process model at design time allowing the selection of the most appropriate execution path to be made at runtime for each process instance” (p. 3).

5.1 FLEXIBILITEIT DOOR ONTWERP: CONTROL FLOW PATTERNS

Het creëren van flexibiliteit door ontwerp, i.e. het door ontwerp mogelijk maken te reageren op veranderingen, kan met behulp van verschillende patronen. Een eerste mogelijkheid is keuze (Schonenberg et al., 2008). Door op bepaalde plaatsen in het proces de mogelijkheid in te bouwen om te beslissen welke van een reeks taken worden uitgevoerd, is het proces in staat te reageren op veranderingen.

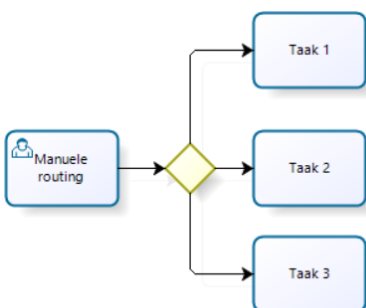
Om dit te realiseren, plaatsen we in het proces een patroon dat keuze toelaat. Wanneer we aan het patroon een zogenaamde “user task” laten voorafgaan, kan een werknemer beslissen welke activiteit of activiteiten vervolgens zullen worden uitgevoerd. Dit geeft nog meer flexibiliteit.

Het keuzepatroon kan in BPMN op verschillende manieren geïmplementeerd worden, afhankelijk van de gewenste keuzevrijheid. Om te modelleren dat één of meer taken uit een verzameling van taken gekozen worden, kan gebruikgemaakt worden van het “multiple choice pattern” (Figuur 0.1). Dit patroon laat echter toe geen, één of meerdere taken uit te voeren. De optie om geen taak te kiezen, kan leiden tot “deadlocks”. Om te vermijden dat het proces vastloopt, kan er aan de hand van een “default flow” voor gezorgd worden dat steeds minstens één taak wordt gekozen.



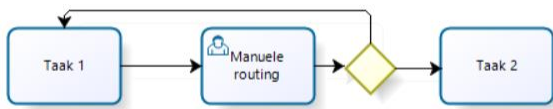
Figuur 0.1

Wanneer we echter willen modelleren dat er slechts één taak gekozen wordt uit een verzameling van taken, kunnen we gebruikmaken van het “exclusive choice pattern” (Figuur 0.2). Dit patroon laat toe precies één activiteit te kiezen uit een verzameling van activiteiten. Opnieuw gaat aan het keuzepatroon een user task vooraf. Het voorzien van een default flow is hier echter overbodig, aangezien het exclusive choice pattern oplegt dat er exact één taak gekozen wordt.



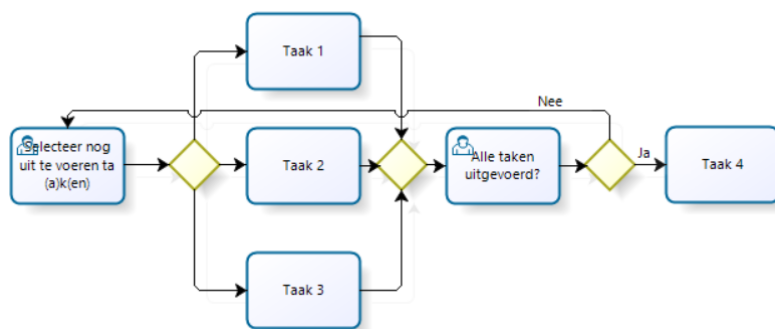
Figuur 0.2

Een tweede mogelijkheid om flexibiliteit door ontwerp te verkrijgen, is door het toelaten van iteratie (Schonenberg et al., 2008): één of meerdere activiteiten kunnen herhaaldelijk worden uitgevoerd. In BPMN kan iteratie geïmplementeerd worden aan de hand van het “exclusive choice pattern” (Figuur 0.3). Ook hier kan aan het keuzepatroon een user task voorafgaan. In onderstaande figuur wordt na het uitvoeren van “Taak 1” een keuze gemaakt tussen het opnieuw uitvoeren van deze taak of direct over te gaan naar “Taak 2”. Wanneer er voor gekozen wordt “Taak 1” opnieuw uit te voeren, zal nadien dezelfde keuze zich opnieuw stellen.



Figuur 0.3

Een derde mogelijkheid tot het verkrijgen van flexibiliteit door ontwerp is aan de hand van het “interleaved routing pattern” (Schonenberg et al., 2008). Dit patroon laat toe om elke taak van een bepaalde verzameling in willekeurige volgorde uit te voeren. Elke taak dient exact één keer te worden uitgevoerd voordat het proces verder gaat. Dit patroon zou gemodelleerd kunnen worden door middel van een zogenaamd “ad-hoc subproces”. Zo’n proces laat toe een verzameling taken in willekeurige volgorde uit te voeren. Het is echter niet duidelijk hoe kan worden afgedwongen dat elke taak exact één keer wordt uitgevoerd alvorens het proces verder gaat. Dit probleem kan verholpen worden door gebruik te maken van een user task. Een werknemer beslist welke taken wanneer worden uitgevoerd, rekening houdend met het feit of een taak reeds is uitgevoerd of niet. In onderstaande figuur selecteert een werknemer één of meerdere nog uit te voeren taken. Na de uitvoering beslist diezelfde werknemer of er nog taken zijn die niet uitgevoerd zijn. Indien dit het geval is, zal er opnieuw een nog uit te voeren taak gekozen worden. Pas wanneer alle taken voltooid zijn, zal het proces verder gaan (Figuur 0.4).



Figuur 0.4

Een vierde mogelijkheid is het gebruikmaken van het “multiple instance pattern” (Schonenberg et al., 2008). Hierbij kunnen verschillende instanties van eenzelfde activiteit uitgevoerd worden. Afhankelijk van of het aantal instanties al dan niet op voorhand gekend is en of alle instanties wel of niet uitgevoerd moeten worden alvorens de volgende activiteit kan plaatsvinden, worden er verschillende “multi instance patterns” onderscheiden.

Wanneer een activiteit verschillende keren moet worden uitgevoerd, maar wanneer niet alle instanties moeten uitgevoerd worden alvorens het proces verder loopt, wordt het “multiple instances without synchronization pattern” gebruikt. Dit patroon wordt in BPMN gemodelleerd door een subproces. Het gewenste gedrag wordt afgedwongen door het instellen van bepaalde parameters. In onderstaande figuur kan nadat “Taak 1” is uitgevoerd, “Taak 2” starten. Deze taak kan herhaaldelijk worden

uitgevoerd. “Taak 3” hoeft echter niet te wachten tot alle instanties van “Taak 2” zijn uitgevoerd alvorens van start te gaan (Figuur 0.5).

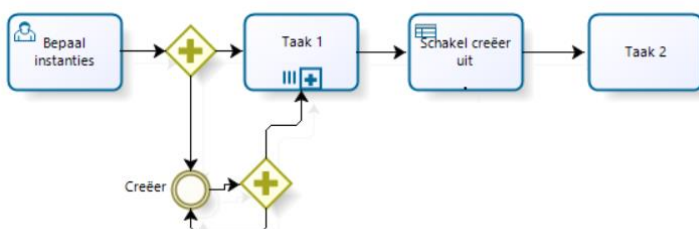


Figuur 0.5

Wanneer het aantal uit te voeren instanties van een bepaalde taak bekend is bij het ontwerpen van het proces en alle instanties uitgevoerd moeten worden alvorens het proces verder loopt, wordt er gebruikgemaakt van het “multiple instances with a priori design-time knowledge pattern”. Dit patroon wordt op dezelfde manier gemodelleerd als het “multiple instances without synchronization pattern”. Het verschil zit in de waarden die aan de verschillende parameters van het subprocess worden gegeven. Deze parameters dwingen af dat een vooraf bepaald aantal instanties van een taak worden uitgevoerd alvorens het proces verder loopt.

Wanneer echter het aantal uit te voeren instanties van een bepaalde activiteit afhangt van een voorgaande activiteit, dan is het aantal uit te voeren instanties van die laatste activiteit pas gekend wanneer het proces loopt. In dat geval wordt het “multiple instances with a priori run-time knowledge pattern” gebruikt. Dit patroon wordt wederom gemodelleerd zoals de twee voorgaande, via parameters wordt het aantal uit te voeren instanties gekoppeld aan de voorgaande activiteit.

Het aantal uit te voeren instanties kan noch bij het ontwerp, noch bij de uitvoering van het proces gekend zijn. In dat geval wordt het “multiple instances without a priori design-time knowledge pattern” gebruikt. Dit patroon laat toe een vooraf bepaald aantal instanties van een taak uit te voeren. Zolang niet al deze instanties zijn uitgevoerd, is het echter mogelijk instanties toe te voegen die ook dienen te worden uitgevoerd alvorens het proces verder gaat. Wanneer er een moment komt dat er geen instanties meer zijn om uit te voeren, gaat het proces verder en wordt de mogelijkheid om nog instanties toe te voegen uitgeschakeld. Dit patroon wordt in BPMN geïmplementeerd met behulp van een “multiple instance task”, een event dat toelaat extra instanties van een taak te creëren en een “business rule task”, die ervoor zorgt dat er geen instanties meer kunnen worden gecreëerd eens er een moment is geweest dat alle instanties uitgevoerd waren. In onderstaande figuur kan nadat het aantal initiële instanties van “Taak 1” bepaald is, gestart worden met de uitvoering ervan. Zolang als er instanties van “Taak 1” in behandeling zijn, is er de mogelijkheid om instanties toe te voegen. Dit wordt mogelijk gemaakt door het “Creëer”-event. Wanneer er een moment is dat er geen instanties meer zijn, wordt deze mogelijkheid afgesloten door het afdwingen van een “business rule” en kan “Taak 2” worden uitgevoerd (Figuur 0.6).



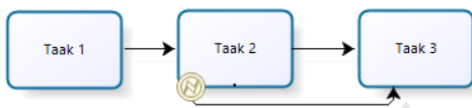
Figuur 0.6

Tot dusver moesten alle instanties van een activiteit uitgevoerd worden, alvorens de volgende activiteit gestart kon worden. Dit is niet altijd het geval. Wanneer het aantal uit te voeren instanties gekend is tijdens het ontwerp en slechts N van de M gecreëerde instanties uitgevoerd moeten worden alvorens een volgende activiteit kan starten ($M > N$), wordt het “static partial join for multiple instances pattern” gebruikt. Een “complex gateway” zorgt ervoor dat minstens N instanties worden uitgevoerd alvorens de volgende activiteit start (Figuur 0.7).



Figuur 0.7

Een laatste manier om flexibiliteit in het proces in te bouwen is door toe te staan dat bepaalde taken geannuleerd worden (Schonenberg et al., 2008). Het “cancel task pattern” laat toe een bepaalde activiteit die uitgevoerd kan worden te annuleren voor ze van start gaat. Wanneer de activiteit al gestart is wordt ze, indien mogelijk, gestopt. Dit patroon wordt in BPMN geïmplementeerd door aan een activiteit een “intermediate error event” vast te maken waarbij het attribuut op “cancel” wordt ingesteld. In onderstaand voorbeeld is er de mogelijkheid om “Taak 2” te annuleren. Nadat “Taak 1” is uitgevoerd is het dus mogelijk om “Taak 2” uit te voeren of om meteen met “Taak 3” te starten (Figuur 0.8).



Figuur 0.8

Het is ook mogelijk om een volledige case te annuleren. Het “cancel case pattern” is sterk gelijkend op het cancel task pattern, met als verschil dat een intermediate error event wordt vastgemaakt aan een subprocess (Figuur 0.9).



Figuur 0.9

5.2 FLEXIBILITEIT DOOR ONTWERP: WORKFLOW ADAPTATION PATTERNS

Wanneer we een proces flexibel willen maken aan de hand van workflow patterns, stelt zich het volgende probleem: met de flexibiliteit stijgt ook de complexiteit van het proces. Dit is in strijd met wat we willen bereiken, aangezien complexere processen minder controleerbaar zijn. Om dit probleem het hoofd te bieden, hebben Zimmermann & Döhring (2011) een benadering ontwikkeld op basis van modellen, regels, gebeurtenissen en aanpassingspatronen. Deze benadering laat toe het procesmodel waar nodig aan te passen tijdens de uitvoering ervan.

Om het procesmodel minder complex te maken, wordt er enkel een eenvoudig basismodel gedefinieerd. Aan dit model worden regels gekoppeld die aangeven welke aanpassingen dienen te worden doorgevoerd aan het basismodel als reactie op bepaalde gebeurtenissen. Naast regels horen bij het model ook patronen, die aangeven hoe het model kan worden aangepast (Zimmermann & Döhring, 2011).

Regels vormen dus de link tussen de context van het proces en de benodigde aanpassing: ze geven aan welke aanpassingen doorgevoerd moeten worden in een bepaalde situatie.

Om het mogelijk te maken een procesmodel aan te passen tijdens de uitvoering, moet tijdens het ontwerp vastgelegd worden welke delen van het proces aangepast kunnen worden. Om dit duidelijk te maken, worden processegmenten die aanpasbaar zijn afgebakend met haken. Wanneer tijdens het uitvoeren van het proces zo'n segment bereikt wordt, wordt een event getriggerd dat leidt tot de evaluatie van de regels die bij dat processegment horen. Wanneer op basis van de geëvalueerde regels een aanpassing nodig blijkt, wordt er een aanpassingspatroon gekozen. Zo'n patroon bepaalt hoe het model moet worden aangepast (Zimmermann & Döhring, 2011).

Zimmermann en Döhring (2011) hebben verschillende aanpassingspatronen ontwikkeld, die altijd zijn opgebouwd volgens een bepaalde syntax. Elk patroon heeft een naam, een omschrijving van het probleem waarop het patroon van toepassing is, een omschrijving van de oplossing die het patroon biedt, een voorbeeld dat het gebruik van het patroon illustreert, een grafische voorstelling van het patroon, parameters die nodig zijn voor het implementeren van het patroon en beperkingen van het patroon.

Drie categorieën van aanpassingspatronen worden onderscheiden: "Basic Adaptation Patterns" beschrijven basisaanpassingen zoals het toevoegen van een element. "Time adaptation Patterns" beschrijven tijdaspecten zoals duur en "Exception Handling Adaptation Patterns" beschrijven hoe met uitzonderingen dient te worden omgegaan.

Flexibiliteit en eenvoud kunnen zo dus gecombineerd worden door te vertrekken van een eenvoudig basismodel, waarbij flexibiliteit verkregen wordt door het interpreteren van regels en het implementeren van aanpassingspatronen.

Hierboven werd uiteengezet hoe met de nodige aandacht voor controle, monitoring en sturing processen voorspelbaar, meetbaar en aanpasbaar worden. Hierdoor kunnen processen beter beheerd worden, wat van groot belang is gezien de rol van bedrijfsprocessen in de transformatie van input naar output en met het oog op de realisatie van bedrijfsdoelstellingen.

Aan de hand van een gevalstudie zullen de hierboven beschreven theoretische concepten aan de praktijk getoetst worden. De gevalstudie is opgebouwd als volgt: in een eerste fase wordt een dertigtal studenten gevraagd een aantal processen te modelleren, meer specifiek de processen voor een autoverhuurbedrijf. Er wordt hen niet gevraagd met bepaalde aspecten zoals controle, monitoring of sturing rekening te houden. In een tweede fase wordt aan de studenten het belang van deze concepten bij het modelleren van processen uitgelegd, en wordt hen vervolgens gevraagd hun procesmodellen aan te passen, rekening houdend met deze concepten.

Meer specifiek wordt hen gevraagd het proces controleerbaar te maken door waar mogelijk de concepten van taakeliminatie, het samenvoegen en opsplitsen van taken, het verplaatsen van taken, het gelijktijdig uitvoeren van taken, het toevoegen van controletaken, bronoptimalisatie, informatieoptimalisatie en het minimaliseren van het aantal betrokken afdelingen toe te passen.

Daarnaast dienen Process Performance Indicators gedefinieerd te worden om het proces te monitoren. Bij gebrek aan de benodigde BPMN-extensie, hoeven deze PPI's niet grafisch te worden weergegeven.

Tot slot wordt hen ook gevraagd het proces zo flexibel mogelijk te maken aan de hand van control flow patterns die flexibiliteit door ontwerp mogelijk maken zoals keuze, herhaling, willekeurige volgorde, meerdere taakinstanties en annuleren van taken of cases.

Aan de hand van de resultaten van de gevalstudie kan dan voor elk aspect van beheerbaarheid worden nagegaan hoe eenvoudig elk deelaspect te implementeren is en welke patronen het meest en welke minder gebruikt worden om een bepaald deelaspect te implementeren. Daarnaast kunnen de procesmodellen van beide fases vergeleken worden om te bepalen welke deelaspecten goed gemodelleerd worden zonder dat hier extra aandacht aan besteed wordt en welke deelaspecten meer aandacht vereisen om er zeker van te zijn dat ze gemodelleerd worden.

In wat volgt wordt eerst de context van de gevalstudie geschetst. Vervolgens wordt voor de drie deelaspecten van beheerbaarheid de mate van aanwezigheid in beide fases van de gevalstudie nagegaan. Daarnaast wordt voor de deelaspecten controle en sturing onderzocht welke patronen het vaakst geïmplementeerd werden en welke patronen minder vaak.

6.1 CONTEXT

Easy Car is een fictief autoverhuurbedrijf dat auto's verhuurt aan klanten volgens verschillende formules. Afhankelijk van de gekozen formule kan de klant kiezen uit bepaalde wagens om te huren en zijn er ook extra diensten beschikbaar zoals verzekeringen tegen diefstal of ongevallen.

De werking van Easy Car kan gemodelleerd worden aan de hand van vier basisprocessen: een reservatieproces, een proces dat de afhaling van de wagen regelt, een proces voor de terugname van de wagen en een facturatieproces. In het reservatieproces worden de klantgegevens geregistreerd en wordt een bepaalde wagen, indien beschikbaar, gereserveerd. Bij de afhaling van de wagen worden alle papieren in orde gemaakt en wordt de wagen aan de klant overhandigd. Wanneer de wagen wordt teruggenomen, wordt er gecontroleerd op schade en worden ook de kilometerstand en het brandstofpeil gecontroleerd. Tijdens de facturatie wordt het bedrag dat aan de klant dient te worden aangerekend bepaald op basis van de afgelegde kilometers en eventuele schade.

6.2 OPTIMALISATIE VAN CONTROLE: TOEPASSING

Wanneer we de controle over een proces van Easy Car willen verbeteren, trachten we het proces voorspelbaar te maken door aan de hand van bepaalde patronen de variantie in tijd, kwaliteit en kost zoveel mogelijk te beperken. De patronen die hiervoor in aanmerking komen zijn, zoals eerder beschreven: het elimineren van taken, het samenvoegen en opsplitsen van taken, het toevoegen van controletaken, het gelijktijdig uitvoeren van taken, het minimaliseren van het aantal betrokken afdelingen, bronoptimalisatie en informatieoptimalisatie.

6.2.1 TAAKELIMINATIE

Taken die geen waarde toevoegen aan het proces kunnen weggelaten worden om het proces te vereenvoudigen. Voorbeelden van zulke taken zijn het afdrukken of kopiëren van een document, het opslaan van een bestand of het archiveren van een dossier. In andere gevallen is een bepaalde taak op zich wel nuttig, maar wegen de kosten van het uitvoeren van de taak niet op tegen de baten. Voorbeelden hiervan zijn het uitvoeren van controles waarbij de kost van het controleren groter is dan de verwachte kost als gevolg van het niet controleren. In onderstaand voorbeeld wordt in het administratieproces van Easy Car een eindrapport opgesteld dat vervolgens gecontroleerd en gearcheeerd wordt. Afhankelijk van de kost die verbonden is aan het uitvoeren van de controle en de gevolgen van een foutief eindrapport, kan de taak “controleer eindrapport” geëlimineerd worden. De taak “Archiveer eindrapport” voegt weinig toe aan het proces en kan daarom worden weggelaten. Zo kunnen drie taken gereduceerd worden tot één enkele taak (Figuur 0.1).



Figuur 0.1

6.2.2 SAMENVOEGEN EN OPSPLITSEN VAN TAKEN

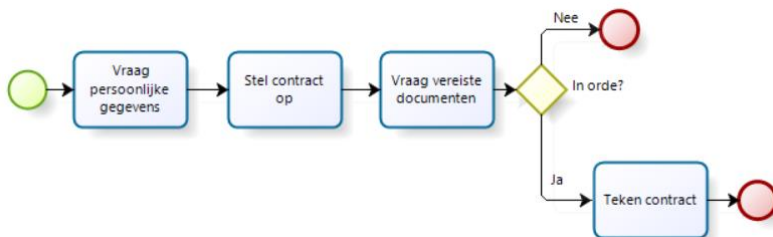
Kleine taken kunnen soms worden samengevoegd tot een grotere taak om het aantal setups te beperken, terwijl het soms beter is te grote taken op te splitsen in meerdere kleinere taken om er voor te zorgen dat een taak goed kan worden uitgevoerd. In onderstaand voorbeeld wordt aan het einde van een huurcontract de wagen terug opgehaald bij de klant. Eens de wagen terug bij de verhuurfirma is, moet deze grondig gecontroleerd worden en daarna dient er een eindrapport te worden opgesteld. Deze drie taken kunnen echter samengevoegd worden tot één meer algemene taak die door één bepaalde werknemer dient uitgevoerd te worden. Zo zullen de gegevens van de klant en de voorwaarden van het contract slechts één keer opgevraagd moeten worden, aangezien het ophalen van de wagen, het controleren van de wagen en het opstellen van het eindrapport één activiteit is die door dezelfde werknemer wordt uitgevoerd (Figuur 0.2).



Figuur 0.2

6.2.3 VERPLAATSEN VAN TAKEN

Soms is het mogelijk taken te verplaatsen waardoor onnodig werk vermeden kan worden. In onderstaand voorbeeld wordt de volledige registratie van een nieuwe klant afgewerkt alvorens hem in de laatste stap van het proces gevraagd wordt de nodige papieren zoals een rijbewijs voor te leggen. Aangezien in het bezit zijn van een rijbewijs een vereiste is om een huurcontract te kunnen afsluiten, kan deze taak verplaatst worden en in plaats van op het einde bij het begin van het proces uitgevoerd worden. Zo zullen onnodige kosten vermeden worden, aangezien het hele registratieproces niet zal moeten worden uitgevoerd wanneer blijkt dat de klant niet in het bezit is van de vereiste papieren (Figuur 0.3).

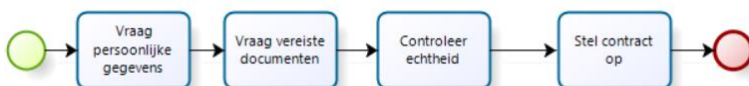


Figuur 0.3

6.2.4 TOEVOEGEN VAN CONTROLETAKEN

Wanneer de baten van het toevoegen van een controletaak de kosten overstijgen, kan extra controle nuttig zijn. Wanneer fouten vroegtijdig worden vastgesteld, kan veel tijd en geld bespaard worden. Wanneer een nieuwe klant bij Easy Car een huurovereenkomst afsluit, kan alles op het eerst gezicht in orde lijken. Omdat de klant over alle vereiste documenten beschikt, wordt een contract opgemaakt en kan de klant zijn wagen in ontvangst nemen. Wanneer achteraf bij het afrekenen echter blijkt dat de klant gebruikmaakte van een valse identiteit en dat ook het opgegeven creditcardnummer vervalst is, blijft Easy Car met een financiële kater achter: in plaats van winst wordt er verlies gemaakt. Dit onverwachte verlies had echter vermeden kunnen worden door in het registratieproces een extra controle in te bouwen: de opgevraagde documenten hadden ook op hun echtheid gecontroleerd moeten worden.

In onderstaand voorbeeld wordt in het registratieproces een controletaak toegevoegd die fraude moet vermijden (Figuur 0.4).



Figuur 0.4

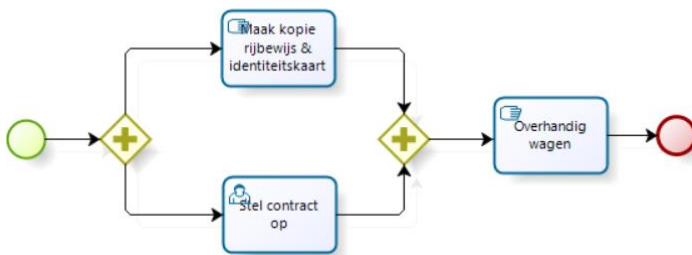
6.2.5 GELIJKTIJDIGE UITVOERING

Door taken die gelijktijdig uitgevoerd kunnen worden in parallel te plaatsen, kunnen wachttijden gereduceerd worden en zal de variantie in tijd afnemen. Wanneer er bijvoorbeeld aan een bepaalde taak twee andere taken voorafgaan, zal wanneer deze taken in sequentie staan de derde taak pas kunnen uitgevoerd worden wanneer de twee voorafgaande taken voltooid zijn. Als er bij de eerste

taak echter iets fout loopt en er moet gewacht worden op een oplossing, zal wanneer het probleem opgelost is eerst de tweede taak nog moeten uitgevoerd worden vooraleer de derde taak kan starten.

Wanneer de eerste twee taken echter in parallel geplaatst worden en hetzelfde scenario doet zich voor, zal de wachttijd voor de derde taak kleiner zijn dan in het eerste scenario. De twee eerste taken kunnen dan immers gelijktijdig worden uitgevoerd, zodat wanneer het probleem bij de eerste taak opgelost is, taak 3 onmiddellijk kan starten en er niet meer gewacht moet worden tot de tweede taak voltooid is. Deze taak werd namelijk al gelijktijdig met de eerste taak uitgevoerd. De wachttijd wordt zo gereduceerd, waardoor de variantie in tijd zal afnemen.

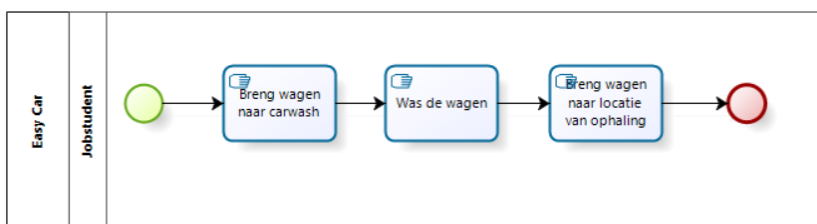
In onderstaand voorbeeld dient er een kopie van het rijbewijs en de identiteitskaart van de klant gemaakt te worden alvorens het contract kan worden opgesteld. Vervolgens kan de wagen aan de klant overhandigd worden. Wanneer deze taken echter in sequentie staan en het kopieerapparaat is stuk, zal wanneer het apparaat hersteld is eerst nog het contract moeten worden opgesteld alvorens de klant zijn wagen kan meenemen. Door de eerste twee taken in parallel te plaatsen wordt het mogelijk om het contract reeds op te stellen wanneer er op de reparatie van het kopieerapparaat gewacht wordt. De klant kan zijn wagen meteen meenemen wanneer het apparaat hersteld is (Figuur 0.5).



Figuur 0.5

6.2.6 BRONOPTIMALISATIE

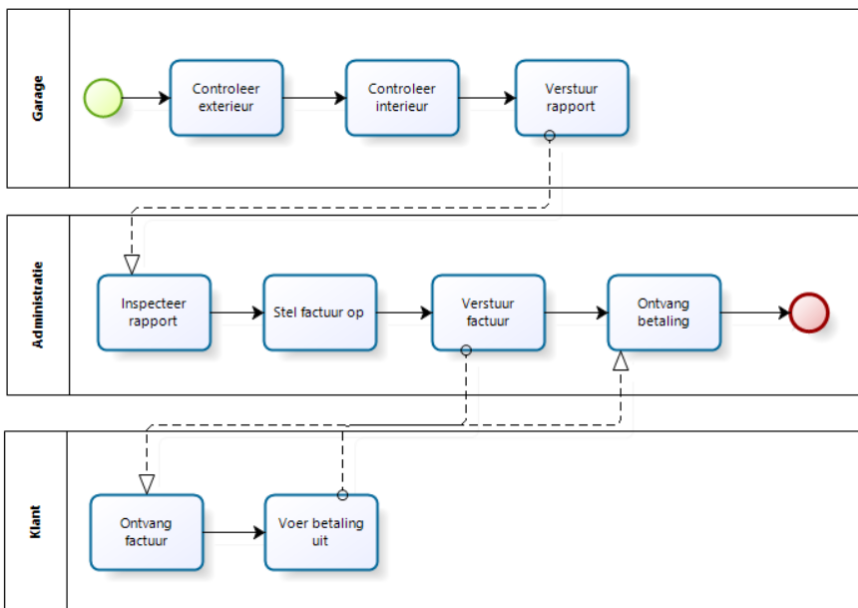
Tijdens drukke periodes kunnen bepaalde afdelingen overbelast raken. Om dit te vermijden, kunnen jobstudenten ingezet worden om afdelingen te ontlasten. Op deze manier kan de controle over het proces verbeterd worden, mits de jobstudenten op de juiste manier worden ingezet. Wanneer taken waarvoor een bepaalde expertise of ervaring vereist is aan jobstudenten worden toegewezen, kan dit net een averechts effect hebben. De jobstudent zal de taak niet naar behoren kunnen uitvoeren, waardoor tijd verloren gaat en mogelijk extra kosten gemaakt moeten worden. Een voorbeeld hiervan is het inzetten van jobstudenten voor het wassen van de wagens van Easy Car. Hoewel het wassen van wagens op het eerste gezicht een taak lijkt die uitermate geschikt is voor uitvoering door jobstudenten, is dit niet het geval. De wagens moeten immers ook naar de carwash gebracht worden. Vooral bij dure en/of snelle wagens houdt dit een risico in, waardoor het belangrijk is dat deze taak wordt uitgevoerd door meer ervaren werknemers (Figuur 0.6).



Figuur 0.6

6.2.7 INFORMATIEOPTIMALISATIE

Om een proces goed te kunnen uitvoeren, is het belangrijk dat alle betrokken partijen over de nodige informatie beschikken. In onderstaand voorbeeld wordt na de verhuur een wagen gecontroleerd op eventuele schade aan de carrosserie alsook op beschadigingen aan het interieur. Het is namelijk belangrijk dat wanneer er schade aan de wagen wordt vastgesteld, de kosten voor herstel aan de klant worden doorgerekend, bovenop het te betalen bedrag voor de huur van de wagen. Om dit mogelijk te maken, moet de administratie door de garage op de hoogte gebracht worden van het feit of er al dan niet beschadigingen zijn vastgesteld. De garagist bundelt daarom zijn bevindingen in een rapport dat naar de administratie verstuurd wordt, zodat die de extra kosten in rekening kan brengen. Om de betaling te ontvangen, dient de administratie op haar beurt te communiceren met de klant door die een factuur te sturen zodat de klant kan betalen en de administratie de betaling kan afhandelen (Figuur 0.7).



Figuur 0.7

6.3 OPTIMALISATIE VAN CONTROLE: RESULTATEN

Om na te gaan in welke mate rekening gehouden werd met het deelaspect controle zonder dat de studenten op het belang hiervan attent gemaakt werden, kunnen we enkel uitspraken doen over het gelijktijdig uitvoeren van taken. Enkel voor dit patroon kunnen we het aantal keer dat het patroon geïmplementeerd werd in de verschillende fases met elkaar vergelijken. De andere patronen voor het bevorderen van controle zijn immers niet ex-post waarneembaar: zo is het bijvoorbeeld onmogelijk na te gaan hoeveel keer taken werden samengevoegd of hoeveel taken werden geëlimineerd tijdens de eerste fase.

Onderstaande tabel geeft voor beide fases het aantal keer weer dat taken in parallel geplaatst werden in de processen van de verschillende cases (Tabel 0.1).

Case	1	2	3	4	5	6	7	8	9	10
Fase 1	8	0	3	0	2	2	5	0	0	1
Fase 2	10	2	3	0	2	4	5	0	4	2

Tabel 0.1

Aangezien slechts in twee gevallen het aantal gelijktijdige uitvoeringen van taken verbeterd kon worden, kunnen we stellen dat aan dit patroon voldoende aandacht wordt besteed. Wat opvalt, is dat in case 4 geen enkele keer taken gelijktijdig werden uitgevoerd, en er hier ook geen verbetering mogelijk was. Een verklaring ligt in de granulariteit van het procesmodel: de taken werden hier heel algemeen omschreven.

Om na te gaan in welke mate de andere patronen van controle geïmplementeerd werden, moeten we ons baseren op het aantal verbeteringen die nog mogelijk waren. Onderstaande tabel geeft het aantal verbeteringen voor de verschillende patronen (Tabel 0.2).

Case	1	2	3	4	5	6	7	8	9	10
Elimineren van taken	1	3	3	0	1	1	3	0	1	0
Samenvoegen/opsplitsen van taken	6	7	0	1	3	4	1	6	5	5
Verplaatsen van taken	1	0	0	0	0	1	0	2	0	0
Toevoegen controletaken	0	0	0	0	0	0	0	0	2	1
Bronoptimalisatie	0	0	1	0	0	0	0	0	0	0
Informatieoptimalisatie	1	0	1	0	0	1	0	0	0	0
Minimaliseren van # betrokken afdelingen	0	0	0	0	0	0	0	0	0	0

Tabel 0.2

Uit bovenstaande tabel kunnen we concluderen dat de controle over een proces vooral nog verbeterd kon worden door taken samen te voegen of op te splitsen en door taken te elimineren. We stellen vast dat reeds in de eerste fase genoeg controlemechanismen werden ingebouwd, aangezien er geen controletaken werden toegevoegd. In geen enkel geval kon het aantal betrokken afdelingen verminderd worden. Dit is te verklaren doordat er meestal niet meer dan twee afdelingen betrokken waren bij het proces. Bronoptimalisatie is dan weer moeilijk in de praktijk te brengen doordat er weinig informatie beschikbaar is over wie welke taak zal uitvoeren.

Voor het deelaspect controle kunnen we besluiten dat het samenvoegen/opsplitsen en het elimineren van taken het makkelijkst te implementeren zijn. Het minimaliseren van het aantal afdelingen, bronoptimalisatie en informatieoptimalisatie zijn moeilijk in de praktijk te brengen in processen waarbij niet veel verschillende afdelingen betrokken zijn.

6.4 OPTIMALISATIE VAN MONITORING: TOEPASSING

Hoewel er zoveel mogelijk controle is ingebouwd bij het ontwerp van het proces, moet het proces ook gemonitord worden om zo ongewenst gedrag te kunnen vaststellen en er op te kunnen reageren. Door geschikte Process Performance Indicators (PPI's) te definiëren kunnen de processen van Easy Car gemonitord worden op eventuele onregelmatigheden. In onderstaande tabel worden een aantal PPI's voor de processen van Easy Car gedefinieerd (Tabel 0.3).

ID	Beschrijving	Doel	Maatstaf	
			proces/instantie	basis/geaggregeerd/afgeleid
PPI 1	Percentage wagen niet beschikbaar	<5%	proces	basis (samengesteld)
PPI 2	Gemiddelde duur betaling	<2 weken	proces	geaggregeerd
PPI 3	Percentage ontevreden klanten	<5%	proces	basis (samengesteld)
PPI 4	Gemiddelde duur controle wagen	<5 min	proces	geaggregeerd

Tabel 0.3

6.4.1 PPI 1: PERCENTAGE WAGEN NIET BESCHIKBAAR

Wanneer een klant een huurovereenkomst sluit, heeft hij de keuze uit een ruim aantal wagens. Het kan echter voorvallen dat een bepaalde wagen die normaal deel uitmaakt van het aanbod, niet meer beschikbaar is. Dit kan verschillende redenen hebben: alle beschikbare exemplaren van een bepaald model zijn verhuurd of een specifiek model is wel nog beschikbaar, maar is al gereserveerd vanaf een datum die voor de einddatum van het contract van de nieuwe klant valt. Een andere mogelijke oorzaak is dat de wagen niet beschikbaar is als gevolg van een defect. Wat de oorzaak ook zij, deze situatie moet zoveel mogelijk vermeden worden om de klant tevreden te houden.

Een eerste Proces Performance Indicator berekent het percentage van de cases waarin de gevraagde wagen niet beschikbaar is. Voor elke procesinstantie wordt bijgehouden of de klant wel of niet kan beschikken over de wagen van zijn eerste keuze. Periodiek kan het aantal keer dat dit niet het geval was gedeeld worden door het totale aantal procesinstanties. De samengestelde basismaatstaf die zo verkregen wordt, geeft het percentage van de cases waarin een klant niet over de wagen van zijn keuze kan beschikken. Wanneer dit percentage hoger is dan bijvoorbeeld 5%, dient de oorzaak van dit hoge percentage onderzocht te worden om in de toekomst het voorvallen van dergelijke situaties zoveel mogelijk te beperken.

6.4.2 PPI 2: GEMIDDELDE DUUR BETALING

Easy Car staat haar klanten een maximum betalingstermijn van één maand toe. Teneinde aan haar financiële verplichtingen te kunnen voldoen, moet er echter steeds genoeg cash voorhanden zijn. Daarom rekent Easy Car erop dat een groot deel van haar klanten reeds binnen de twee weken betaalt. Wanneer een groot deel van het cliënteel echter de toegestane termijn ten volle benut, kan Easy Car in de problemen komen. Daarom is het nodig dat de gemiddelde tijd die verstrijkt tussen het zenden van de factuur en het ontvangen van de betaling gemonitord wordt. Door voor elke instantie de tijd

tussen het zenden van de factuur en het ontvangen van de betaling te meten en hier het gemiddelde van te nemen, kan Easy Car voorspellen of de positieve cashflow groot genoeg is om aan haar verplichtingen op korte termijn te voldoen. Wanneer het gemiddelde bijvoorbeeld meer dan twee weken bedraagt, moet de maximaal toegestane betalingstermijn verlaagd worden om niet in de problemen te komen.

6.4.3 PPI 3: PERCENTAGE ONTEVREDEN KLANTEN

Wanneer de klant zijn wagen terug inlevert, wordt hem gevraagd een enquête in te vullen die de mate van klanttevredenheid toetst. Door voor elke klant de ingevulde enquête te analyseren en bij te houden of hij globaal gezien tevreden of ontevreden was, kan periodiek het percentage tevreden en het percentage ontevreden klanten bepaald worden. Wanneer bijvoorbeeld blijkt dat meer dan 5% van de klanten uitgesproken ontevreden is, moet de oorzaak hiervan verder onderzocht worden en moeten aanpassingen worden doorgevoerd om het percentage tevreden klanten te doen stijgen.

6.4.4 PPI 4: GEMIDDELDE DUUR CONTROLE WAGEN

Wanneer een klant zijn huurwagen terug inlevert, vindt er steeds onmiddellijk een controle plaats om eventuele schade vast te stellen. Om discussie achteraf te vermijden, wordt het bewijs van inlevering pas na de controle aan de klant overhandigd. Aangezien de klant dus moet wachten tot de controle voltooid is alvorens hij kan vertrekken, is het belangrijk dat die controle efficiënt gebeurt. Daarom dient de tijd die nodig is voor het uitvoeren van de controle bijgehouden te worden. Wanneer periodiek de gemiddelde benodigde tijd bepaald wordt en blijkt dat die bijvoorbeeld meer dan vijf minuten bedraagt, moet er naar oplossingen gezocht worden om de controle sneller te kunnen uitvoeren zodat de klant sneller geholpen wordt.

6.5 OPTIMALISATIE VAN MONITORING: RESULTATEN

Tijdens de eerste fase van het modelleren werd in geen van de tien cases rekening gehouden met Proces Performance Indicators. Nadat de studenten op het belang hiervan attent gemaakt werden, werden in alle cases nuttige PPI's gevonden om de processen te monitoren.

De meest voorkomende PPI's die werden toegevoegd werden hierboven besproken. Ze hadden betrekking op de beschikbaarheid van de wagens, de tijd die verstrijkt tussen het opsturen van de factuur en het ontvangen van de betaling, de tevredenheid van klanten en de duur van controles. De tabel hieronder geeft voor elke case het aantal PPI's in de eerste fase en in de tweede fase. Zoals gezegd werd in de eerste fase door geen enkele groep rekening gehouden met PPI's, waardoor het aantal PPI's voor elke case gelijk is aan nul in fase 1. Elke groep hield rekening met PPI's tijdens de tweede fase (Tabel 0.4).

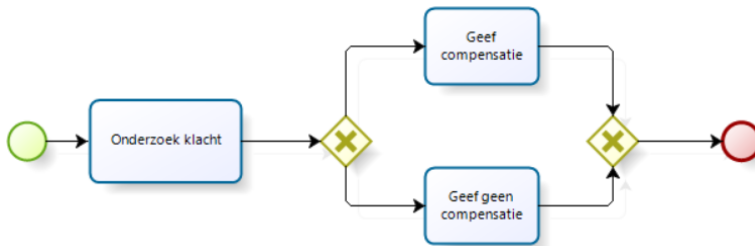
Case	1	2	3	4	5	6	7	8	9	10
Fase 1	0	0	0	0	0	0	0	0	0	0
Fase 2	1	5	4	2	3	2	2	2	2	5

Tabel 0.4

6.6 OPTIMALISATIE VAN STURING: TOEPASSING

6.6.1 KEUZE

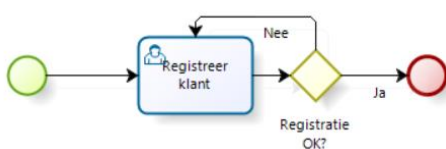
Afhankelijk van de uitkomst van een bepaalde activiteit, moet het dikwijls mogelijk zijn om twee of meerdere verschillende paden te volgen in het proces. Wanneer een klant van Easy Car niet tevreden is over de geleverde service, kan hij een klacht indienen. De klacht zal dan door de klantendienst onderzocht worden, en afhankelijk van of de klacht gegrond of ongegrond bevonden wordt, zal er wel of geen compensatie volgen voor de klant. In onderstaand voorbeeld wordt dit scenario geïmplementeerd met behulp van keuze (Figuur 0.8).



Figuur 0.8

6.6.2 ITERATIE

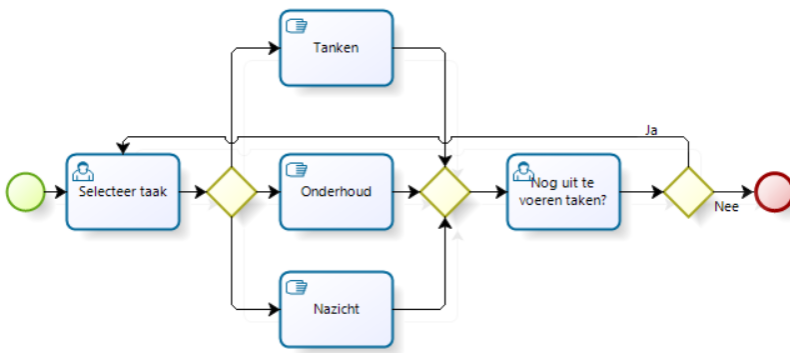
Tijdens het registreren van een klant moeten verschillende gegevens in het systeem worden ingevoerd, zoals het identiteitskaartnummer en het rijbewijsnummer. Wanneer bij het voltooien van de registratie bijvoorbeeld blijkt dat het ingevoerde identiteitskaartnummer een ongeldig formaat heeft, moet het mogelijk zijn het registratieproces te hernemen, en dit totdat alle gegevens correct zijn ingevoerd. Om dit mogelijk te maken moet iteratie in het proces mogelijk gemaakt worden. In onderstaande figuur wordt het registratieproces net zolang hernomen tot er geen fouten meer zijn (Figuur 0.9).



Figuur 0.9

6.6.3 INTERLEAVED ROUTING

Soms is het mogelijk om bepaalde taken in willekeurige volgorde uit te voeren. Wanneer een wagen wordt ingeleverd, moeten er standaard drie zaken gebeuren: de wagen moet gewassen worden, de wagen moet een onderhoudsbeurt krijgen en de wagen moet terug volgetankt worden. De volgorde waarin deze activiteiten uitgevoerd worden, speelt echter geen rol. In onderstaande figuur kunnen de drie taken in willekeurige volgorde worden uitgevoerd. Alle taken dienen uitgevoerd te zijn voor het proces verder gaat, en er kunnen geen taken gelijktijdig uitgevoerd worden (Figuur 0.10).



Figuur 0.10

6.6.4 ANNULATIE

Soms kan een taak geannuleerd worden omdat uit een voorgaande taak blijkt dat het niet meer nodig is die bepaalde taak uit te voeren. Wanneer een klant zich bijvoorbeeld meldt en het blijkt uit zijn persoonsgegevens dat deze klant al eerder een wagen gehuurd heeft, dan is het overbodig om zijn rijbewijs en identiteitskaart te vragen. Deze taak kan geannuleerd worden en het contract kan onmiddellijk opgesteld worden, aangezien Easy Car reeds over alle benodigde gegevens van de klant beschikt. In onderstaand voorbeeld wordt dit scenario gemodelleerd aan de hand van een intermedie error event dat op cancel wordt ingesteld (Figuur 0.11).



Figuur 0.11

6.7 OPTIMALISATIE VAN STURING: RESULTATEN

Tijdens de eerste fase werd bijna uitsluitend de meest voor de hand liggende vorm van flexibiliteit geïmplementeerd door de verschillende groepen: keuze. De helft van de groepen modelleerden ook iteratie. Slechts één groep maakte gebruik van het interleaved routing pattern. Het mogelijk maken om meerdere instanties van een activiteit uit te voeren werd in de eerste fase door geen enkele groep geïmplementeerd en ook met de mogelijkheid om taken te annuleren werd geen rekening gehouden (Tabel 0.5).

Case	1	2	3	4	5	6	7	8	9	10
Keuze	27	20	5	21	26	17	20	12	21	9
Iteratie	2	0	0	2	0	2	1	1	0	0
Interleaved routing	0	0	0	0	0	1	0	0	0	0
Meerdere instanties	0	0	0	0	0	0	0	0	0	0
Annulatie	0	0	0	0	0	0	0	0	0	0

Tabel 0.5

Tijdens de tweede fase implementeerden zeven groepen iteratie, terwijl drie groepen gebruikmaakten van interleaved routing. Ook tijdens de tweede fase maakte geen enkele groep gebruik van de mogelijkheid om meerdere instanties van een activiteit uit te voeren en slechts twee groepen maakten gebruik van de mogelijkheid om activiteiten te annuleren (Tabel 0.6).

Case	1	2	3	4	5	6	7	8	9	10
Keuze	27	21	6	21	26	17	20	12	21	9
Iteratie	3	0	1	2	0	2	2	1	0	1
Interleaved routing	0	0	0	0	1	2	1	0	0	0
Meerdere instanties	0	0	0	0	0	0	0	0	0	0
Annulatie	0	1	1	0	0	0	0	0	0	0

Tabel 0.6

We kunnen dus besluiten dat flexibiliteit het makkelijkst geïmplementeerd wordt door keuze. Iteratie en interleaved routing worden minder gebruikt, en van de mogelijkheid om activiteiten te annuleren of om er meerdere instanties van uit te voeren wordt weinig tot geen gebruik gemaakt.

In deze masterproef werden drie deelaspecten van beheerbaarheid onderscheiden: controle, monitoring en sturing. Het beheren van bedrijfsprocessen werd dus gezien als het bewaren van de controle over de processen, het uitvoeren van metingen om na te gaan of de processen nog steeds onder controle zijn en het eventueel bijsturen van de processen indien bepaalde afwijkingen zich voordoen.

Voor elk van bovengenoemde deelaspecten werd aan de hand van een literatuurstudie nagegaan wat het deelaspect precies inhield en hoe het verbeterd kon worden. Controle werd gedefinieerd als voorspelbaarheid met betrekking tot tijd, kosten en kwaliteit. Dit aspect kan verbeterd worden door processen te vereenvoudigen, taken te elimineren, samen te voegen of op te splitsen en te verplaatsen. Ook het gelijktijdig uitvoeren van taken, bronoptimalisatie, informatieoptimalisatie, het toevoegen van controletaken en het minimaliseren van het aantal betrokken afdelingen kan de controle over een proces verbeteren. Monitoring werd gedefinieerd als het meten van procesaspecten. Door het formuleren van zogenaamde “Process Performance Indicators” (PPI’s) kunnen processen beter gemonitord worden. Om het ook mogelijk te maken om PPI’s in een procesmodel te integreren, werd een uitbreiding van BPMN 2.0 besproken die dit toestaat. Sturing werd gedefinieerd als de mogelijkheid om in te grijpen wanneer afwijkingen ten opzichte van het gewenste gedrag worden vastgesteld. Om dit te bereiken moeten processen flexibel zijn. Het creëren van flexibiliteit door ontwerp, i.e. het door ontwerp mogelijk maken te reageren op veranderingen, kan met behulp van verschillende “workflow patterns”: “choice”, “iteration”, “interleaved routing”, “multiple instances activities” en “cancelation”.

De bevindingen uit de literatuurstudie werden toegepast op een gevalstudie om de beheerbaarheid van concrete processen te verbeteren. Voor het deelaspect controle kunnen we besluiten dat het samenvoegen/opsplitsen en het elimineren van taken het makkelijkst te implementeren zijn. Het minimaliseren van het aantal afdelingen, bronoptimalisatie en informatieoptimalisatie zijn moeilijk in de praktijk te brengen in processen waarbij niet veel verschillende afdelingen betrokken zijn. Voor wat het formuleren van PPI’s betreft, stelden we vast dat er pas rekening mee gehouden werd nadat studenten op het belang ervan attent gemaakt werden. Flexibiliteit ten slotte, werd het makkelijkst geïmplementeerd door keuze. Iteratie en interleaved routing werden minder gebruikt, en van de mogelijkheid om activiteiten te annuleren of om er meerdere instanties van uit te voeren werd weinig tot geen gebruik gemaakt.

- Buzacott, J.A. (1996). Commonalities in reengineered business processes: models and issues. *Management Science*, 42, 768-782. doi: 10.1287/mnsc.42.5.768
- Cabanillas, C., Resinas, M., & Ruiz-Cortés, A. (2011). Mixing RASCI matrices and BPMN together for responsibility management. *VII Jornadas en Ciencia e Ingeniería de Servicios (JCIS 2011)*, 1, 167-180.
- Cardoso, J., Garcia, F., Piattini, M., Rolón, E., & Ruiz F. (2009). Analysis and validation of control-flow complexity measures with BPMN process models. *Lecture Notes in Business Information Processing*, 29, 58-70. doi: 10.1007/978-3-642-01862-6_6
- Castano, S., & De Antonellis, V. (1998). A framework for expressing semantic relationships between multiple information systems for cooperation. *Information Systems*, 23, 253-277. doi: 10.1016/S0306-4379(98)00012-X
- Castano, S., De Antonellis, V., & Melchiori, M. (1999). A methodology and tool environment for process analysis and reengineering. *Data & Knowledge Engineering*, 31, 253-278. doi: 10.1016/S0169-023X(99)00028-2
- Del-Rio-Ortega, A., Resinas, M., & Ruiz-Cortés, A. (2010). Defining process performance indicators: an ontological approach. *Lecture Notes in Computer Science*, 6426, 555-572. doi: 10.1007/978-3-642-16934-2_41
- Friedenstab, J.-P., Janiesch, C., Matzner, M., & Muller, O. (2012). Extending BPMN for business activity monitoring. *Proceedings of the 2012 45th Hawaii International Conference on System Sciences*, 0, 4158-4167. doi: 10.1109/HICSS.2012.276
- Furnas, G.W., Landauer, I.K., Gomez, L.M., & Dumais, S.T. (1987). The vocabulary problem in human-system communication. *Communications of the ACM*, 30, 964-971.
- Hammer, M., & Champy J. (1993). *Reengineering the corporation: a manifesto for business revolution*. New York, NY: Harper Business Editions.

- Henry, S., & Kafura, K. (1981). Software structure metrics based on information flow. *IEEE Transactions on Software Engineering*, 7, 510-518. doi: 10.1109/TSE.1981.231113
- Jeston, J., & Nelis, J. (2006). *Business process management: practical guidelines to successful implementations*. Oxford: Butterworth-Heinemann.
- Laue, R., & Gruhn, V. (2006). Complexity metrics for business process models. *Lecture Notes in Informatics*, 85, 1-12.
- McCormack, K., & Johnson, W.C. (2001). *Business process orientation: gaining the e-business competitive advantage*. Florida: St. Lucie Press.
- Petrovich, M. (2000). Strategies for improvement of process control. Geraadpleegd op <http://www.mvpprograms.com/docs/procctrl.pdf>
- Reijers, H.A. (2003). Design and control of workflow processes. *Lecture Notes in Computer Science*, 2617. doi: 10.1007/3-540-36615-6
- Reijers, H.A. & Liman-Mansar, S. (2005). Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. *Omega*, 33, 283-306. doi: 10.1016/j.omega.2004.04.012
- Russell, N., ter Hofstede, A.H.M., van der Aalst, W.M.P., Mulyar, N. (2006). Workflow control-flow patterns. a revised view. Geraadpleegd op <http://www.workflowpatterns.com/documentation/documents/BPM-06-22.pdf>
- Schonenberg, H., Mans, R., Russell, N., Mulyar, N., & van der Aalst, W. (2008). Process flexibility: a survey of contemporary approaches. *Lecture Notes in Business Information Processing*, 10, 16-30. doi: 10.1007/978-3-540-68644-6_2
- Smith, M.L., & Erwin, J. (2007). Role & responsibility charting (RACI). Geraadpleegd op http://myclass.peelschools.org/sec/12/4268/Resources/RACI_R_Web3_1.pdf
- Van der Aalst, W.M.P. (2001). Re-engineering knock-out processes. *Decision Support Systems*, 30, 451-468. doi: 10.1016/S0167-9236(00)00136-6

Wade, D. & Recardo, R. (2001). *Corporate performance management: how to build a better organization through measurement-driven, strategic alignment*. Woburn, MA: Butterworth-Heinemann.

webMethods. (2006). Business activity monitoring (BAM): the new face of BPM.

Geraadpleegd op http://www.business.unr.edu/faculty/kuechler/788/bam-the_new_face_of_bpm.pdf

Weske, M. (2007). Business Process Management Concepts, Languages, Architectures.

doi:10.1007/978-3-642-28616-2

Weske, M., van der Aalst, W. M., & Verbeek, H. M. (2004). Advances in business process management. *Data & Knowledge Engineering*, 50, 1-8.

Zimmermann, B., & Döhring, M. (2011). Patterns for flexible BPMN workflows. In EuroPLoP,

Proceedings of the 16th European Conference on Pattern Languages of Programs. doi: 10.1145/2396716.2396723

Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:

Het verbeteren van de beheerbaarheid van bedrijfsprocessen: een framework voor het middle management

Richting: **master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica**

Jaar: **2014**

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,

Huskens, Jan

Datum: **1/06/2014**