

# **PETRINET+**

*EEN ANALYSE MIDDELS METAMODELLERING*

**DELFT, JUNI 2002**

Abdul Boztas  
Jos de Bruijn  
Mark Dumay  
Dieter Hoenke  
Eyal Oren

**Technische Universiteit Delft**  
**Faculteit Informatie Technologie en Systemen**  
Afdeling Technische Informatica



## Voorwoord

Dit rapport behoort bij het vak Meta Modeling & Method Engineering, welke onderdeel uitmaakt van de doctoraalfase van de opleiding Technische Informatica. Doel van het vak is inzicht geven in het vakgebied bedrijfsprocesmodellering. Hiertoe wordt kennis opgedaan met een aantal modelleertechnieken, om daarna tot een evaluatie van deze technieken te komen.

Om een goede evaluatie mogelijk te maken, worden er middels een methode metamodellen van de modelleertechnieken opgesteld. In dit rapport wordt ingegaan op de modelleertechniek Petrinet+, welke is gebaseerd op de reeds bestaande petrinet techniek.

Abdul Boztas,  
Jos de Bruijn,  
Mark Dumay,  
Dieter Hoenke en  
Eyal Oren.

Delft, juni 2002.

## **Inhoudsopgave**

<b>VOORWOORD</b>	<b>2</b>
<b>1 INLEIDING</b>	<b>4</b>
<b>2 THEORETISCHE BESCHOUWING VAN MODELLEERTECHNIKEN</b>	<b>5</b>
2.1 DE SAMENHANG TUSSEN ASPECT EN MODELLEERTECHNIEK	5
2.2 DE TAAL CAPTURE MODELS FOR EVALUATION (C-ME)	6
2.3 KEUZE VOOR C-ME	7
<b>3 BESCHRIJVING VAN PETRINET+</b>	<b>8</b>
3.1 PETRINET ALS UITGANGSPUNT VOOR PETRINET+	8
3.2 TOEVOEGINGEN EN WIJZIGINGEN IN PETRINET+	9
3.3 GENERIEKE CONCEPTEN EN MOGELIJKE CONSTRUCTIES IN PETRINET+	9
<b>4 METAMODEL BESCHRIJVING VAN PETRINET+</b>	<b>15</b>
<b>5 KWALITATIEVE ANALYSE PETRINET+</b>	<b>18</b>
5.1 VOORDELEN VAN PETRINET+	18
5.2 NADELEN VAN PETRINET+	18
5.3 SUGGESTIES VOOR VERBETERING	20
5.4 KOPPELING AAN EEN DATA MODEL	21
<b>6 CONCLUSIE</b>	<b>22</b>
<b>REFERENTIES</b>	<b>24</b>
<b>BIJLAGE – VOORBEELDUITWERKING PETRINET+</b>	<b>25</b>

# 1 Inleiding

Dit rapport geeft een evaluatie van de modelleertechniek Petrinet+, welke wordt gebruikt om bedrijfsprocessen te modelleren. Petrinet+ vormt een uitbreiding op de techniek petrinet, en is door het GAK intern opgesteld [02]. De evaluatie heeft als doel het kwaliteitsniveau van de modelleertechniek vast te stellen. Dit wordt gedaan door de aangeleverde specificatie hiervan te formaliseren.

Met de methode C-Me [04] wordt op basis van deze formele specificatie een metamodel opgesteld. Middels dit model is het mogelijk Petrinet+ te vergelijken met andere bedrijfsprocesmodelleertechnieken. Dit valt echter buiten de grenzen van dit onderzoek. Wel worden op basis van dit metamodel en de opgedane praktische ervaring met Petrinet+ aanbevelingen gedaan om deze methode waar mogelijk te verbeteren.

Allereerst wordt in dit rapport een theoretische beschouwing gegeven van het concept modelleertechniek. Hier wordt tevens de methode C-Me toegelicht. Vervolgens wordt de Petrinet+ techniek beschreven aan de hand van de verschillende elementen die worden ondersteund. Op basis van deze beschrijving wordt vervolgens een formeel metamodel opgesteld. Dit wordt gevolgd door een kwalitatieve analyse van Petrinet+. Tenslotte worden de conclusies en aanbevelingen weergegeven.

## 2 Theoretische beschouwing van modelleertechnieken

In dit hoofdstuk beschrijven we de theorie van modelleertechnieken. Dit gebeurt allereerst door te kijken naar de samenhang tussen een reëel fenomeen en aspectmodellen. Vervolgens wordt de taal C–Me toegelicht, welke het mogelijk maakt verschillende multimodelleertechnieken gestructureerd te analyseren. Tot slot geven we een verantwoording waarom we voor dit rapport gebruik maken van deze taal.

### 2.1 De samenhang tussen aspect en modelleertechniek

Een model is een vereenvoudigde weergave van een fenomeen in de werkelijke wereld. Volgens Hommes [04] dienen modelleertechnieken om te helpen bij de constructie van conceptuele modellen. Dit doen zij door zowel een bepaalde taal aan te bieden waarin de kennis over die fenomenen uitgedrukt kan worden, alsmede een procedure aan te bieden waarmee die kennis verzameld kan worden.

Het is mogelijk het fenomeen vanuit één bepaald oogpunt te benaderen en vanuit dat aspect een model op te stellen. Als we bijvoorbeeld een bakkerij gaan bekijken puur vanuit het oogpunt van de boekhouder, dan kunnen we een model opstellen van in- en uitgaande geldstromen en hebben we een model opgesteld van de bakkerij als financieel apparaat.

Maar de bakkerij is natuurlijk ook vanuit een ander oogpunt te modelleren, bijvoorbeeld als logistiek proces waarin grondstoffen door middel van een aantal processtappen tot product worden omgezet. Van de bakkerij is dus ook een ander model op te stellen; het verschil zit in het aspect waarop de modelleertechniek zich richt. Technieken die zich – zoals in de voorbeelden – richten op één bepaald aspect van het te modelleren fenomeen noemen we aspectmodelleertechnieken. Maar omdat beide aspectmodellen gaan over dezelfde situatie in de werkelijke wereld, zullen ze op de één of andere manier concepten (uit die wereld) met elkaar gemeen hebben.

Multimodelleertechnieken maken hiervan gebruik om een fenomeen vanuit verschillende standpunten te modelleren: Multimodelleertechnieken bieden een verzameling aspect modelleertechnieken aan die op de één of andere manier verbonden zijn en samen een coherente modelleertechniek vormen.

Volgens Hommes [04] zijn de verschillende aspectmodellen onderling verbonden doordat ze bepaalde modelleerconcepten en –kennis delen om hetzelfde fenomeen te modelleren. De mate van samenhang en coherentie is daarin een maat is voor de kwaliteit van de multimodelleertechnieken: te weinig samenhang betekent dat de aspecten niet één coherent model opleveren; te veel samenhang betekent dat er een overlap en daarmee redundantie in de modelleertechniek zit. Dit laatste leidt ertoe dat de verschillende aspect–modellen moeilijk onderling consistent gehouden kunnen worden.

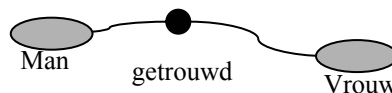
Hommes [04] beweert dus dat om de kwaliteit van multimodelleertechnieken te kunnen beoordelen het van belang is te kijken naar de verschillende aspectmodellen daarin en de mate van coherentie daartussen. Daartoe voldoet klassieke metamodellering volgens hem

niet aangezien daarin deze coherentie niet goed te modelleren valt. Hij stelt een modelleertaal voor waarmee de verschillende aspectmodellen van een multimodelleertechnieken en hun coherentie wél te modelleren zijn: Capture Models for Evaluation (C–Me). Hommes [04] stelt dat met deze modelleertaal wél een goede kwalitatieve beoordeling van verschillende multimodelleertechnieken mogelijk is.

## 2.2 De taal Capture Models for Evaluation (C-Me)

De taal C–Me richt zich zoals gezegd op de concepten die door de modelleertechniek worden aangeboden en hun onderlinge verbanden. Van elk aspect–model dat door de modelleertechniek wordt aangeboden kan een metamodel worden gemaakt. In dit metamodel worden de concepten beschreven die in dát aspectmodel kunnen bestaan. Ook is het mogelijk de verbanden tussen concepten binnen dat aspectmodel aan te geven, en daaraan beperkingregels te koppelen. Daarnaast biedt C–Me de mogelijkheid de verschillende aspectmodelleertechnieken met elkaar te verbinden door te kijken welke concepten in meerdere aspectmodellen voorkomen en welke andere verbanden er bestaan tussen de verschillende aspectmodellen.

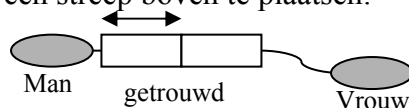
We zullen deze drie mogelijkheden kort bespreken. Het maken van een metamodel van één aspectmodel richt zich als eerste slechts op de daarin voorkomende concepten en hun onderlinge relaties. De concepten worden getekend als bolletjes, en een lijn tussen twee bolletjes geeft vervolgens aan dat daartussen een bepaalde relatie bestaat:



Daarna is het mogelijk de relatie een beperking mee te geven, bijvoorbeeld dat een bepaalde man slechts getrouwd kan zijn met één vrouw. Deze beperkingen kunnen in verschillende notaties gedaan worden, in dit rapport wordt gebruik gemaakt van ORM [03].

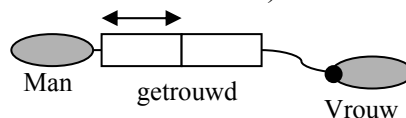
In het kort komt de ORM–notatie op het volgende neer:

Elk concept dat meedoet aan een relatie speelt in die relatie een rol. Aan het spelen van een rol kan als beperking worden opgelegd dat elke instantie van dat concept die rol maar één keer mag spelen, door er een streep boven te plaatsen:



In dit voorbeeld staat genoteerd dat een bepaalde man maar één keer mee mag doen aan de relatie ‘getrouwd’. Oftewel: mannen mogen maar getrouwd zijn met één vrouw, vrouwen mogen zo vaak trouwen als ze willen.

Een ander belangrijk concept is de mogelijkheid om aan te geven dat alle instanties van een bepaalde populatie mee *moeten* doen in een relatie. Dit wordt aangegeven met een klein zwart rondje op de lijn tussen concept en relatie. Hiermee kan bijvoorbeeld aangegeven worden dat alle vrouwen die er zijn, getrouwd zijn (hetgeen uiteraard niet de situatie modelleert zoals die in onze wereld bestaat):



Dan is er nog de mogelijkheid om concepten uit verschillende aspectmodellen met elkaar te verbinden. Dit gebeurt simpelweg door de verschillende metamodellen van de aspectmodellen naast elkaar te tekenen, en tussen de concepten die in verschillende aspectmodellen voorkomen een stippellijn te tekenen. Ook is het mogelijk tussen verschillende concepten uit verschillende aspectmodellen relaties aan te geven. Op deze relaties kunnen vervolgens –indien gewenst– beperkingen worden aangegeven, op dezelfde manier als binnen één aspectmodel (dus bijvoorbeeld met ORM).

### **2.3 Keuze voor C–Me**

De C–Me taal is door ons toegepast bij de beoordeling van Petrinet+. Het voordeel dat C–Me biedt bij de analyse van multimodelleertechnieken speelde in dit specifieke geval geen rol, aangezien Petrinet+ een aspectmodelleertechniek is. Maar aangezien dit onderzoek zich richt op een analyse van Petrinet+ met de bedoeling deze later te kunnen vergelijken met andere modelleertechnieken is het belangrijk een modelleertechniek te kiezen die deze vergelijking mogelijk maakt.

De eerste voorwaarde voor vergelijking is dat de verschillende metamodellen gemodelleerd zijn in dezelfde modelleertaal. Aangezien –in het kader van hetzelfde collegevak– andere modelleertechnieken door andere studenten zijn geanalyseerd met C–Me is onze keuze om Petrinet+ met C–Me te gaan modelleren vooral daarvan afhankelijk. Aangezien de modellen dan in dezelfde taal geschreven zijn, is een vergelijking later goed mogelijk. De tweede reden is dat C–Me goede mogelijkheden biedt om kwantitatieve analyse uit te voeren op de gevonden analyseresultaten. Zo kan er in een later stadium dus zowel op kwantitatieve als op kwalitatieve gronden een vergelijking worden gemaakt tussen verschillende modelleertechnieken.

## 3 Beschrijving van Petrinet+

Dit hoofdstuk geeft een beschrijving van de modelleertechniek Petrinet+. Allereerst wordt het begrip petrinet uiteengezet, welke als uitgangspunt voor Petrinet+ dient. Vervolgens worden de toevoegingen en wijzigingen besproken. Tenslotte worden de generieke concepten en toegestane constructies weergegeven.

### 3.1 Petrinet als uitgangspunt voor Petrinet+

De techniek petrinet, welke nu bekend staat als ‘klassiek petrinet’, is voor het eerst uiteengezet in een proefschrift [05] en is daarna onderwerp tot verdere studie geweest. Momenteel wordt het als volgt gedefinieerd [01]: een petrinet is een gerichte bipartite graaf bestaande uit de elementen *toestanden* en *transities*. De elementen zijn gericht met elkaar verbonden, waarbij verbindingen tussen twee elementen van hetzelfde type niet zijn toegestaan. Toestanden worden aangeduid met een cirkel en transities met een rechthoek.

Als een toestand wordt ingenomen dan bezit deze een teken (Engels: *token*). Een transitie wordt ingegaan op het moment dat alle invoertokens aanwezig zijn. Na de overgang wordt een token geplaatst in de uitvoertoestanden.

Er bestaan diverse uitbreidingen op petrinet. Deze zijn ondermeer [01]:

- Uitbreiding met kleur;
- Uitbreiding met tijd;
- Uitbreiding met hiërarchie.

Gekleurde petrinetten kennen een waarde toe aan de tokens in het model. Traditioneel zijn de tokens betekenisloos, maar nu kunnen ze beschouwd worden als objecten met een bepaalde identiteit (vergelijkbaar met het idee uit *object georiënteerde talen*). Transitie's bepalen de waarden van de uitvoer tokens op basis van de invoer tokens.

Met name voor realtime toepassingen is het van belang dat tijdsduur en vertraging worden opgenomen in het model. De tijdsduur kan op verschillende manieren worden toegevoegd aan het model, en kan gelden voor zowel de toestanden, de transities als de tokens.

Bij grote modellen kan het handig zijn om elementen hiërarchisch te groeperen. Het concept *subnet* biedt een oplossing om de complexiteit van het model te reduceren. Een subnet kan toestanden, transities en andere subnetten bevatten en kan ingepast worden in een petrinet.



## 3.2 Toevoegingen en wijzigingen in Petrinet+

De methode Petrinet+ is gebaseerd op de petrinet techniek, maar kent enkele toevoegingen en wijzigingen. Deze bestaan uit:

- Tokens zijn gekleurd;
- De toevoeging van een uitvoerings- en besturingslaag;
- Opdeling van toestandovergangen op vijf aggregatieniveaus;
- De toevoeging van het concept trigger.

Gekleurde tokens worden besproken in paragraaf 3.1. Mede vanwege de extra waarde die in Petrinet+ aan een token wordt toegekend, wordt hiervoor de term *bestuurd object* gehanteerd. Daarnaast is de naamgeving van bijvoorbeeld labels van toestanden gebonden aan specifieke conventies.

Het onderscheid in uitvoeringslaag en besturingslaag maakt het mogelijk in te grijpen op de toestanden en tokens in het model. Een transitie kan bijvoorbeeld conditioneel worden gestart.

Toestandsovergangen kunnen op vijf verschillende aggregatieniveaus worden gedefinieerd. Deze niveaus kunnen desgewenst door elkaar heen worden gebruikt. Bij geabstraheerde diagrammen bestaat er een relatie tussen een toestandsovergang op twee aggregatieniveaus. De gedefinieerde niveaus geven min of meer structuur aan het concept *hiërarchisch petrinet* (zie paragraaf 3.1).

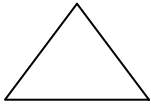
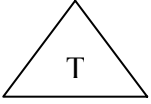



De *trigger* kan verschillende processen opstarten. In tegenstelling tot het token, is de trigger systeemafhankelijk. De informatie die een trigger bevat is dus buiten het systeem gedefinieerd en kan ook verschillende vormen aannemen. Een bijzondere trigger is de tijdstrigger, die afgaat op bijvoorbeeld een vooraf bepaald interval.

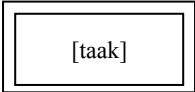
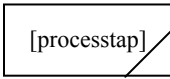
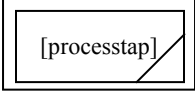
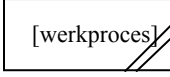



## 3.3 Generieke concepten en mogelijke constructies in Petrinet+

In de vorige paragraaf zijn de diverse uitbreidingen op petrinet weergegeven. De genoemde elementen worden nu in detail besproken in tabel 3-1. Om aan te geven dat een toestandsovergang zich in de besturingslaag bevindt (en dus het document c.q. geval niet transformeert), worden vierkante haken om de naam gezet.

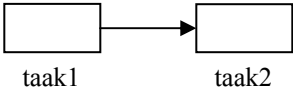
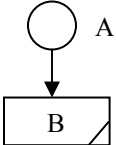
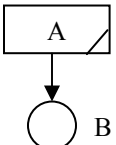
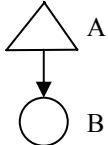
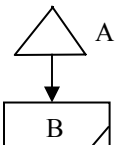
De grafische representatie van de mogelijke constructies in Petrinet+ wordt gegeven in tabel 3-2. Het is onderverdeeld naar aggregatieniveau. Een formele definitie van petrinet is te vinden in [01] en [05].

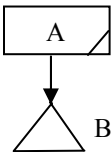
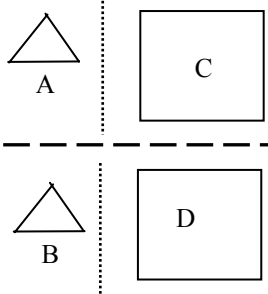
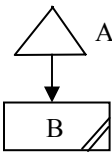
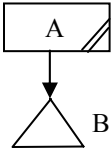
In de bijlage wordt ter illustratie een voorbeeldmodel getoond en uitgelegd. Daarnaast is er per aggregatieniveau een illustratie met korte toelichting voorhanden.

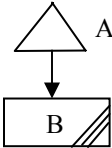
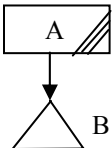
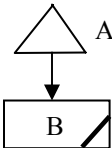
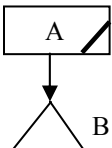
Generiek concept	Notatie	Omschrijving	Opmerkingen
Trigger	 [event]	Is de start van een werkproces, bedrijfsproces of keten; of het is een interrupt in een werkproces die een apart proces opstart om de interrupt af te handelen.	<ul style="list-style-type: none"> <li>•In de top staan de letter A (Analoog) of de letter D (Digitaal) om de aard van binnengekomen document aan te duiden;</li> <li>•In de hoek linksonder staat S (gestructureerd) of O (ongestructureerd) om de aard van de binnengekomen gegevens aan te duiden;</li> <li>•In de hoek rechtsonder staat G (Gevuld) of K (Kaal). Dit geeft aan of het document data bevat of slechts een referentie naar de data;</li> <li>•Als een interrupt plaatsvindt, wordt een ander werkproces opgestart die de interrupt afhandelt aan de hand van de huidige toestand van het originele werkproces.</li> </ul>
Tijdstrigger	 [tijdsduur]	Een speciaal soort trigger die vuurt na een bepaalde periode.	
Multiple trigger	 [event]	Dit is een trigger die een complex object voorstelt. Het is vergelijkbaar met een stapel documenten.	
Toestand	 [object] is [actie]	Worden alleen gebruikt in de beschrijving van een werkproces. Kan aan het begin en/of eind van een werkproces staan en staat altijd tussen twee processtappen.	<ul style="list-style-type: none"> <li>•Als aan het begin van een werkproces een toestand staat direct na de trigger, dan wordt een dossier aangemaakt voor de trigger. Als er geen toestand staat, maar een processtap, dan kan het document verloren gaan als de processtap er niets mee doet;</li> <li>•Als de naam van de toestand in hoofdletters is weergegeven, dan moet de informatie over de toestand worden opgenomen in het management informatie systeem.</li> </ul>
Toestandsovergang: taak	 [taak]	Met behulp van een sequentie van taken wordt een processtap beschreven.	<ul style="list-style-type: none"> <li>•De enige relatie die voorkomt in het taakmodel is de precedentie.</li> </ul>

Toestandsovergang: subtaak		Een subtaak is een verzameling taken.	<ul style="list-style-type: none"> <li>•De subtaak komt voor in hetzelfde model als de taak.</li> </ul>
Toestandsovergang: processtap		Met behulp van processtappen, toestanden en triggers wordt een werkproces beschreven.	<ul style="list-style-type: none"> <li>•Tussen twee processtappen moet altijd een toestand staan;</li> <li>•Als de naam van de processtap tussen vierkante haken staat, vindt het proces in de besturingslaag plaats en transformeert het object niet.</li> </ul>
Toestandsovergang: subprocesstap		Een subprocesstap is een verzameling processtappen en toestanden.	<ul style="list-style-type: none"> <li>•De subprocesstap komt voor in hetzelfde model als de processtap;</li> <li>•Een subprocesstap omvat naast een aantal processtappen ook de tussenliggende toestanden.</li> </ul>
Toestandsovergang: werkproces		Met behulp van werkprocessen en triggers wordt een bedrijfsproces beschreven.	<ul style="list-style-type: none"> <li>•Een werkproces moet altijd beginnen en eindigen met een trigger.</li> </ul>
Toestandsovergang: subwerkproces		Een subwerkproces is een verzameling werkprocessen met de tussenliggende triggers.	<ul style="list-style-type: none"> <li>•Het subwerkproces komt voor in hetzelfde model als het werkproces;</li> <li>•Een subwerkproces omvat naast een aantal werkprocessen ook de tussenliggende triggers.</li> </ul>
Toestandsovergang: bedrijfsproces		Met behulp van bedrijfsprocessen en triggers wordt een keten beschreven.	<ul style="list-style-type: none"> <li>•Een bedrijfsproces begint altijd met een trigger en tussen twee bedrijfsprocessen staat altijd een trigger.</li> </ul>
Toestandsovergang: keten		Een keten start altijd met een trigger.	

tabel 3-1 Beschrijving van generieke concepten uit Petrinet+

Taakniveau: de beschrijving van een processtap			
Constructie	Notatie	Beschrijving	Opmerkingen
Precedentie		Taak2 volgt op taak1.	<ul style="list-style-type: none"> <li>•Dit is de enige mogelijke constructie in de beschrijving van een processtap.</li> </ul>
Processtapniveau: de beschrijving van een werkproces			
Constructie	Notatie	Beschrijving	Opmerkingen
Invoer plaats		De processtap (of subprocesstap) wordt uitgevoerd als alle invoer plaatsen A een token hebben.	<ul style="list-style-type: none"> <li>•Een processtap kan meer dan één invoer plaats hebben;</li> <li>•Alleen de eerste processtap van een werkproces kan een event als invoer hebben. Alle andere processtappen hebben een toestand als invoer plaats.</li> </ul>
Uitvoer plaats		Als processtap (of subprocesstap) a is uitgevoerd, wordt in alle uitvoer plaatsen (B) een token geplaatst.	<ul style="list-style-type: none"> <li>•Een processtap kan meer dan één uitvoer plaats hebben;</li> <li>•Elke processtap of subprocesstap heeft een uitvoer plaats, behalve de laatste processtap van het werkproces. Als op dit werkproces een nieuw werkproces volgt, heeft deze processtap geen toestand, maar een trigger als uitvoer.</li> </ul>
Dossier aanmaken		Als trigger A vuurt, komt er een nieuw token in toestand B, waar een dossier wordt aangemaakt.	<ul style="list-style-type: none"> <li>•Deze constructie wordt gebruikt om ervoor te zorgen dat het token niet verloren gaat. Bij deze constructie wordt gegarandeerd een geval aangemaakt;</li> <li>•De trigger A is het begin van het werkproces.</li> </ul>
Trigger vuurt op processtap		Als trigger A vuurt, kan processtap B starten.	<ul style="list-style-type: none"> <li>•Als processtap B niet direct wordt uitgevoerd als de trigger vuurt, dan gaat deze verloren;</li> <li>•Trigger A kan de start van het werkproces zijn;</li> <li>•Als trigger A niet de start is van het werkproces, dan moet ook een toestand C voorafgaan aan processtap B. Processtap B kan nu uitgevoerd worden als het proces zich in de toestand C bevindt en de trigger A vuurt.</li> </ul>

Processtap vuurt trigger		Processtap A vuurt trigger B.	<ul style="list-style-type: none"> <li>•Het werkproces eindigt met een trigger als er na dit werkproces een ander werkproces wordt opgestart. Anders eindigt het met een toestand, zodat de besturingslaag weet dat het werkproces beëindigd is;</li> <li>•Elke processtap halverwege het werkproces kan een trigger vuren. Deze trigger start een ander proces op in het systeem.</li> </ul>
Interrupt afhandeling		C en D zijn modellen op processtapniveau. C is het reguliers werkproces en D is de interrupt-afhandeling. Trigger A is een interrupt voor werkproces C. Als (gevulde) trigger A vuurt, wordt een dossier aangemaakt en wordt C onderbroken. Dan vuurt (kale) trigger B en wordt D uitgevoerd.	<ul style="list-style-type: none"> <li>•Alle toestanden in C moeten terugkomen in D. Aan de hand van de huidige toestand in C en de trigger die vuurt wordt bepaald hoe de interrupt wordt afgehandeld.</li> </ul>
<b>Werkprocesniveau: de beschrijving van een bedrijfsproces</b>			
<b>Constructie</b>	<b>Notatie</b>	<b>Beschrijving</b>	<b>Opmerkingen</b>
Werkproces starten		Het werkproces B wordt gestart met het vuren van trigger A.	<ul style="list-style-type: none"> <li>•De trigger A komt terug in de beschrijving van het werkproces;</li> <li>•Een model van een bedrijfsproces begint altijd met deze constructie;</li> <li>•Een werkproces wordt alleen maar gestart als er een trigger aan vooraf gaat;</li> <li>•Een trigger kan meerdere werkprocessen starten. Door voorwaarden (die betrekking hebben op de inhoud van de trigger) bij een pijl van trigger A naar werkproces B te schrijven wordt bepaald of de trigger wel of niet werkproces B start;</li> <li>•De constructie geldt ook voor subwerkprocessen.</li> </ul>
Werkproces eindigen		Als werkproces A klaar is, dan wordt trigger B gevuurd.	<ul style="list-style-type: none"> <li>•Deze constructie is altijd aanwezig als er nog een werkproces volgt of als er na dit bedrijfsproces nog een bedrijfsproces plaatsvindt. Is dat niet het geval, dan wordt geen trigger gevuurd;</li> <li>•Een werkproces kan meerdere triggers vuren. Door naamgeving van de triggers (die betrekking hebben op de inhoud van de triggers) wordt bepaald welke daadwerkelijk vuurt;</li> <li>•De constructie geldt ook voor subwerkprocessen.</li> </ul>

Bedrijfsprocesniveau: de beschrijving van een keten			
Constructie	Notatie	Beschrijving	Opmerkingen
Bedrijfsproces starten		Het bedrijfsproces B wordt gestart met het vuren van trigger A.	<ul style="list-style-type: none"> <li>•De trigger A komt terug in de beschrijving van het bedrijfsproces;</li> <li>•Een model van een keten begint altijd met deze constructie;</li> <li>•Een bedrijfsproces wordt alleen maar gestart als er een trigger aan vooraf gaat;</li> <li>•Een trigger kan meerdere bedrijfsprocessen starten. Door voorwaarden (die betrekking hebben op de inhoud van de trigger) bij een pijl van trigger A naar bedrijfsproces B te schrijven wordt bepaald of de trigger wel of niet bedrijfsproces B start.</li> </ul>
bedrijfsproces eindigen		Als bedrijfsproces A klaar is, dan wordt trigger B gevuurd.	<ul style="list-style-type: none"> <li>•Deze constructie is altijd aanwezig als er nog een bedrijfsproces volgt of als er na deze keten nog een keten wordt uitgevoerd. Is dat niet het geval, dan wordt geen trigger gevuurd;</li> <li>•Een bedrijfsproces kan meerdere triggers vuren. Door naamgeving van de triggers (die betrekking hebben op de inhoud van de triggers) wordt bepaald welke daadwerkelijk vuurt.</li> </ul>
Ketenniveau			
Constructie	Notatie	Beschrijving	Opmerkingen
Keten starten		De keten B wordt gestart met het vuren van trigger A.	<ul style="list-style-type: none"> <li>•De trigger A komt terug in de beschrijving van de keten;</li> <li>•Een model op ketenniveau begint altijd met deze constructie;</li> <li>•Een keten wordt alleen maar gestart als er een trigger aan vooraf gaat;</li> <li>•Een trigger kan meerdere ketens starten. Door voorwaarden (die betrekking hebben op de inhoud van de trigger) bij een pijl van trigger A naar keten B te schrijven wordt bepaald of de trigger wel of niet keten B start.</li> </ul>
Keten eindigen		Als keten A klaar is, dan wordt trigger B gevuurd.	<ul style="list-style-type: none"> <li>•Deze constructie is altijd aanwezig als er nog een keten volgt. Is dat niet het geval, dan wordt geen trigger gevuurd;</li> <li>•Een keten kan meerdere triggers vuren. Door naamgeving van de triggers (die betrekking hebben op de inhoud van de triggers) wordt bepaald welke daadwerkelijk vuurt.</li> </ul>

tabel 3-2 Beschrijving van constructies per aggregatieniveau in Petrinet+

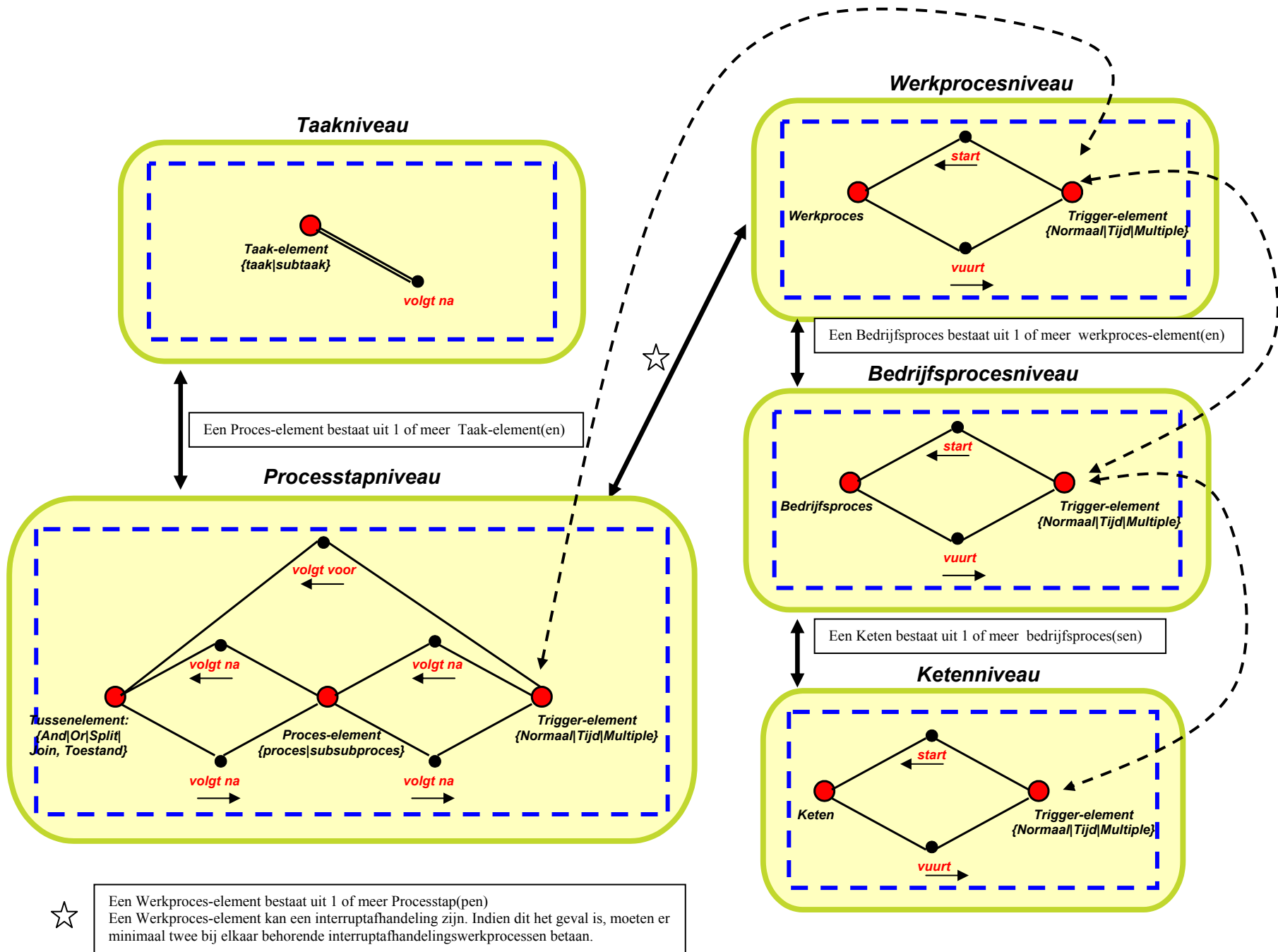
## 4 Metamodel beschrijving van Petrinet+

De metamodelen die gemaakt zijn met behulp van de C–Me techniek resulteren in een overzicht van de gebruikte elementen uit de gemodelleerde taal. In dit overzicht is de structuur te zien van de elementen en hoe deze zich onderling verhouden. Deze onderlinge verhoudingen bestaan in de eerste plaats simpelweg uit de relaties die mogelijkwijs tussen de elementen kunnen worden aangebracht. Ten aanzien van deze relaties zijn beperkingen te leggen, zogenaamde constraints. Een voorbeeld hier van is dat in de relatie tussen man en vrouw, genaamd ‘huwelijk’, elke individuele man/ vrouw slechts één keer mee mag doen. Simpel gezegd: een man mag maar met één vrouw trouwen en vice versa.

We hebben twee diagrammen gemaakt met daarin het door ons gevonden metamodel van Petrinet+. Het eerste diagram (zie figuur 4-1) bestaat uit de gevonden concepten en hun onderlinge relaties. Hierin zijn de concepten van Petrinet+ overzichtelijk samengevat hetgeen een basis vormt voor vergelijking met andere metamodelen die met behulp van C–Me zijn beschreven. Het tweede diagram (zie figuur 4-2 tot en met figuur 4-7) gaat in op de eerder genoemde beperkingen voor individuele objecten ten aanzien van het meespelen in bepaalde relaties. Dit diagram is geschreven in ORM, zie daarvoor § 2.2 en [03].

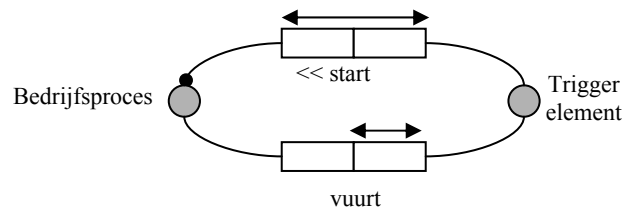
In het eerste diagram is te zien dat Petrinet+ is verdeeld in vijf aggregatie niveaus. Elk niveau is apart gemodelleerd, dit vanwege het overzicht en de verschillende regels die voor elk niveau gelden. Dit levert dus vijf modellen op, te weten: Taakniveau, Processtapniveau, Bedrijfsprocesniveau, Werkprocesniveau en Ketenniveau. Omdat deze modellen tot één geheel behoren, hebben zij elementen met elkaar gemeen. Hoofdzakelijk zijn dit de triggers die op meerdere niveaus onderscheiden worden. Deze eigenschap wordt in het model aangegeven met een gestippelde pijl. Tevens bestaan de hogere aggregatieniveaus uit samengestelde elementen die op de lagere niveaus uitgesplitst zijn. Een voorbeeld hiervan zijn de proceselementen uit het processtapniveau; deze bestaan weer uit taken die op taakniveau gedetailleerder gespecificeerd zijn.

Het tweede diagram is een formele weergave van hetgeen reeds in § 3.3 is beschreven. Dit is een uitbreiding op het eerste diagram. Alle concepten en relaties die in het eerste diagram zijn besproken komen hierin terug, waarbij voor elke relatie is aangegeven of de rol verplicht is voor de gehele populatie en of een rol uniek is voor de populatie.

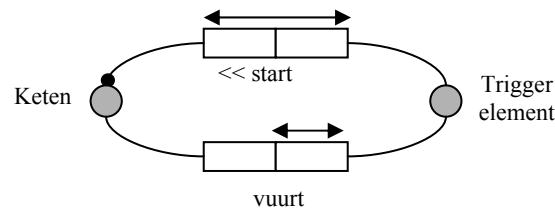


figuur 4-1 C-Me metamodel Petri net+

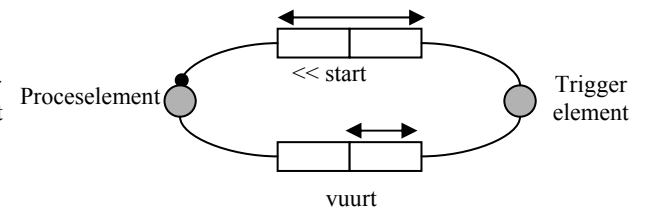




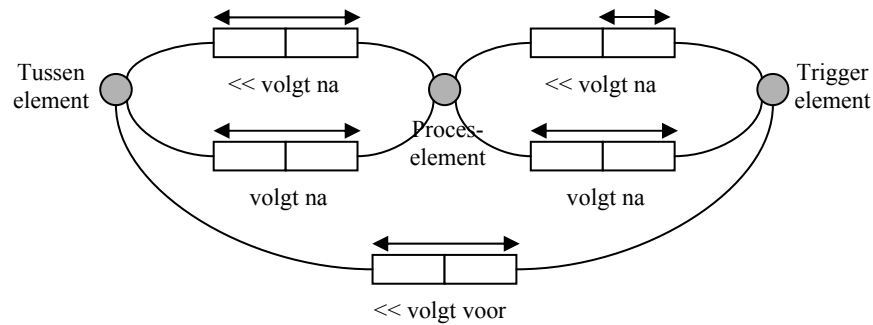
figuur 4-3 ORM model bedrijfsprocesniveau



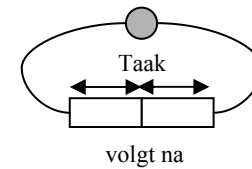
figuur 4-4 ORM model ketenniveau



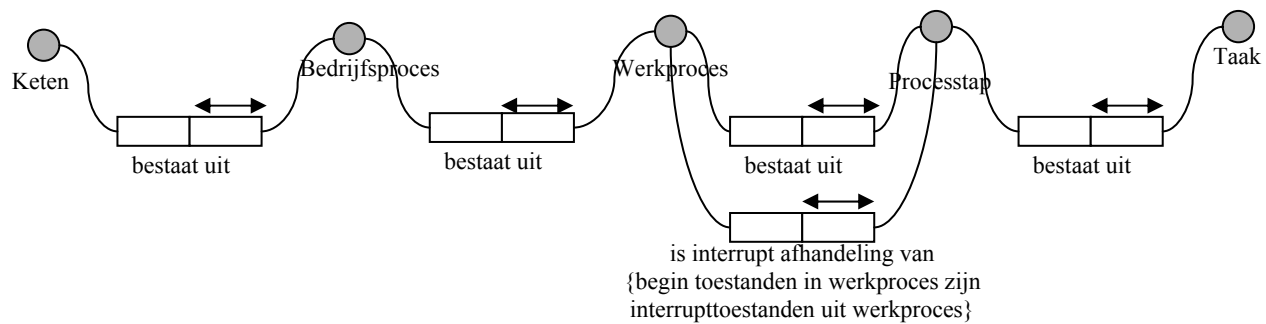
figuur 4-2 ORM model werkprocesniveau



figuur 4-6 ORM model processtapniveau



figuur 4-7 ORM model taakniveau



figuur 4-5 ORM model intramodelniveau afhankelijkheden

## 5 Kwalitatieve analyse Petrinet+

Petrinet+ is een aspectmodelleertechniek. Dat wil zeggen dat er één aspect van de bedrijfsactiviteiten wordt gemodelleerd. In Petrinet+ wordt de uitvoering van de bedrijfsprocessen gemodelleerd. Daarnaast wordt nog een uitstapje gemaakt (door middel van de aggregatieniveaus) naar de control van de bedrijfsprocessen.

In dit hoofdstuk worden eerst de belangrijkste voor- en nadelen van Petrinet+ bekeken; vervolgens worden een paar suggesties gedaan voor verbetering van de techniek. Tenslotte wordt aangegeven hoe een eventuele koppeling met een datamodel gerealiseerd zou kunnen worden.

### 5.1 Voordelen van Petrinet+

Petrinet+ is gebaseerd op de klassieke petrinet [01] techniek, zoals beschreven in hoofdstuk 3. Dit brengt een aantal voordelen met zich mee. Allereerst heeft de petrinet techniek zich al bewezen in de praktijk. Verder zijn er al veel mensen bekend met petrinetten, waardoor men sneller zal kunnen wennen aan Petrinet+.

De concepten die aan petrinet zijn toegevoegd in Petrinet+ zorgen voor een grotere uitdrukingskracht van het model. Het gekleurde token (dat overigens al een bekend concept is in het gekleurde petrinet<sup>1</sup>) brengt goed tot uitdrukking dat het bestuurd object gegevens bevat. De communicatie tussen de verschillende werkprocessen, bedrijfsprocessen en ketens en de communicatie met de actoren buiten het systeem worden gemodelleerd met triggers, een belangrijke uitbreiding.

De aggregatieniveaus (in feite in dit concept afkomstig van de hiërarchische petrinetten<sup>2</sup>) in Petrinet+ zijn een belangrijk onderdeel van de modelleertechniek. Dit concept zorgt ervoor dat de scope van de techniek aanzienlijk wordt verbreed. In veel multi-modelleertechnieken (zoals DEM<sup>3</sup>) zit een aparte aspect modelleertechniek die de koppeling verzorgt tussen de verschillende processen. In Petrinet+ wordt dit probleem al opgelost met behulp van de aggregatieniveaus, waardoor men één aspect-modelleertechniek minder nodig heeft, zodat semantische controle van het uiteindelijke model eenvoudiger zal zijn.

### 5.2 Nadelen van Petrinet+

Na een formele en een gebruiksevaluatie van de methode Petrinet+ komen een aantal negatieve aspecten naar voren:

---

<sup>1</sup> Zie <http://www.daimi.aau.dk/CPnets/intro/>

<sup>2</sup> Zie <http://www.ee.umanitoba.ca/tech.archive/pnex5.html>

<sup>3</sup> Zie <http://iew3.technion.ac.il/Labs/Dynam/>

- Er is geen duidelijke systeemaafbakening;
- Er is geen formele beschrijving van de aggregatieniveaus;
- Er is geen duidelijke werkwijze om tot een model te komen;
- Syntactische verificatie wordt bemoeilijkt door complexiteit van het metamodel;
- Er ontbreken actieregels bij processtappen of taken;
- De aggregatieniveaus brengen complexiteit met zich mee.

Petrinet+ kent geen duidelijke systeemaafbakening. Hierdoor is niet te achterhalen wat wel en wat niet binnen het gemodelleerde systeem valt. Conceptueel gezien lijkt het waarschijnlijk dat de toestanden en toestandsovergangen (waaronder o.a. processtappen en werkprocessen) binnen het systeem vallen, en triggers daarbuiten. Triggers kunnen echter wel intern worden opgenomen, zoals bij de start van een nieuw werkproces. Deze logische scheiding wordt in Petrinet+ dan ook niet consequent doorgevoerd.

Er is niet formeel is vastgelegd op welk aggregatieniveau een bepaalde activiteit zich bevindt. Er zit geen denkmodel of theorie achter de verschillende aggregatieniveaus. Het is dus niet duidelijk of iets een taak, processtap, werkproces, etc. is. Wellicht levert dit binnen het GAK minder problemen op, omdat deze aggregatieniveaus binnen de bedrijfsvoering bekend zijn. Voor toepassing in een ander bedrijf, of uitvoering door een niet ingewijd persoon, is dit echter minder goed bruikbaar.

Er is geen duidelijke werkwijze voorhanden om tot een correct Petrinet+ model te komen. Zo kan er bijvoorbeeld worden gekozen voor een top-down benadering (waarbij wordt begonnen op ketenniveau), of voor een bottom-up benadering (te beginnen op taakniveau). Ook is niet duidelijk of aggregatieniveaus mogen worden weggelaten. Hierdoor lijkt het uiteindelijke model sterk persoonsgebonden en is het moeilijk een objectieve uitspraak te doen over de kwaliteit van het model.

Vanwege de eenvoud van het metamodel van petrinetten, is het vrij gemakkelijk na te gaan of een petrinet model syntactisch correct is. Petrinet+ heeft echter een veel ingewikkelder metamodel (zie hoofdstuk 4), dat maakt deze controle moeilijker. Ook semantisch gezien is het vanwege de grote hoeveelheid concepten en relaties daar tussen lastig na te gaan of een model volledig is.

Transformaties op het niveau van processtap of taak worden niet inhoudelijk beschreven. Zogenaamde actieregels, die concreet aangeven wat voor bewerkingen er worden uitgevoerd op de data, zijn onontbeerlijk voor het mogelijk automatiseren van stappen of taken. Het ontbreken van actieregels bemoeilijkt de overgang naar een ontwerp voor een informatiesysteem.

Het bestaan van de verschillende aggregatieniveaus elimineert de noodzaak van het hebben van een aspect-modelleertechniek die de koppeling tussen processen weergeeft. Echter, Petrinet+ is minder krachtig dan deze techniek zou kunnen zijn. Concepten als systeemgrens en actoren zijn niet aanwezig in Petrinet+, maar zijn wel terug te vinden in veel modelleertechnieken die de koppeling tussen verschillende processen in bedrijven beschrijven. Als Petrinet+ wordt gekoppeld aan zo'n techniek zal er heel veel overlap in

de modellen zijn, zodat consistentie moeilijk te behouden is. Als Petrinet+ niet wordt gekoppeld aan zo'n techniek, kan er veel niet worden uitgedrukt in het model, wat ook niet wenselijk is.

### 5.3 Suggesties voor verbetering

Petrinet+ probeert erg veel concepten in zich te verenigen. Ook concepten die oorspronkelijk tot het domein van andere aspectmodelleertechnieken behoren. Toch worden die andere aspecten van de bedrijfsvoering niet uitvoerig genoeg beschreven in Petrinet+ om die andere techniek overbodig te maken. Petrinet+ is niet gebaseerd op een theorie van organisaties.

Er zijn nu twee mogelijkheden. Ten eerste kunnen alle aspecten die niet uitvoerig genoeg worden beschreven worden toegevoegd aan Petrinet+. De tweede mogelijkheid is om ze te verwijderen uit Petrinet+ (het beperken van de scope) en gebruik te maken van andere aspectmodelleertechnieken om deze te modelleren. De eerste optie zou de nu al complexe techniek een stuk complexer maken. Als men kiest voor de tweede optie, worden de Petrinet+ modellen een stuk eenvoudiger en kunnen alle aspecten van de bedrijfsvoering toch worden belicht (in andere modellen).

Om tot een goed model van de bedrijfsprocessen te komen, moeten alle aspecten worden belicht in een multimodelleertechniek. Petrinet+ kan dit niet alleen (zoals hierboven aangegeven). Het is van belang dat de verschillende aspectmodelleertechnieken op elkaar aansluiten en weinig overlap hebben (om inconsistentie en overmatige complexiteit te voorkomen). Er moet nauw worden samengewerkt met de groepen die bezig zijn met de verschillende aspecten. Er moeten duidelijke afspraken worden gemaakt over welke aspecten in welke techniek worden belicht en welke concepten terugkomen in de verschillende technieken om de koppeling te realiseren.

Aan de hand van het metamodel uit hoofdstuk 4 kunnen we concluderen dat de modellen op werkproces-, bedrijfsproces- en ketenniveau syntactisch gezien praktisch hetzelfde zijn. Hier rijst de vraag of het wel van belang is om onderscheid te maken tussen al deze niveaus. Het zal moeten worden uitgezocht of deze niveaus wel een toegevoegd waarde hebben in deze modelleertechniek.

Wat we ook kunnen concluderen uit het metamodel is dat er veel concepten en relaties tussen concepten aanwezig zijn in Petrinet+ op het processtapniveau. Dit duidt op een complexe techniek, waar mogelijk veel fouten mee worden gemaakt. Hier kunnen echter nog geen verdere conclusies aan worden verbonden, aangezien we geen kwantitatieve vergelijking met andere modelleertechnieken of gebruikstesten hebben gedaan.

Voor de koppeling van de verschillende aspectmodelleertechnieken is het van belang dat de concepten in de technieken formeel worden gedefinieerd. Men moet weten wat een processtap is, wat een werkproces is, wat een trigger is, of een trigger intern dan wel extern is, waar de systeemgrens ligt, etc.

Er zal voor de uiteindelijke multimodelleertechniek een werkwijze moeten komen die modellers precies vertelt hoe ze een model moeten opbouwen. Dit voorkomt semantische fouten.

## 5.4 Koppeling aan een data model

Binnen Petrinet+ wordt met behulp van de triggers wel iets gezegd over de vorm van de data, maar er wordt niets gezegd over de inhoud. Er wordt bijvoorbeeld wel gezegd of data gestructureerd, dan wel ongestructureerd is, maar er wordt niet gezegd wat de data is (wat de entiteiten zijn). Hierdoor is het moeilijk om een data model te koppelen aan een Petrinet+ model.

Voordat een koppeling met een data model mogelijk is, moeten allereerst de verschillende aggregatieniveaus formeel worden gedefinieerd. Dan zal één van de aggregatieniveaus (het beste is het processtapniveau, omdat hier de toestanden worden beschreven) moeten worden gekozen om het data model aan te koppelen.

De koppeling tussen het data model en het Petrinet+ model kan op twee manieren worden gerealiseerd. De naam van de data entiteit komt terug in de namen van de toestandsovergangen en de toestanden of er wordt een element toegevoegd aan het Petrinet+ model dat de data entiteit voorstelt en dat wordt gekoppeld aan de processen om aan te geven welke processen deze data raadplegen.

De volgende stap is de koppeling aan een data modelleertechniek. De juiste modelleertechniek zal moeten worden gekozen aan de hand van de aanwezige kennis in de organisatie en de eigenschappen van de data.

## 6 Conclusie

In dit rapport is middels een metamodellering analyse een aanzet gegeven naar een kwantitatieve en kwalitatieve vergelijking van Petrinet+ met andere technieken. Deze analyse bestond uit een formalisering van de technische specificaties van de Petrinet+ modelleertechniek, het maken van metamodellen middels de methode C–Me inclusief beperkingregels met behulp van ORM.

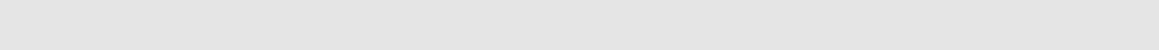
Bij het formaliseren van de gegeven specificaties bleek dat veel zaken onduidelijk zijn. Zo is de te volgen werkwijze niet gegeven, hierdoor is het onduidelijk wat de precieze aanpak moet zijn bij het construeren van een Petrinet+ model (vanuit welk opzicht: bottom–up of top–down). Triggers moeten gedetailleerder gespecificeerd worden, anders is het onduidelijk op welk niveau de trigger behoort en of het een interne of externe trigger is. De systeemgrens is niet direct te bepalen uit de specificaties, waardoor er snel fouten kunnen ontstaan in de interpretatie en het construeren van de modellen. Tevens ontbreken er actieregels, waardoor het vinden van actoren en de link naar een mogelijk datamodel bemoeilijkt wordt. Tenslotte zal een syntactische controle niet eenvoudig zijn, door het ontbreken van eenduidigheid in de regels en de complexiteit van het geheel.

Na het construeren van het metamodel valt te concluderen dat verschillende aggregatieniveaus veel overeenkomsten bezitten, waardoor een heroverweging raadzaam is van de gebruikte vijf niveaus. Tevens zijn de aggregatieniveaus niet duidelijk gespecificeerd, een voorbeeld hierbij is de vraag of een processtap (noodzakelijkerwijs) bestaat uit één of meer taken en wanneer er bij het modelleren een processtap of juist een werkproces gebruikt moet worden. Door gebruik te maken van aggregatie begeeft Petrinet+ begeeft zich ten dele op het gebied van andere modelleertechnieken, maar is daarbij te beperkt om deze te vervangen. De onduidelijkheden hebben onder meer tot gevolg dat een koppeling met een mogelijk datamodel (nog) niet mogelijk is.

Er moet echter niet worden vergeten dat Petrinet+ weldegelijk ook goede punten heeft. Zo is de keuze voor gekleurde petrinetten een goede manier om extra functionaliteit te bieden ten opzichte van het klassieke Petrinet. Het gebruiken van triggers is eveneens een goede aanvulling, er kan hierdoor meer gemodelleerd worden dan voorheen. Tenslotte kan het concept van de aggregatieniveaus, indien beter gespecificeerd, een aansluiting geven naar een datamodel en de overzichtelijkheid bevorderen bij het beschouwen van het totale model op ketenniveau.

Naar aanleiding van de boven besproken bevindingen raden wij het gebruik van Petrinet+ (in zijn huidige staat) af, aangezien er te veel onduidelijkheden zijn hetgeen fouten in de hand kan werken. Hierbij gaan wij uit van het gegeven dat bestaande modelleertechnieken zich reeds jaren bewezen hebben, waardoor een nieuwe techniek een waardevolle (unieke) eigenschap moet bezitten om de status ‘aanrader’ te krijgen. Deze unieke eigenschap zien wij in de huidige vorm van Petrinet+ enigszins, deze is echter te beperkt en te onduidelijk gespecificeerd.

Tenslotte willen wij opmerken dat de Petrinet+ modelleertechniek in de context waarin het ontwikkeld is (GAK), weldegelijk een waardevolle aanvulling kan zijn indien het beter en nauwkeuriger gespecificeerd wordt. Het feit de techniek is ontstaan middels best-practice geeft aan dat er, in de praktijk, een behoefte bestaat voor een dergelijke methode. Wij raden daarbij aan de ogen niet te sluiten voor bestaande modelleertechnieken die zich al bewezen hebben.



## Referenties

- [01] AALST, W. VAN DE, “Petri nets”, [On line] Zie: <http://www.wis.win.tue.nl/~wsinwa/jcsc/node3.html>, 1997.
- [02] GROOT, R. DE, E.A., “Petrinet+. Procesmodelleren volgens de Petrinet+ modelleertechniek.”, Draft Versie 1.0, UWV Gak, 2002.
- [03] HALPIN, T., “Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design”, The Morgan Kaufmann, 2001.
- [04] HOMMES, B., “Evaluating Conceptual Coherence in Multi Modeling Techniques”, 2002.
- [05] PETRI, C.A., “Kommunikation mit Automaten”, Proefschrift, Institut für instrumentelle Mathematik, Bonn, 1962.

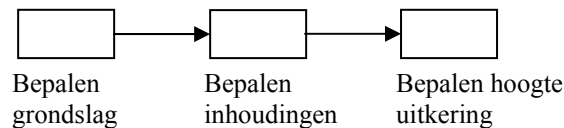


## Bijlage – Voorbeelduitwerking Petrinet+

Deze bijlage bespreekt de uitwerking van een Petrinet+ model. Allereerst is per aggregatieniveau een voorbeeld geconstrueerd. Dit wordt gevolgd door een geïntegreerd model, waaruit enkele constructies in verder detail worden besproken.

### Aggregatieniveau taak

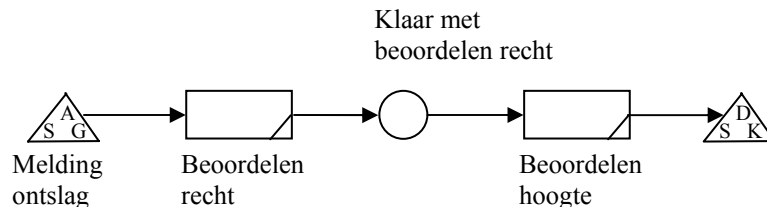
De taak wordt in de modellen gebruikt voor het meest gedetailleerde niveau. Een aantal taken bij elkaar vormen een processtap. Tussen de taken bevinden zich verbindingen. In de naam van de taak wordt de handeling genoemd. In figuur 6-1 wordt de processtap "Bepalen hoogte" uitgewerkt middels drie taken. In dit voorbeeld hebben de taken binnen de processtap een bepaalde volgorde, dit is echter niet altijd zo.



figuur 6-1 Aggregatieniveau taak

### Aggregatieniveau processtap

De processtap wordt gebruikt voor het beschrijven afgeronde gehelen in het model. Na de processtap bevindt zich een herkenbare status, welke wordt weergegeven door een toestand. Dit resulteert in het schema in processtappen, triggers en toestanden die om en om met elkaar verbonden worden. Hieronder wordt de processtap met een voorbeeld beschreven.

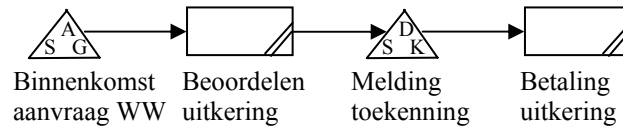


figuur 6-2 Aggregatieniveau processtap

In dit voorbeeld wordt de processtap "Beoordelen recht" geïnitieerd door de trigger "Melding ontslag". Na het afronden van de processtap "Beoordelen recht" komt het bestuurd object in de toestand "Klaar met beoordelen recht" terecht. De besturingsprocessen zien door de plaatsing van het bestuurd object in de toestand "Klaar met beoordelen recht" dat de processtap "Beoordelen recht" is afgerond. Vanuit deze toestand kan het bestuurd object door de processtap "Beoordelen hoogte" worden opgepakt. Het product van processtap is bedoeld als invoer voor andere werkprocessen. De gereed melding van het product wordt door middel van een trigger aan de besturingslaag gestuurd. De besturingslaag zorgt voor een verdere afhandeling van de trigger.

### Aggregatieniveau werkproces

Een werkproces wordt gebruikt voor een verzameling processtappen die gezamenlijk een product voortbrengen.

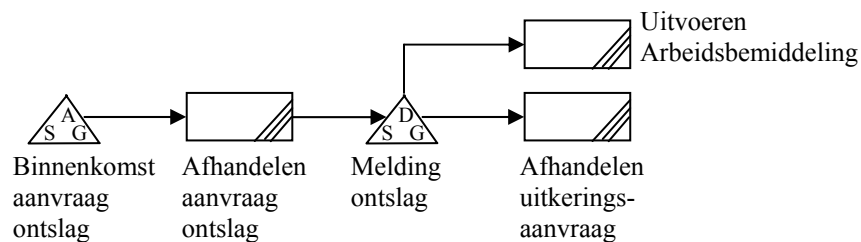


figuur 6-3 Aggregatieniveau werkproces

In dit voorbeeld worden de processtappen "Beoordelen recht" en "Beoordelen hoogte" samengevoegd tot het werkproces "Beoordelen uitkering". Het proces "Toekennen uitkering" levert als resultaat een trigger die bijvoorbeeld het proces "Uitbetalen uitkering" opstart. Het proces "Uitbetalen uitkering" is als afzonderlijk werkproces opgenomen, omdat het werkproces "Uitbetalen uitkering" door een aantal andere processen getriggerd kan worden en een afzonderlijk product oplevert.

### Aggregatieniveau bedrijfsproces

Een aantal werkprocessen vormen bij elkaar een bedrijfsproces. Een bedrijfsproces wordt gebruikt voor een proces dat zich binnen een bedrijf afspeelt. De bedrijfsprocessen volgen elkaar op en vormen samen een keten. De bedrijfsprocessen worden verbonden middels triggers.

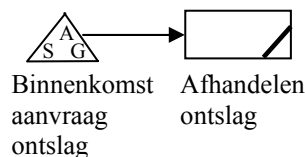


figuur 6-4 Aggregatieniveau bedrijfsproces

Bij de rechtbank wordt beoordeeld of de trigger "Aanvraag ontslag" voldoende aanleiding bevat om werknemer te ontslaan. Als dit het geval is dat wordt dit door de rechtbank in de vorm van de trigger "Melding ontslag". De trigger is de reden tot starten voor de bedrijfsprocessen "Zorgen voor arbeidsbemiddeling" en "Verstrekken uitkering". Het bedrijfsproces "Vertrekken uitkering" is het totaal van alle werkprocessen die bij de UWV plaats vinden. Voorbeelden van deze werkprocessen zijn "Beoordelen uitkering" en "Uitbetalen uitkering".

### Aggregatieniveau keten

De keten omvat het gehele proces vanaf de grondstoffen tot de oplevering van het eindproduct. Hieronder wordt de keten door middel van een voorbeeld beschreven.

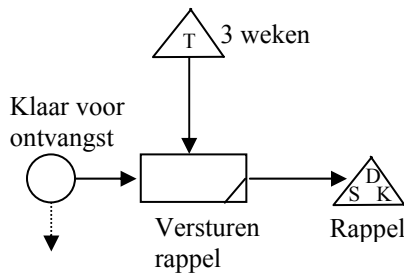


figuur 6-5 Aggregatieniveau keten

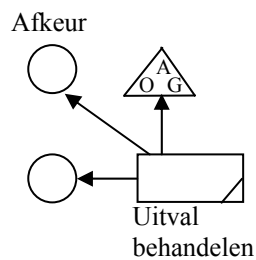
Na aanvraag van het ontslag van een werknemer zullen verschillende instanties op de hoogte worden gesteld. Het op de hoogte stellen is in dit voorbeeld weergegeven als een driehoek, de trigger. Het totaal van alle betrokken bedrijven, bijvoorbeeld de UWV en de arbeidsbureaus, is weergegeven door de rechthoek met de dikgetrokken streep.

### Voorbeeld Uitvalafhandeling

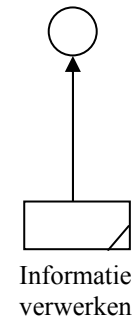
In figuur 6-9 wordt Petrinet+ door middel van een voorbeeld beschreven. In dit voorbeeld wordt periodiek gekeken naar uitval. Bij uitval kan worden gedacht aan werkgevers waarvoor geen afrekening is aangemaakt, omdat jaaropgaven voor bepaalde medewerkers niet zijn ontvangen. Uit het totaal van de werkgevers zullen deze werkgevers worden geselecteerd. Aangezien alle uitgevallen werkgevers in één keer aan het volgende proces worden aangeboden, wordt dit in het schema weergegeven als multiple trigger. Hieronder worden een paar belangrijke elementen beschreven die in het voorbeeld voorkomen.



figuur 6-7 Trigger/or-join



figuur 6-8 Uitval behandelen



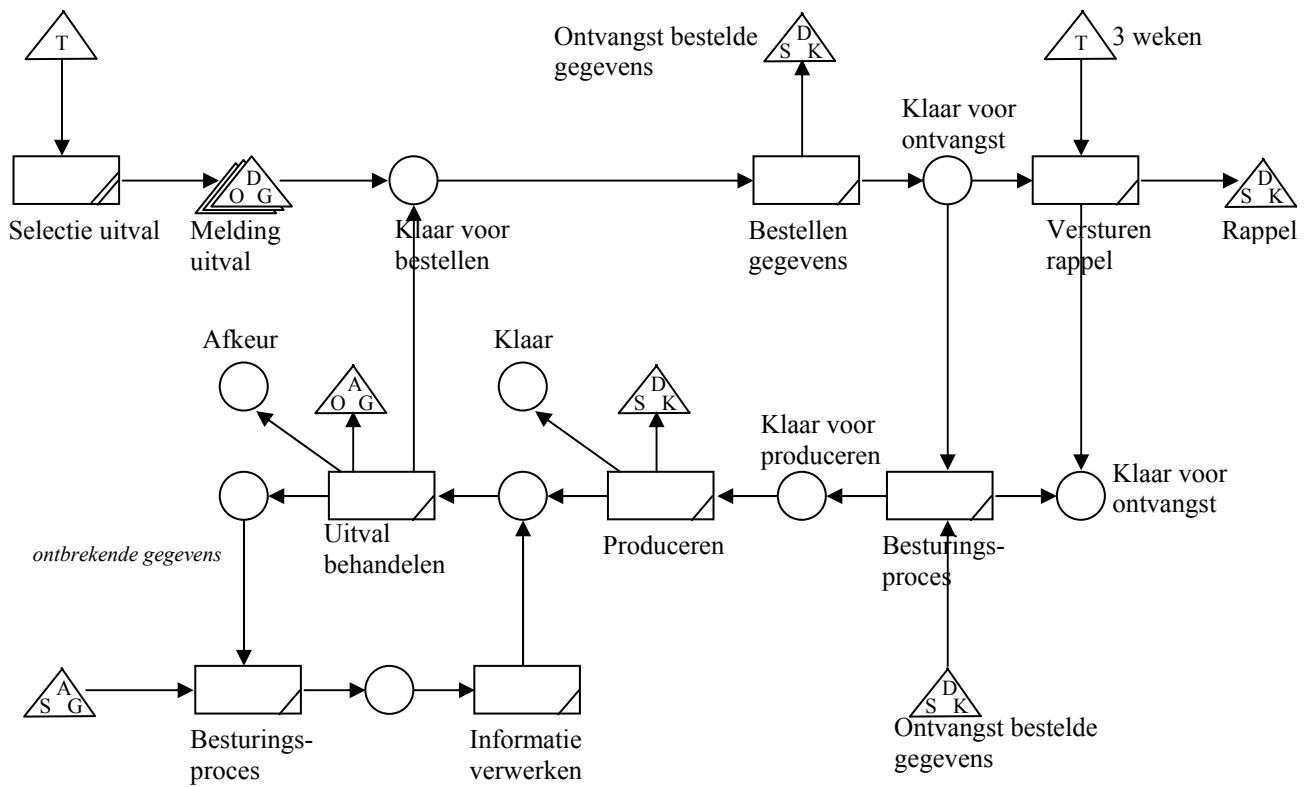
figuur 6-6 Informatie verwerken

De trigger in figuur 6-7 is een signaal voor het bestellen van de gegevens bij de centrale gegevensverzameling. Als de gegevens niet binnen een bepaalde termijn worden geleverd of als de gegevens niet volledig zijn, dan wordt een ander signaal verstuurd door een van de andere triggers.

In figuur 6-8 is er sprake van een OR-join. Nadat de gegevens binnen zijn gekomen worden twee andere processtappen uitgevoerd. Op die manier wordt het bestellen van de gegevens aan de besturingslaag gemeld en wordt ook op gelijk tijdstip gecontroleerd of deze binnen een bepaalde termijn binnen zijn.

Bij de uitval wordt er gekeken welke gegevens ontbreken. Deze gegevens worden bij de klant opgevraagd. Het opvragen van de gegevens wordt aan de besturingslaag gemeld zodat de besturingslaag na het binnen komen van de gegevens weet wie op de gegevens wacht.

De binnengekomen informatie wordt bijgewerkt in figuur 6-6. Bij onvoldoende gegevens wordt het uitvalproces nogmaals doorlopen. Bij voldoende gegevens kan het proces produceren worden gestart.



figuur 6-9 Voorbeelduitwerking Petrinet+