

Petrinet⁺, een klassieker in een nieuw jasje

*Bouwbare modellen
voor het hele bedrijf*

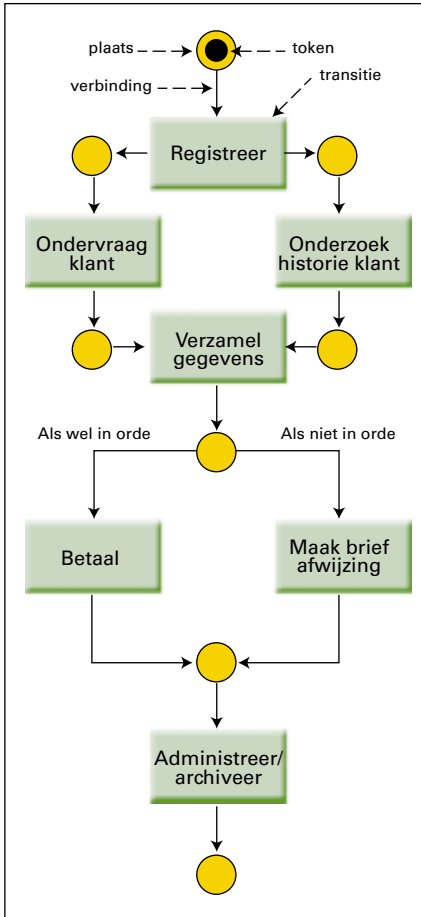




Illustratie: Leon van Leeuwen

"Als je niet eerst de processen van een bedrijf goed in kaart brengt, dan zijn alle andere activiteiten om veranderingen in het bedrijf door te voeren gedoemd te mislukken". Een veelgehoorde uitspraak, alleen handelt niemand ernaar. Met als gevolg dat als de automatiseringsafdeling al klaar is, het bedrijf nog bezig is haar processen in kaart te brengen. Een ISO-traject achteraf moet dan proberen de zaken nog wat op elkaar aan te laten sluiten. Binnen de nieuwe uitvoeringsinstelling UWV, ontstaan door een fusie van LISV, Gak, SFB-bouw, USZO, Guo en Cadans, maakt men beleid om dit tij te keren. Onder meer door het opzetten van een integrale referentie-architectuur voor bedrijfsprocessen en ICT en het opstellen van strikte conventies voor procesmodellering. Petrinet-modellen in een nieuw jasje vormen hierbij het uitgangspunt voor procesmodellering.

*Hans Tönissen, Willem Feijen, Ronald de Groot,
Egon Willemsz en Mark Backer*



Afbeelding 1. Petrinet-model voor een schade-toekenning.

Petrinet-modellen zijn formele en toch overzichtelijke modellen. Dat is belangrijk, want procesmodellering levert 'praatplaatjes' doordat de modellen duidelijk en transparant zijn. Formele modellen zijn bovendien belangrijk omdat ze een wiskundige basis bieden voor simulatie (hoe gedraagt het gemodelleerde proces zich onder bepaalde omstandigheden) en animatie van de modellen. Afbeelding 1 toont een simpel voorbeeld van een Petrinet-model.

Gemis klassieke modellen

Aan de klassieke Petrinet-modellen ontbreekt echter iets. Dit is al lang geleden onderkend. De klassieke Petrinet-modellen zijn na deze constatering verrijkt met zogenaamde hoogniveau-uitbreidingen kleur, tijd en hiërarchie. Hiërarchie is toege-

voegd voor een meer overzichtelijke structuur. Structuur is natuurlijk een sleutelbegrip. Structuur maakt complexiteit hanteerbaar. Structuur in een proces aanbrengen betekent dat je de samenhang *binnen* en de samenhang *tussen* de processen beschrijft (respectievelijk de modulaire en hiërarchische structuur!). Als de hiërarchische niveaus echter abstract blijven en een bepaalde betekenis (semantiek) ontberen, dan communiceert dat moeilijk.

Het niveau van de hiërarchie is allesbepalend voor de doelgroep (manager, workflow-engineer, systeemontwerper, gebruiker) die men probeert aan te spreken met de praatplaatjes. In de praktijk zie je zelfs vaak dat men voor de verschillende doelgroepen geheel eigen plaatjes maakt (Powerpoint-animaties, matrices, stroomschema's, enzovoort). De kans op misinterpretaties van het ene model naar het andere is dan groot. Vanuit één model moet het mogelijk zijn meerdere doelgroepen te bedienen.

Om interacties aan te geven tussen processen en zeker tussen processen van verschillende bedrijfsfuncties is het ook erg handig om de communicatie tussen de processen (zowel hiërarchisch als modulair) vanuit een bepaalde semantiek te omschrijven. De klassieke Petrinet-modellen kennen hier maar één manier voor: de *verbinding*. Zolang men maar één specifiek en geïsoleerd proces in kaart hoeft te brengen lukt dat nog wel met de klassieke Petrinet-modellen, maar het vervolg is ook bekend. Uiteindelijk maakt men voor 'elk probleem een eigen systeem'. Later blijken het echter helemaal geen geïsoleerde processen te zijn. Verkopers van integratiesoftware varen hier al jaren wel bij.

Een ander gemis van de klassieke Petrinet-modellen is de mogelijkheid om samenhang met andere bedrijfs-

aspecten te borgen, zoals besturingsmodellen, productbeschrijvingen, inrichting van de organisatie en de informatievoorziening. Logisch, want het is een *procesmodelleertechniek*. Dat borgen is echter wel belangrijk. Het slim borgen van deze samenhang staat ook wel bekend als 'architectuurdenken'. Architectuurdenken staat momenteel erg in de belangstelling en is zeker zo belangrijk als het al dan niet vanuit de klant geredeneerde procesgeoriënteerde denken.

De waarde van Petrinet+

Petrinet+ houdt de klassieke waarden van de Petrinet-schematechniek over-eind maar vult deze op een aantal fronten aan. Het geeft de niveaus van de processen en de relatie tussen die processen (de communicatie) een specifieke bedrijfskundige betekenis. Bovendien slaat het, vanuit die bedrijfskundige optiek, bruggetjes naar de andere architectuurdomeinen. We zullen hierna zien dat je eigenlijk maar weinig hoeft te doen om zinvolle niveaus te onderkennen, de communicatie tussen de processen te structureren en daarmee de modellen geschikt te maken voor verschillende doelgroepen en ook nog eens de samenhang naar andere architectuurdomeinen te borgen.

Het klassieke Petrinet kent slechts vier bouwstenen om processen te modelleren: de transitie, de plaats, de verbinding en het token (zie ook afbeelding 1). De transitie beschrijft een activiteit die de toestand van het proces (of de status van het geval) verandert. Zoals we gezien hebben kunnen we bij een hoog niveau Petrinet een aantal transities hiërarchisch groeperen. Bij Petrinet+ proberen we dit groeperen en de niveaus waartoe dit leidt vanuit een bedrijfskundige optiek in te vullen: wat maakt zo'n proces, hoe wordt het proces bestuurd en wie is er verantwoordelijk voor het proces? Maar

wat misschien nog wel belangrijker is: we worden ook gedwongen na te denken over de herbruikbaarheid van de bouwstenen en daarmee over de flexibiliteit van het procesmodel. Dit leidt tot een veel rijker scala aan bouwstenen. In afbeelding 2 is een vergelijking gemaakt tussen de Petri-net-bouwstenen en de Petrinet+-bouwstenen. Naast de semantische niveaus kunnen we in Petrinet+-modellen ook nog steeds groeperingen toepassen, enkel en alleen vanwege de overzichtelijkheid (visuele niveaus). Als je deze semantiek onderkent en je groepeerde de processen volgens werkproces, processtap en zovoort, dan worden de modellen veel eenvoudiger en eenduidiger en daarmee krachtiger en sprekender. In een aantal andere schematechnieken komt deze benadering overigens ook terug², maar Petrinet+ gaat nog verder of anders gezegd, men kan er nog verder mee gaan. Petrinet+-modellen kunnen, mits men de conventies zorgvuldig toepast, bouwbaar modellen opleveren. Als de afdeling automatisering hierop wacht, kan ze het bedrijf echt geven wat het nodig heeft.

Bouwstenen van Petrinet+

We zullen nu beschrijven wat de karakteristieken zijn van de verschillende niveaus binnen Petrinet+ en de communicatie tussen die niveaus. Het gaat om vijf niveaus, te weten: de keten, het bedrijfsproces, het werkproces, de processtap en de handeling.

Niveau 1. De keten

Op het hoogste niveau onderscheiden we de keten. De keten is een sequentie van processen over het bedrijf heen met vastgelegde communicatieafspraken. Met de ketenpartners (of een klant, dat maakt eigenlijk niet veel uit) worden afspraken gemaakt over de wijze van communicatie. De communicatie kan op analoge wijze verlopen (brief of telefoonje) of digitaal (e-mail, FTP). Ook

kan er veel structuur in de communicatie zitten (XML-bericht) of juist weinig (een handgeschreven brief). Verder kan de communicatie zich kenmerken doordat het enkel een verwijzing naar de inhoud geeft ("volgens ons eerdere bericht al in uw bezit") of dat alle inhoud van de communicatie in het bericht zelf zit.

Niveau 2. Het bedrijfsproces

Het bedrijfsproces ligt een niveau lager dan de keten. Het bedrijfsproces is een reeks processen binnen de muren van het bedrijf die de initiële vraag van de klant verwerkt tot (het beoogde) product voor die klant. De afspraken met zo'n klant worden in zo'n bedrijfsproces ook bewaakt. Zo mag de doorlooptijd van de behandeling bijvoorbeeld niet langer zijn dan drie weken. Een (instantie van een) bedrijfsproces wordt altijd aangemaakt door een initiële trigger.

Niveau 3. Het werkproces

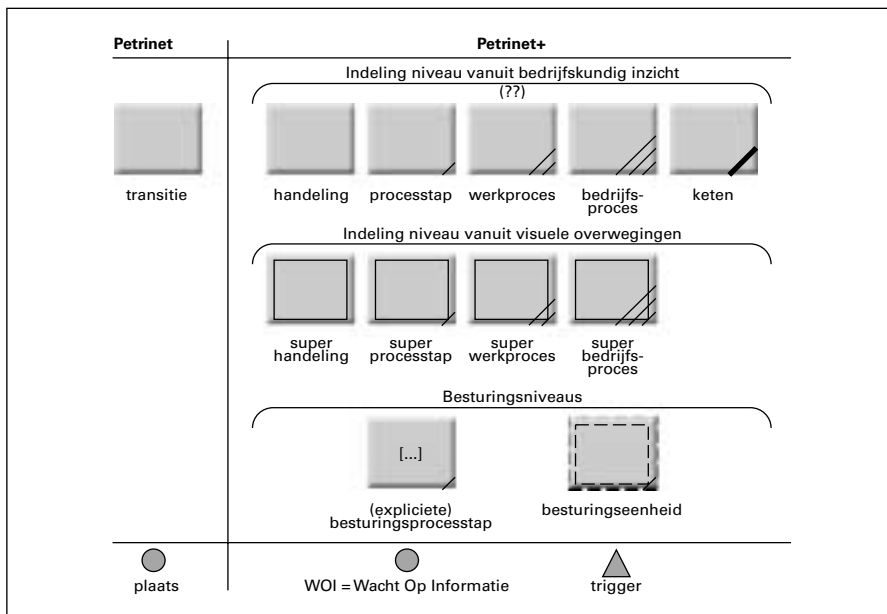
Het bedrijfsproces bestaat uit één of meerdere werkprocessen. Typisch aan een werkproces is dat het altijd één object bestuurt, overeenkomstig de grootte van het geval. Tussen werkprocessen kunnen er overgangen plaatsvinden in de grootte van het bestuurd object. Zo kan een aanvraag bijvoorbeeld 'in bulk' binnenkomen, zoals de postzak met brieven of een tape met belastingaangiften van meerdere werkgevers. Het eerste werkproces zorgt ervoor dat via 'de-batching' het te besturen geval tot op het juiste niveau wordt teruggebracht. Bijvoorbeeld door de post te scannen en de brieven vervolgens één voor één toe te wijzen aan de juiste medewerker. Binnen de referentie-architectuur van de UWW worden al deze processen tezamen inputmanagement genoemd. Het volgende werkproces gaat er vervolgens per geval mee aan de slag. Een andere reden om het bedrijfsproces op te splitsen in meerdere werkpro-

cessen is dat gedurende het bedrijfsproces meerdere interne afnemers parallel iets moeten doen. Denk hier bijvoorbeeld aan een constatering van samenloop op uitkeringen. Op basis hiervan kan een organisatie een onderzoekstraject op fraude starten én de excasso-afdeling op de hoogte stellen. Men kan zelfs werkprocessen geheel onafhankelijk opstarten. Omdat er meerdere interne afnemers van de producten of diensten van een werkproces kunnen zijn, kan het resultaat van een werkproces een halfproduct of deeldienst zijn. Het werkproces bepaalt dus ook de grootte van een proceseigenaar of de resultaatverantwoordelijke eenheid (RVE) binnen een bedrijf.

Een werkproces wordt altijd geïnitieerd door een trigger en het eindigt met een trigger die het halfproduct of deeldienst definieert. Gedurende een werkproces kan men echter ook een trigger uitsturen (bijvoorbeeld een brief voor de aanvraag van extra informatie). Een reeds opgestart werkproces kan ook triggers ontvangen, bijvoorbeeld een brief met die extra informatie. Dit kunnen we dan beschouwen als een doorstarttrigger. Triggers kunnen ook het werkproces verstoren. De interrupt is dus een trigger die van buiten het werkproces op diverse plaatsen in het werkproces kan ingrijpen. Binnen Petrinet+ worden de verwerking van dit soort interrupts, alhoewel naast het model, volledig uitgemodelleerd. De teken-techniek zorgt ervoor dat dit soort ingewikkelde constructies dan beter leesbaar is.

Niveau 4. De processtap

Een werkproces is weer opgedeeld in processtappen. Een processtap is een proces dat wordt uitgevoerd door een specifieke rol of functie; die rol kan ook een server zijn. De processtap leidt tot meerwaarde in het werkproces en dus in het maken van het deelproduct of de deeldienst.



Afbeelding 2. Bouwstenen klassiek Petrinet versus Petrinet+.

Meestal kent een werkproces een aantal procestappen die medewerkers in een bepaalde vaste volgorde moeten afhandelen: eerst informatie controleren, dan de uitkering vaststellen. Als er echter te weinig informatie aanwezig is, moet men eerst extra informatie inwinnen. Via een soort zijpad wordt er eerst iets gedaan voordat het proces weer verder kan. Het kan ook zo zijn dat als de uitkering hoger is dan een bepaald bedrag, er eerst nog iemand naar moet kijken voordat de uitkering echt definitief wordt gemaakt. Wat opvalt is dat men vóór elke processtap gevallen op een stapeltje legt en hier weer één voor één gevallen vanaf haalt. De uitkomst van een processtap kenmerkt zich dan ook als wachtmoment voor een volgende processtap. Het onderkennen van zo'n wachtmoment is zeker zo belangrijk als het onderkennen van de processen zelf. Het geeft immers inzicht in hoe lang werk ligt te wachten voordat men het oppakt. Daarom is het ook zeer belangrijk voor managementinformatie. De grootte van een processtap is gelijk aan wat men in UML-kringen de usecase noemt. Een processtap wordt getriggerd

omdat een trigger in principe eerst een wachtmoment maakt en tezamen met een reeds bestaande wachtmoment het proces opstart. Dit is de zogenaamde AND-constructie. Dit kan ook een trigger zijn die - indien dat zo gemodelleerd is - als interrupt werkt. Het geval kan natuurlijk ook door een persoon (met een bepaalde

Petrinet+ dwingt ons na te denken over de herbruikbaarheid van bouwstenen

rol) uit een wachtmoment worden gehaald. De actie van de betreffende persoon is dan eigenlijk de trigger; dit is zo triviaal dat we dit niet via een extra wachtmoment modelleren.

Als uit het procesmodel niet helemaal duidelijk is wat een trigger met een bepaalde waarde doet, dan onderkennen we een expliciet besturingsproces of beter gezegd: een besturingsprocesstap. Hierin beschrijven we onder welke condities het

procesmodel naar het ene wachtmoment gaat of naar een andere. Zo'n besturingsproces voegt dus eigenlijk geen waarde toe aan het proces, maar het geeft het geval onder besturing alleen maar een andere status in het proces. Organisaties gebruiken expliciete besturingsprocessen ook indien er sprake is van synchronisatie en andere moeilijk in een model te modelleren patronen. Zo kan een aantal ingewikkelde patronen van Van der Aalst³ met behulp van expliciete besturingsprocessen een stuk gemakkelijker worden beschreven.

Petrinet+ onderscheidt ook nog een niveau met een speciaal karakter: de besturingseenheid. Dit is een aantal typisch bij elkaar horende processtappen. De typering is die van een kenniswerker of die van een productiewerker. De procesmodelleur kan deze groepering binnen een werkproces aangeven. Bij de implementatie van het model kan men dan rekening houden met de inzet van casemanagementsoftware of een klassiek workflowtool voor de ondersteuning van deze processtappen.

Niveau 5. De handeling

Het kleinste proces is de handeling. De handeling kenmerkt zich door het uitvoeren van een afgerond stukje werk door mens of computer (binnen de rol) en wel zodanig, dat het een zekere mate van herbruikbaarheid bezit. Denk hierbij aan de bedrijfsfunctie controleren sofinummer, het voeren van een telefoongesprek met de klant voor extra informatie of het bijwerken van een recidive database. De grootte van een handeling is wat men binnen een usecase (UC) een regel met regelnummer noemt. Zo'n regel kan verwijzen naar een andere UC ('includes UC'; sub UC) of onder voorwaarden verwijzen naar een andere UC ('extends UC'; extension UC). Dit zijn beide superhandelingen in Petrinet+; afhankelijk van de plek

binnen de processtap en een eventuele conditie op de verbinding is het een sub-UC dan wel een extension UC. Zo'n usecase moet nog een aantal andere elementen bevatten (zie kadertje op deze pagina).

Usecases en het UC-model vormen de basis voor de bouw van informatiesystemen. Uit een Petrinet+-model vallen dus éénduidig usecases te genereren. Hierop kan men vervolgens andere UML-diagrammen maken. Het voordeel van doormodelleren tot aan het handelingenniveau is dat het procesmodel is geborgd en dat de handelingen ook grafisch zijn weer te geven.

Bouwbaar procesmodel

Tot en met de handeling (of de regel binnen de usecase) houden we ons eigenlijk alleen bezig met het feit dat er iets gebeuren moet. We hebben ons dus eigenlijk alleen beziggehouden met de functie en de werking van een proces (waar is het voor en wat levert het op) en nog niet met hoe het werkt (het ontwerp). Vergelijk dit met de black-box binnen Component Based Development. De processen kunnen ook van een verschillend type zijn. Petrinet+ onderscheidt drie typen: volledig handmatige, geautomatiseerd ondersteunde (workflow of casemanagement) en volledig geautomatiseerde processen. Het zal duidelijk zijn dat als alle handelingen in een processtap volledig zijn geautomatiseerd, de processtap zelf ook geautomatiseerd kan worden uitgevoerd. Als de processen zijn uitgemodelleerd tot en met de handeling ligt er ook een basis voor het Functioneel Ontwerp (FO), maar nog niet het FO zelf. We noemen het procesmodel dan ook een bouwbaar procesmodel.

De robuustheid en flexibiliteit van een model, alsmede de herbruikbaarheid van de gebruikte bouwstenen, kan men voor een groot deel bepalen door de korrelgrootte van de proces-

sen goed te kiezen en een duidelijk onderscheid te maken tussen verwerkingsprocessen en besturingsprocessen. Voor de flexibiliteit van het model en de herbruikbaarheid van bouwstenen is het belangrijk te onderzoeken hoe 'schoon' het verwerkingsproces is. Een schoon proces is een proces zonder besturingslogica.

Als de processtap 'schoon' is, doet het haar werk (bijvoorbeeld het uitrekenen van een uitkeringshoogte), zonder dat die processtap weet wat

Daarentegen mag informatie over de leeftijd van de persoon waarvoor de uitkering in dit geval wordt berekend in principe niet worden meegegeven aan de besturingslaag. De besturingslaag kan deze informatie namelijk zelf ophalen en als dat van belang is een beslissing nemen voor de vervolgstap. Na een processtap wordt er door de besturingslaag altijd een wachtmoment aangemaakt. De relatie tussen flexibiliteit en herbruikbaarheid 0gegeven in afbeelding 3.

	Flexibel	Herbruikbaar
1. Keten	++	--
2. Bedrijfsproces	+	-
3. Werkproces	+	+
4. Processtap	-	+
5. Handeling	--	++

Afbeelding 3. De flexibiliteit en herbruikbaarheid van Petrinet+-bouwstenen.

er vervolgens moet gebeuren. Daar hebben we namelijk de besturingslaag of Business Process Management-laag voor. Als een processtap klaar is en zich 'gereed meldt' aan de besturingslaag, dan moet daar zo weinig mogelijk procesinformatie bijzitten. Alleen informatie met betrekking tot het specifieke geval onder besturing mag worden meegegeven, bijvoorbeeld een foutcode na een controleproces. De besturingslaag beslist dan overigens nog steeds welk vervolproces het systeem op basis van die foutcode moet opstarten.

Triggermanagement

De flexibiliteit van de modellen wordt nog groter als de deelproducten alleen triggers opleveren zonder procesinformatie. De werkprocessen kunnen een abonnement nemen op deze triggers. Soms is dat vanzelfsprekend: een bedrijfsproces bestaat uit twee werkprocessen die elkaar normaal logisch sequentieel opvolgen. Soms is dat minder logisch, maar wel heel handig. Een uitkeringstoekenning groter dan vijftienduizend euro gaat ook naar een (extra) controleproces bij de sociale dienst. Zij hebben daar een abonnement op genomen. Een trigger met de naam faillissement zal veel afnemers kennen en zal bij veel werkprocessen als de afhandeling van een interrupt zijn gemodelleerd. Een trigger maakt in eerste instantie een wachtmoment: de trigger wordt in het werkproces bevroren. Het is ook mogelijk om een trigger rechtstreeks met een processtap te verbinden. Als er dan geen wachtmoment aanwezig is van waaruit de processtap kan worden opgestart, zal de trigger verloren gaan. In

Usecase	Petrinet+
Primary actor	Rol
Stakeholders	Betrokkenen (Taken Bevoegdheden Verantwoordelijkheden)
Preconditions	Wachtmomenten van waaruit processtap kan worden opgestart
Postconditions	Wachtmomenten waar een processtap kan eindigen
Trigger	Trigger of wachtmoment die door een trigger wordt gemaakt

Elementen van een usecase.

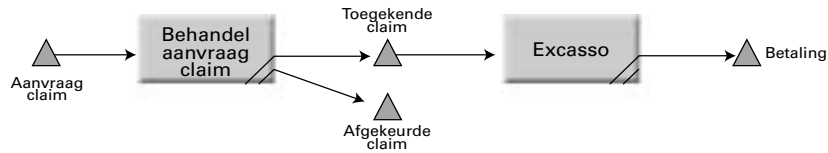
Een Petrinet+-model

In dit kader een voorbeeld van een Petrinet+-model van een proces waarbij het bedrijf extra informatie aanvraagt om een claim te kunnen

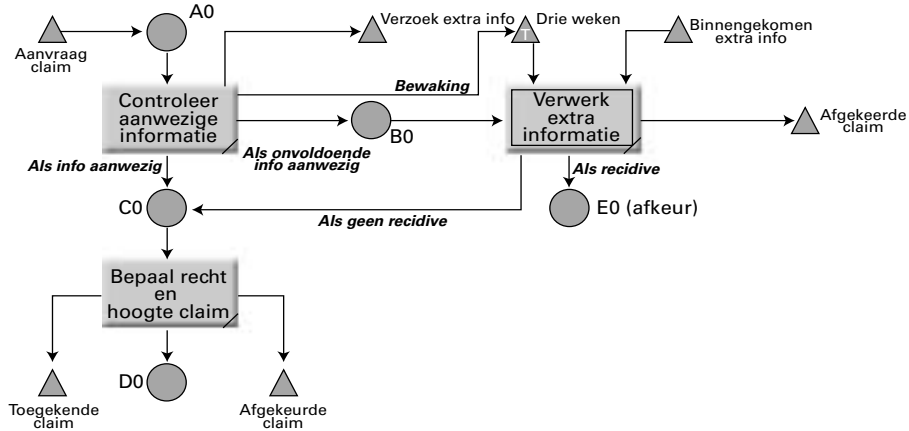
verwerken. In de recidive database staan de klanten die zich niet erg 'netjes' hebben gedragen in een eerder contact met het bedrijf. Als

er geen recidive is, verwerkt een medewerker van de organisatie de (extra) informatie.

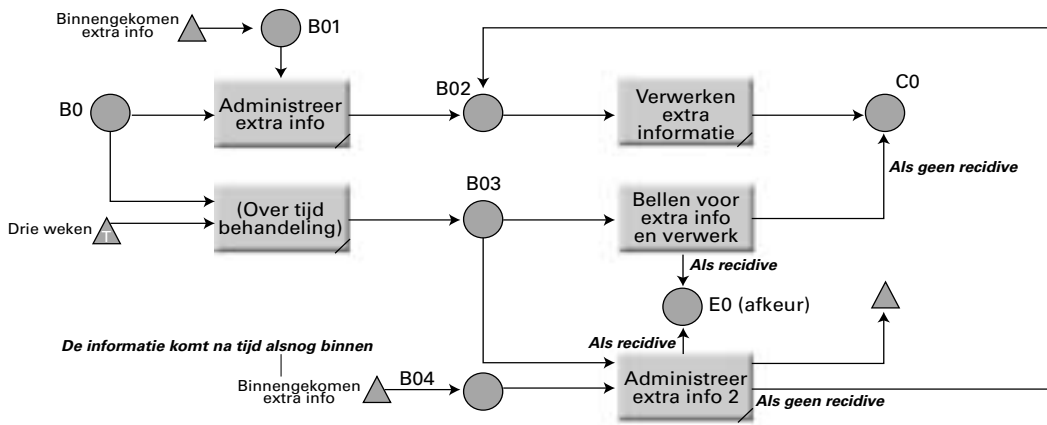
Bedrijfsproces "Claim" met twee werkprocessen



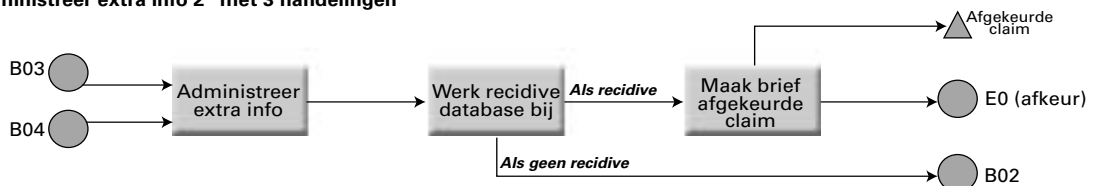
Werkproces "Behandel aanvraag claim" met twee processtappen en een (super)processtap



Superprocesstap "Verwerk extra informatie" met een (besturings)proces en vier gewone processtappen



Processtap "Administreer extra info 2" met 3 handelingen



Voorbeeld van een Petrinet+-model.

het kader 'Een Petrinet+-model' is een Petrinet+-model weergegeven waarin een aantal van de hier beschreven elementen terugkomt.

Architectuur

Het borgen van de overige architectuurdomeinen aan de processen is heel belangrijk. We gebruiken hier het architectuurmodel van Nolan Norton⁵. Ieder architectuurmodel heeft haar eigen (schema)technieken. Voorbeelden zijn organigrammen en rollenmodellen voor de organisatiearchitectuur, ERD of objectmodellen voor de gegevens, stuklijsten voor de productarchitectuur, enzovoort. Het is belangrijk dat deze architecturen op elkaar aansluiten. Vanuit de semantische niveaudefinities van Petrinet+ zijn er algemene richtlijnen te omschrijven voor de aansluitende architectuurdomeinen. We noemen dat het 'bruggetjes bouwen' vanuit de procesarchitectuur naar de overige architecturen. Hier is heel veel over te zeggen. We beperken ons hier tot enkele highlights:

- Vanuit de productarchitectuur is het bouwen van bruggetjes relatief eenvoudig. Een werkproces maakt halfproducten of deeldiensten. Een bedrijfsproces maakt producten en diensten;
- Vanuit de organisatiearchitectuur is een belangrijke brug het onderkennen van verschillende rollen met hun taken, bevoegdheden en verantwoordelijkheden (TBV) voor functionarissen bij elke operationele uitvoering van de processtap. Bovendien kan men aan een werkproces een

afdeling (resultaatverantwoordelijke eenheid RVE) koppelen;

- Ook voor de profielen met hun TBV die de processen moeten ontwikkelen en beheren kunnen we een relatie leggen. De business-engineer houdt zich bezig met bedrijfsprocessen en werkprocessen. De procesmodelleur schuift aan voor de verdere invulling van de processtappen en de system-engineer onderkent herbruikbare handelingen.

Vanuit de gegevensarchitectuur kunnen we stellen dat elke handeling een CRUD-operatie (Create, Read, Update, Delete) uitvoert op een bedrijfsobject, terwijl het bestuurd object wordt onderkend in het werkproces.

Reikwijdte en impactanalyse

Petrinet+ is een schematechniek. Het zegt niets over hoe je tot een bepaald model komt; dat bepaalt een modelleermethode met een stappenplan en dergelijke. Petrinet+ schrijft alleen voor hoe je een model moet opschrijven. Als er goede Petrinet+-modellen aanwezig zijn, dan kunnen organisaties deze overigens wel prima gebruiken als analyse-instrument om de effecten van een verandering via een reikwijdte en impactanalyse te beoordelen en een mogelijk verandescenario te bedenken. Petrinet+ is toolvrij. Dit betekent dat tools als Bwise, Testbed, Aris, Protos en andere tools prima in combinatie zijn te gebruiken met Petrinet+. Het één vervangt het ander niet, Petrinet+ stelt alleen specifieke en formele eisen met betrekking tot de syntax en semantiek van de te

gebruiken bouwstenen in die tools.

Noten

1. Patrick Savalle en Rob Limburg: De Synalysemethode, 90-806454-1-9, Synalyse BV, 2000.
2. Stef Joosten: Het effect van procesarchitectuur, Informatie, september 2001.
3. Wil van der Aalst e.a.: Workflow Patterns. Beta Working Paper Series. WP 47, Eindhoven University of technology, 2000.
4. ICT-referentiearchitectuur, UWV, versie 2.0, 18 maart 2002.
5. Nolan Norton Institute: Beheersing en besturing van complexiteit in het netwerktydperk / Architectuur als managementinstrument, Onderzoeksrapport, 1 december 1999.

Hans Tönissen, Willem Feijen, Ronald de Groot, Egon Willemsz en Mark Backer

Dit artikel is geschreven door Hans Tönissen, adviseur bij Van de Geijn Partners te Houten. Co-auteurs zijn Willem Feijen, Ronald de Groot, Egon Willemsz en Mark Backer (allen van PinkRocade), die samen met Tönissen het Petrinet+-project bij de UWV uitvoeren. Van het project is een rapport verschenen. Momenteel wordt gewerkt aan een herziene versie.

Advertentie

Adv Argeweb
50 x 100 mm

Guido Bayens (hoofd Bedrijfsarchitectuur UWV): "Een nauwkeurig uitgewerkte werkprocesbeschrijving is de basis voor de inrichting van het bedrijf. Dit procesmodel is de moeder van het organisatiemodel, het ontwerp van de informatiehuishouding en biedt een basis voor het ontwerp van het managementinformatiesysteem en de kwaliteitsborging. Het procesmodel zorgt daarmee voor de samenhang tussen al deze bedrijfsaspecten. Petrinet+ slaat een brug tussen de procesarchitectuur en de applicatiearchitectuur."