

TNO
Colosseum 27
7521 PV Enschede

www.tno.nl

T +31 88 8662450
F +31 88 8662147
info-ict@tno.nl

TNO-rapport

Rapport nr TNO 2012 R10544

MOSES: Model gebaseerde Ontwikkeling van SEmantische Standaarden

Datum	14 maart 2012
Auteurs	Ad Schrier, Michael van Bekkum, Dennis Krukkert, Jack Verhoosel en Jasper Roes
Exemplaarnummer	
Oplage	
Aantal pagina's	76
Aantal bijlagen	0
Opdrachtgever	Ministerie van Economische Zaken, Landbouw en Innovatie, Directoraat-Generaal Energie, Telecom en Markten
Projectnaam	MOSES
Projectnummer	055.01327

Alle rechten voorbehouden.

Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt door middel van druk, foto-kopie, microfilm of op welke andere wijze dan ook, zonder voorafgaande toestemming van TNO.

Indien dit rapport in opdracht werd uitgebracht, wordt voor de rechten en verplichtingen van opdrachtgever en opdrachtnemer verwezen naar de Algemene Voorwaarden voor opdrachten aan TNO, dan wel de betreffende terzake tussen de partijen gesloten overeenkomst.

Het ter inzage geven van het TNO-rapport aan direct belang-hebbenden is toegestaan.

Inhoudsopgave

1	Inleiding	4
2	Leeswijzer	7
3	Bedrijfsdomeingebaseerde interoperabiliteits-ontwikkeling	9
3.1	Wat is interoperabiliteit?	9
3.2	MOSES methode en interoperabiliteit	10
3.3	Welke belemmeringen kennen bestaande aanpakken?	11
3.4	Wat is een bedrijfsdomeingebaseerde aanpak?	12
3.5	Waarom een modelgerichte, methodische aanpak?	13
3.6	Wat levert deze aanpak op?	14
4	Inleiding in de methode	15
4.1	Denkwijze	16
4.2	Werkwijze	17
4.3	Begrippen en Technieken voor domeinanalyse	19
4.3.1	Het begrip actor	20
4.3.2	Het begrip object	20
4.3.3	Het begrip gebeurtenis	20
4.3.4	Een techniek voor domeinanalyse	21
4.4	Notatiewijzen	21
4.4.1	Een notatiewijze voor objectrelaties	21
4.4.2	Een notatie voor gebeurtenissen en objectgebeurtenis relaties	22
4.4.3	Een notatie voor volgordebependingen van gebeurtenissen	23
4.5	Een laatste opmerking rondom het GDM	25
4.5.1	De techniek om van een GDM een GIM te maken	25
5	De methode toegepast	27
5.1	Achtergrond en Context - Vereenvoudigd Bestellen en Leveren in de Juweliers Branche	27
5.2	Het Gemeenschappelijke Bedrijfsdomeinmodel (GDM)	28
5.2.1	Klassendiagram - interobjecttype afhankelijkheden:	30
5.2.2	AOE_Tabel (Actor - Object - Event) - Participaties van Partijrollen en Objecttypen in Bedrijfsgebeurtenissen:	31
5.2.3	Bestel- en Leverantieprocessen in de keten en de levenscyclus van de objecttypen	33
5.2.4	Detailering GDM:	34
5.3	GIM - Vereenvoudigd Juweliersbranche Informatiemodel	36
5.3.1	Gebeurtenis: plaatsenBestelling en het Informatieobject: Bestelformulier	36
5.3.2	Gebeurtenis: bevestigenBestelling en het informatieobject: Bestelbevestiging	38
5.3.3	Gebeurtenis: vooraankondigenLeveren en het informatieobject: Pakbon	41
5.3.4	Request: AanvragenCatalogus met informatieobjecten: CatalogusAanvraag en Catalogus	42
6	Modelimplementatie	44
6.1	Inleiding implementatie	44
6.1.1	Transformatie van modellen	44
6.1.2	Standaard oplossingsarchitecturen voor realisatie van interactie	44
6.1.3	Gebruik van bouwstenen	44
6.2	Transformatiepatronen	44
6.2.1	Van bedrijfsgerichte berichtspecificaties naar technische berichtspecificaties	44
6.2.2	Implementatie van dialogen	45

6.2.3	Afbeelding van de specificatie op bestaande uitwisselingsstandaarden	45
7	Semantische notaties	46
7.1	Het gebruik van semantische notaties in MOSES	46
7.1.1	Oplossingsrichting	46
7.1.2	Aanpak	47
7.1.3	Web Ontology Language en Protegé	47
7.2	Model transformatie	48
7.2.1	MOSES meta-model	48
7.2.2	Modeltransformatie	50
7.2.3	MOSES voorbeeldcase	50
7.3	Juweliersstandaard uitgedrukt in OWL	51
7.3.1	Annuleren bestelling	52
7.3.2	Annuleren levering	53
7.3.3	Plaatsen bestelling	55
7.3.4	Voor aankondigen levering	56
7.3.5	Bevestigen bestelling	58
7.3.6	Samenvatting	60
7.4	Evaluatie	60
7.4.1	Aanpak	60
7.4.2	Meest geschikt meta-model	60
7.4.3	Gebruik van semantische talen	61
7.4.4	Future research	61
8	Conclusies en mogelijke uitbreidingen	63
8.1	Formelere onderbouwing en semantische notatiewijzen	63
8.2	Werkwijzen en ontwikkelingsstrategieën	64
9	Appendices	65
9.1.1	OWL specificatie MOSES meta-model	65
9.1.2	OWL specificatie – Juweliersbranche	67

1 Inleiding

Dit document beschrijft een methode om semantische standaarden op een model gebaseerde manier te ontwikkelen. Deze methode, die we MOSES (Model gebaseerde Ontwikkeling van Semantische Standaarden) hebben gedoopt, is ontwikkeld in het kennisopbouw project MASIEF dat wordt uitgevoerd door TNO in het kader van de DGET-regeling van het Ministerie van Economische Zaken, Landbouw en Innovatie. MOSES is gebaseerd op de kennis die de auteurs van dit document hebben opgedaan in verschillende projecten waarin semantische standaarden zijn ontwikkeld voor verschillende branches gecombineerd met uitgebreide kennis over modelgedreven ontwikkeling van software. De combinatie van deze kennis heeft geleid tot deze MOSES methode.

Met behulp van MOSES is het mogelijk om op een modelgebaseerde wijze semantische standaarden te ontwikkelen. Bij deze manier van ontwikkelen van semantische standaarden, wordt het koppelvlak waarop wordt samengewerkt vastgelegd in modellen. Op basis daarvan wordt gespecificeerd hoe informatie tussen partijen uitgewisseld wordt. Dit is volgens de auteurs een uitstekende manier om de semantiek van zowel de informatie als processen vast te leggen. Door met behulp van MOSES semantische standaarden te ontwikkelen wordt er een expliciete scheiding aangebracht tussen de semantiek van de informatie en processen in het bedrijfsdomein aan de ene kant, en de daadwerkelijke manier van het uitwisselen van informatie aan de andere kant. Door deze splitsing legt de opsteller van de standaard de manier van uitwisselen niet in de standaard vast en is het eenvoudiger om te wisselen tussen verschillende uitwisselingsmanieren (denk hierbij aan web services, e-mail of uitwisseling door middel van een gedeelde database). Het gaat volgens de auteurs bij een semantische standaard namelijk primair om de betekenis van de uitgewisselde informatie en de processen (dé semantiek!) en niet om de vorm van uitwisseling.

Het ontwikkelen van oplossingen op basis van een bedrijfsdomein model biedt de mogelijkheid om kritisch na te gaan welke informatie-uitwisseling nu echt nodig is binnen het vastgestelde probleem gebied, in plaats van (klakkeloos) bestaande papieren formulieren te “elektronificeren”. Door de methodische aanpak wordt de kwaliteit van de gerealiseerde oplossingen (interactie-standaarden) verhoogd. Daarbij gaat het om semantische kwaliteit zoals de precieze, eenduidigheid, correctheid en compleetheid van informatie. Met precieze en eenduidigheid bedoelen we dat alle betrokkenen exact weten wat de beschreven informatie betekent, en deze ook op dezelfde manier interpreteren. Daarnaast gaat het ook om de pragmatische kwaliteit zoals de bruikbaarheid, begrijpbaarheid, en overdraagbaarheid van de modellen. Om dit te bewerkstelligen beperken we ons tot een minimale maar voldoende taal die in ieder geval duidelijk en eenduidig zijn, maar vooral ook goed aansluiten bij de belevingswereld van belanghebbende.

Met zulk een methodiek kan TNO diverse sectoren in de zakelijke dienstverlening en het overheidsdomein helpen met het opstellen van kwalitatief goede en goed onderhoudbare standaarden. Dit kan leiden tot efficiëntere samenwerking tussen de betrokken organisaties. Daarnaast biedt het bedrijfsdomeinmodel een stabiele basis voor de ontwikkeling van oplossingen. Ontwikkelde modellen kunnen gemakkelijk hergebruikt worden bij het opkomen van nieuwe technologieën, oftewel, de onderhoudbaarheid van de oplossing neemt toe. Daarmee zijn met name de IT-afdelingen van betrokken organisatie geholpen die nieuwe software ontwikkelen of inkopen.

De MOSES methode is geen 100% formele methode om semantische standaarden te ontwikkelen. Dat zou het wel kunnen worden en daarom moet de huidige versie worden gezien als een work-in-progress en daarmee een in ontwikkeling zijnde methode. De auteurs zien deze beschrijving van de methode als een goed uitgangspunt voor het ontwikkelen van semantische standaarden en moedigen iedereen aan de methode toe te passen en vragen of opmerkingen door te spelen om de methode te kunnen verbeteren en verder aan te scherpen.

Wij denken met de ontwikkeling van MOSES een nuttige toevoeging te hebben gedaan aan het domein van semantische standaarden en wensen u veel lees- en toepasplezier,

Ad, Dennis, Jack, Michael & Jasper, Enschede, maart 2011

2 Leeswijzer

Dit document beschrijft MOSES, een methode voor Model gebaseerde Ontwikkeling van SEmantische Standaarden. Hoofdstuk 3 van dit document geeft een inleiding op bedrijfsdomeingebaseerde interoperabiliteit ontwikkeling, waar het binnen MOSES om draait. Daarnaast geeft dit hoofdstuk ook aan wat de auteurs onder interoperabiliteit verstaan, welke belemmeringen gezien worden in bestaande aanpakken, wat MOSES toevoegt en wat de waarde is van de methode.

In hoofdstuk 4 wordt een inleiding gegeven op de methode. De denkwijze, werkwijze en notatiewijze van de methode worden toegelicht om de lezers van dit document en de toepassers van de methode inzicht te geven in de gedachten achter de methode. Hoofdstuk 5 beschrijft de methode zelf waarbij een case uit de juweliersbranche gebruikt wordt als lopend voorbeeld. Het hoofdstuk beschrijft de verschillende onderdelen van de methode en de geeft daarbij per onderdeel een uitwerking specifiek voor de juweliersbranche. De juweliersbranche-case is gekozen vanwege de uitgebreide ervaringen die de auteurs in die sector hebben opgedaan met een eerste ruwe versie van de methode. Door gebruik te maken van deze case proberen de auteurs de lezer en toepasser te helpen bij het begrijpen en gebruiken van de methode.

Hoofdstuk 6 geeft inzicht in mogelijke modelimplementaties van het semantische model, één van de mogelijkheden die beschreven wordt is een implementatie in XML. Hoofdstuk 7 tot slot beschrijft beknopt mogelijke uitbreidingen van de methodiek die nog verder onderzocht kunnen worden in een mogelijk vervolgtraject.

3 Bedrijfsdomeingebaseerde interoperabiliteits-ontwikkeling

Dit hoofdstuk introduceert de gedachtegang achter het ontwikkelen van interoperabiliteitsoplossingen op basis van een goed model van het bedrijfsdomein. Daartoe wordt in sectie 3.1 eerst beschreven wat in onze ogen interoperabiliteit is. Sectie 3.3 geeft aan wat de belemmeringen zijn in bestaande aanpakken en sectie 3.4 definieert wat een bedrijfsdomeingebaseerde aanpak is. In sectie 3.5 wordt toegelicht waarom de modelgerichte, methodische aanpak is ontwikkeld. Sectie 3.6 sluit dit hoofdstuk af met een beschrijving van wat de toegevoegde waarde van de MOSES-methode is.

3.1 Wat is interoperabiliteit?

Interoperabiliteit gaat over:

- doelmatig samenwerken met behulp van informatievoorzieningen
- samenwerkende, genetwerkte organisaties (CNO)

Begrippen: Interoperabiliteit, Collaborative Networked Organisation (CNO)

Het door de EU gefinancierde “Network of Excellence” Interop heeft interoperabiliteit gedefinieerd als *“the ability for a system or a product to work with other systems or products without special effort of the part of the customer”*. Interoperabiliteit gaat dus over de mogelijkheid van systemen (producten) om samen te werken.

De MOSES methode is gericht op het ontwikkelen van oplossingen voor samenwerking tussen organisaties. Daarom hanteren wij de volgende definitie van interoperabiliteit:

Interoperabiliteit

Interoperabiliteit is de mogelijkheid van organisaties om doeltreffend en doelmatig samen te werken met behulp van een informatievoorziening

Het organisatorische samenwerkingsverband vormt daarmee het primaire aandachtsgebied van de beoogde interoperabiliteitsoplossingen. Het te bestuderen probleemdomein wordt gevormd door het dagelijkse (gemeenschappelijke) reilen en zeilen van de partners in het verband. Een dergelijk verband noemen we een CNO (Collaborative Networked Organisation).

CNO Een CNO is daarbij door Tapia¹ gedefinieerd als: “elk mix-en-match netwerk van winst&verlies-verantwoordelijke organisatorische eenheden of onafhankelijke organisaties, verbonden door ICT welke gedurende een periode samenwerken om gemeenschappelijk taken te volbrengen, doelen te bereiken en klanten te bedienen”.

¹ Assessing business-IT alignment in networked organizations; R. G. S. Tapia; Dissertation UT 2009]

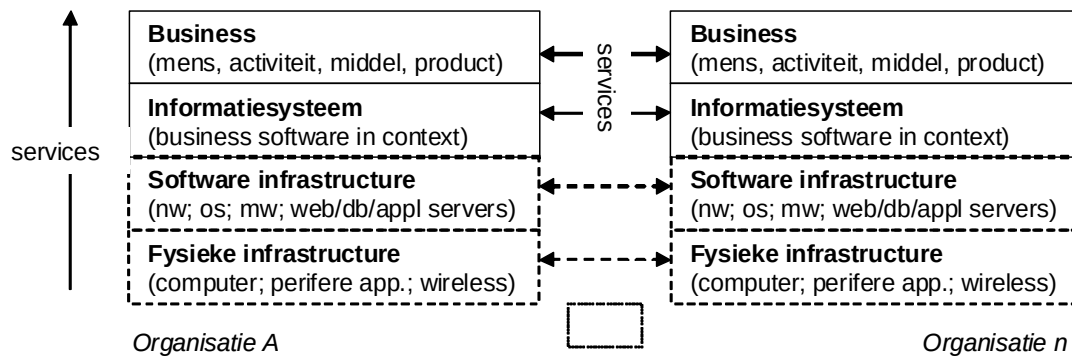
3.2 MOSES methode en interoperabiliteit

De MOSES methode richt zich op samenwerking tussen organisaties op twee niveaus: het niveau van de bedrijfsactiviteiten (business) en dat van de informatiesystemen (business software) die deze bedrijfsactiviteiten direct ondersteunen.

Begrippen: 4-lagen architectuur

4-lagen
architectuur

Om interoperabiliteit te bewerkstelligen moet op verschillende niveaus samenwerking gerealiseerd worden. We maken gebruik van een gebruikelijke, standaard 4-lagen architectuur in de wereld van bedrijfsinformatiesystemen (Figuur 1).



Figuur 1: CNO (Collaborative Networked Organisation)

Op het bovenste niveau vinden we de samenwerking, dienstverlening en activiteiten op bedrijfsniveau tussen mensen en organisaties (bv factureren). Op het tweede abstractieniveau gaat het over bedrijfsinformatiesystemen (business software) die de bedrijfsactiviteiten ondersteunen (bv facturatiesystemen). Niveau drie en vier vormen de software en fysieke infrastructuur die samenwerking mogelijk maakt (netwerkverbindingen, web servers, computers met operating systeem, etc.).

Infrastructuur
als gegeven

De MOSES methode richt zich op het methodisch ontwerpen van de elektronische samenwerking, de interactie tussen organisaties. De focus ligt daarbij op de bovenste twee lagen van de 4-lagen architectuur. We hanteren het uitgangspunt dat een oplossing voor interoperabiliteit gerealiseerd wordt op basis van een generieke, alom aanwezige, technische infrastructuur (het Internet met haar standaarden en protocollen). In figuur 1 veronderstellen we de software en fysieke infrastructuur dus als gegeven.

3.3 Welke belemmeringen kennen bestaande aanpakken?

De MOSES methode neemt het bedrijfsdomein als uitgangspunt voor een toekomstvast ontwerp van elektronische samenwerking.

Begrippen: bedrijfsdomeingebaseerde benadering

MOSES is een methode voor interoperabiliteitsvraagstukken bij interactie en samenwerking tussen organisaties. We kunnen dit interactievraagstuk op uiteenlopende wijzen benaderen. We kunnen ons bijvoorbeeld gelijk van het begin af aan richten op het definiëren van de betekenis, vorm en inhoud van berichten tussen de partijen in het uitwisselingsdomein, waarbij naar analogie van bestaande papierstromen een oplossing wordt gezocht. Deze berichten kan men vervolgens in een XML-schema representeren, zodat de papierstroom als het ware "elektronisch" wordt gemaakt.

In een dergelijke (veel toegepaste en pragmatische) aanpak wordt direct gezocht naar een oplossing, zonder dat het feitelijke probleem goed geanalyseerd wordt. Het op te lossen probleem van samenwerking en interoperabiliteit wordt in feite gereduceerd tot een "berichtenuitwisselings-probleem".

Nadelen van deze benadering zijn echter:

- Geen expliciete bestudering van mogelijke alternatieve samenwerkings- en interactiemodellen voor de samenwerking
- Geen expliciete aandacht voor kwaliteitsaspecten van interoperabiliteit
- Oplossingsarchitectuur- en technologiegericht en daarom minder toekomstvast (de aanpak is beperkt tot nu bekende techniek en oplossingen).

Om deze nadelen te ondervangen willen wij de reikwijdte van onze methode verbreden. Hiertoe hanteren wij een zogenaamde bedrijfsdomeingebaseerde benadering, naar analogie van gebruikelijke domeingedreven en modelgebaseerde software-engineering methoden (met name JSD² en MERODE³). In plaats van "berichten-ontwerp-methode" krijgen we dan een "electronische-samenwerking-ontwerp-methode".

² <http://cisx2.uma.maine.edu/NickTemp/JSP&JSDLec/jsd.html>

³ <http://merode.econ.kuleuven.be/>

3.4 Wat is een bedrijfsdomeingebaseerde aanpak?

MOSES gaat uit van samenwerking tussen organisaties op basis van een gemeenschappelijk en stabiel domeinmodel, waarin begrippen uit de bedrijfsmatige werkelijkheid van de samenwerkende organisaties met elkaar in verband zijn gebracht.

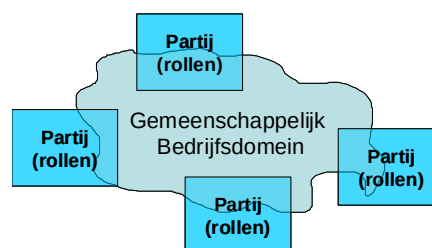
Begrippen: gemeenschappelijk bedrijfsdomeinmodel

Gemeenschappelijk
bedrijfsdomein
model

Partijen die op een structurele manier (willen gaan) samenwerken hebben baat bij structurele interactieafspraken. Voorwaarde voor dergelijke afspraken is een gedeeld beeld van de werkelijkheid waarin men samenwerkt. Dit beeld kan worden vastgelegd in een gemeenschappelijk, door de partijen gedeeld, model van het bedrijfsdomein (mogelijk in de vorm van een branche-model). Hierin zijn de kernbegrippen van het gemeenschappelijk handelen in de werkelijkheid weergegeven op abstracte, maar precieze wijze.

Een dergelijk model benoemt kernbegrippen en de relaties daartussen. Deze kernbegrippen handelen in eerste instantie niet over formuleren, lijsten en dossiers die we elkaar in de vorm van berichten sturen maar over de betekenis van concepten en conventies rond onze primaire materiaal-, diensten-, besturings- en geldstromen. Het model neemt het identificeren, bestuderen, beschrijven en formaliseren van gemeenschappelijke bedrijfsverschijnselen in en rond deze stromen als uitgangspunt. Dat wil zeggen: de elementaire begrippen (bv. bestelling, artikel, levering) waar de eventuele uiteindelijke berichten over kunnen gaan staan centraal, in plaats van de berichten (zoals bv. bestelformulier, pakbon, voorraadlijst).

Het hieruit resulterende bedrijfsdomeinmodel beschrijft “de kern” van de gemeenschappelijke wereld van de verschillende partijen (Figuur 2). Dit bedrijfsdomeinmodel is (relatief) stabiel, want het is een beschrijving van het domein die los staat van de techniek die wordt gebruikt om de bedrijfsactiviteiten uit te voeren (of dat nu met papier, mensen of informatiesystemen of berichten is). Het vormt zo dus een basis voor verdere stapsgewijze, modelgerichte ontwikkeling van de interoperabiliteitsoplossing.



Figuur 2: Multi-actor netwerk met gemeenschappelijk bedrijfsdomein

3.5 Waarom een modelgerichte, methodische aanpak?

MOSES hanteert een aanpak in vier stappen om het interoperabiliteitsvraagstuk aan te pakken:

1. Opstellen bedrijfsdomeinmodel
2. Vastleggen gedeelde informatiebehoefte in het domein
3. Opstellen berichtdefinities en dialoogprocessen
4. Vertalen berichten naar technologische oplossing (bv XML)

Begrippen: modelconsistentie

Onze aanpak hanteert als uitgangspunt dat alle, voor het voorliggende interoperabiliteitsvraagstuk relevante, zaken expliciet benoemd en beschreven worden, teneinde een kwalitatief betere oplossing te realiseren.

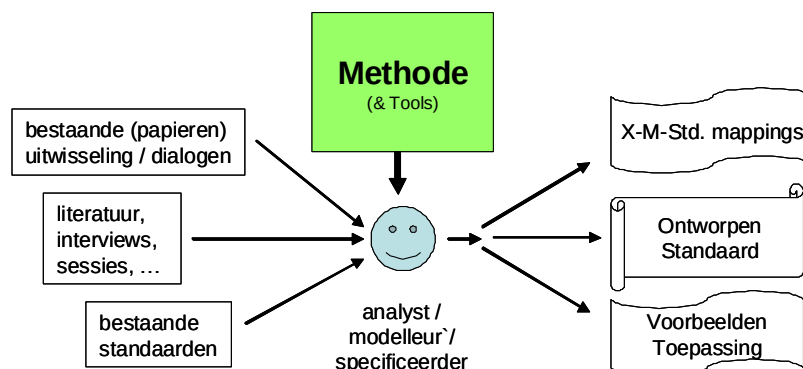
4-stappen aanpak

In een viertal stappen gaat de aanpak van het beschrijven van het probleemgebied naar het vastleggen van de uiteindelijke oplossing:

1. *Bedrijfsdomeinmodel*: het bedrijfsdomein beschrijft het voorliggende probleemgebied, daar waar de betreffende wereld over gaat.
2. *Vastleggen informatie*: Op basis van een bedrijfsdomeinmodel volgt het vastleggen van de informatie die in het domein gedeeld wordt. Denk hierbij aan zaken als besteloverzichten en voorraadlijsten. Deze informatie heeft een meer veranderlijk karakter dan de concepten in het bedrijfsdomeinmodel, maar is nog wel oplossingsonafhankelijk.
3. *Definitie berichten*: Vervolgens wordt op basis van de voorgaande stappen de aan deze informatie refererende, veranderlijke berichten gedefinieerd (in geval het berichtenparadigma als oplossingsrichting wordt gekozen). Denk hierbij aan berichten als verzoek-Voorraadlijst, respons-Bestellingsstatusoverzicht en notificatie-van-Levering. Ook worden bijbehorende dialoogprocessen vastgelegd.
4. *Vertalen naar technologieoplossing*: Tot slot wordt de vertaling gemaakt naar een specifieke technologische oplossing, in veel gevallen in de vorm van XML berichten.

Consistente modellen

Alle modellen die opgeleverd worden in de hierboven beschreven stappen hangen op een consistente wijze met elkaar samen. Aanpassingen in modellen in de ene stap, leiden mogelijk tot (bij voorkeur automatisch vertaalde) wijzigingen in een andere stap. Een methodische aanpak zoals boven omschreven, ondersteunt op een integrale wijze bij de ontwikkeling van globaal idee naar gerealiseerde interoperabiliteitsoplossing. Hoewel de methode ook andersoortige oplossingen ondersteunt, zal vooralsnog uitgegaan worden van de gangbare hedendaagse praktijk: de ontwikkeling van een interactiestandaard (Figuur 3).



Figuur 3: ondersteuning door methode.

3.6 Wat levert deze aanpak op?

De MOSES methode:

1. Voorkomt 'elektronificatie' van bestaande processen en oplossingen.
2. Verhoogt onderhoudbaarheid en duurzaamheid van de oplossing.
3. Verhoogt de kwaliteit van oplossingen (semantisch en pragmatisch).

Begrippen: modelconsistentie

In sectie 3.3 staat een aantal belemmeringen beschreven van huidige benaderingen, die met onze aanpak ondervangen worden:

- Geen expliciete bestudering van mogelijke alternatieve samenwerkings- en interactiemodellen voor de samenwerking.
- Oplossingsarchitectuur- en technologiegericht en daarom minder toekomstvast (de aanpak is beperkt tot nu bekende technieken en oplossingen).
- Geen expliciete aandacht voor kwaliteitsaspecten van interoperabiliteit

Het ontwikkelen van oplossingen op basis van een bedrijfsdomeinmodel biedt de mogelijkheid om kritisch na te gaan welke informatie-uitwisseling nu echt nodig is binnen het vastgestelde probleemgebied, in plaats van (klakkeloos) bestaande papieren formulieren te “elektronificeren”. Dit kan leiden tot efficiëntere samenwerking.

Daarnaast biedt het bedrijfsdomeinmodel een stabiele basis voor de ontwikkeling van oplossingen. Ontwikkelde modellen kunnen gemakkelijk hergebruikt worden bij het opkomen van nieuwe technologieën, oftewel, de onderhoudbaarheid van de oplossing neemt toe.

*Semantische
en pragmatische
kwaliteit*

Door de methodische aanpak wordt verder de kwaliteit van de gerealiseerde oplossingen (interactie-standaarden) verhoogd. Bij kwaliteit maken we onderscheid tussen semantische kwaliteit en pragmatische kwaliteit:

- *Semantische kwaliteit* gaat over de precisie, eenduidigheid, correctheid en compleetheid van informatie. Met precisie en eenduidigheid bedoelen we dat alle betrokkenen exact weten wat de beschreven informatie betekent, en deze ook op dezelfde manier interpreteren. Correctheid betekent dat alle uitspraken gedaan in het model juist en relevant voor het probleem zijn. Een manier om dit te waarborgen is interne consistentie van, en tussen, de beschreven modellen.
- *Pragmatische kwaliteit* gaat over de bruikbaarheid, begripbaarheid, en overdraagbaarheid van de modellen. Om dit te bewerkstelligen streven we o.a. naar een minimale maar voldoende taal. Deze taal moet in ieder geval duidelijk en eenduidig zijn, maar vooral ook aansluitend bij belevingswereld van de belanghebbende(n).

4 Inleiding in de methode

De MOSES methode kent een

- denkwijze: de filosofie en de vragen die men moet beantwoorden om tot een oplossing te komen
- werkwijze: volgorde van activiteiten bij het beantwoorden van de vragen
- notatiewijze: manier om antwoorden op de vragen vast te leggen

Begrippen: denkwijze, werkwijze, notatiewijze

Wij beschouwen de MOSES methode als een drie-eenheid van denk-, werk- en notatiewijze⁴ die ons op een integrale manier ondersteunt bij de ontwikkeling van globaal idee naar gerealiseerde interactiestandaard.

<i>Denkwijze</i>	De <i>denkwijze</i> beschrijft de filosofie van de methode en behandelt de vragen die we dienen te stellen bij de totstandkoming van de interactiestandaard met bijbehorende benaderingswijzen. Het beschrijft de wijze waarop we de (probleem)wereld beschouwen en bestuderen.
<i>Werkwijze</i>	De <i>werkwijze</i> beschrijft de stappen, en hun onderlinge volgorde, die we nemen bij het beantwoorden van de vragen. Het geeft concrete handvaten aan het invullen van de denkwijze.
<i>Notatiewijze</i>	De <i>notatiewijze</i> beschrijft de (grafische) talen die we gebruiken bij het vastleggen van de antwoorden, in de vorm van modellen en specificaties. De notatiewijze geeft vorm aan de resultaten van de activiteiten uit de werkwijze.

In dit hoofdstuk geven we een korte inleidende beschrijving van deze drie-eenheid. Een meer uitgebreide behandeling van de betekenis en totstandkoming van de uiteenlopende modellerings- en specificatieaspecten geven we in het volgende hoofdstuk. De uitgebreidere uitleg zal in dat hoofdstuk toegelicht worden aan de hand van een voorbeeldcasus.

⁴ Praktische toepassing van een methode wordt bovendien begeleid door passend en ondersteunend gereedschap.

4.1 Denkwijze

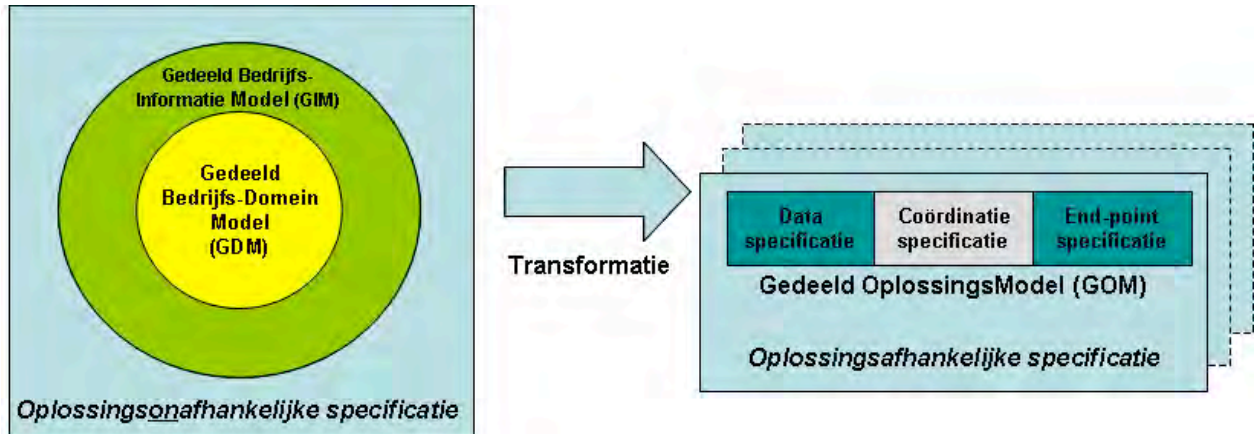
De MOSES denkwijze gaat uit van een 'bedrijfsdomeingerichte en modelgebaseerde benadering' en maakt onderscheid tussen twee soorten specificaties:

1. oplossingsonafhankelijk: beschrijft de gemeenschappelijke bedrijfsvoering, d.m.v. een Gedeeld Bedrijfs-Domein Model (GDM) en een Gedeeld Bedrijfs-Informatie Model (GIM)
2. oplossingsafhankelijk: beschrijft een gemeenschappelijke oplossing d.m.v. een Gedeeld OplossingsModel (GOM)

Begrippen: GDM, GIM, GOM

De denkwijze vormt de basis van onze benadering. Zoals beschreven in hoofdstuk 3 hanteren wij een 'bedrijfsdomeingerichte en modelgebaseerde benadering'. Dit is de kerngedachte achter de methode. De bedrijfsdomeingerichte benadering gaat uit van een gedeeld beeld van de werkelijkheid waarin partijen samenwerken, vastgelegd in een gemeenschappelijk, door de partijen gedeeld, model van het bedrijfsdomein. De modelgerichte benadering hanteert als uitgangspunt dat alle, voor het interoperabiliteitsvraagstuk relevante zaken expliciet benoemd en beschreven worden (in de vorm van modellen), zodat we een kwalitatief betere oplossing kunnen realiseren.

Wat deze beide benaderingen voor de MOSES methode betekenen, verwoorden we aan de hand van onderstaande illustratie.



We maken in de MOSES methode een formeel onderscheid tussen specificatie van de gemeenschappelijke bedrijfsvoering en specificatie van de gemeenschappelijke oplossing voor het interoperabiliteitsvraagstuk. In bovenstaand figuur is deze benadering uitgebeeld. Hier komen de twee pijlers van de denkwijze tot uiting:

1. De beschrijving van de bedrijfsvoering is losgekoppeld van de beschrijving van de oplossingsrichting (bedrijfsdomeingericht).
2. Beide beschrijvingen zijn in modellen gevat, die op transformeerbare wijze met elkaar samen hangen (modelgebaseerd).

Oplossingsonafhankelijke specificatie

De oplossingsonafhankelijke bedrijfsspecificatie representeert de probleemcontext en de informatiebehoeften die binnen deze context bestaan (gemeenschappelijke bedrijfsvoering). De oplossingsonafhankelijke specificatie bestaat uit twee delen:

- GDM* ■ Het Gedeeld Bedrijfs-Domein Model (GDM) representeert de gemeenschappelijke bedrijfsomgeving, m.a.w. de context van de interactieoplossing. Het GDM bevat die zaken “waar het in de kern over gaat”, zoals gemeenschappelijk bedrijfsvocabulaire en -regels, in de vorm van bedrijfsobject-, gebeurtenis- en regelspecificaties.
- GIM* ■ Het Gedeeld Bedrijfs-Informatie Model (GIM) beschrijft de relevante informatie van bedrijfsobjecten uit het GDM die uitgewisseld wordt bij het optreden van (gemeenschappelijke) gebeurtenissen in het domein. Hiertoe wordt per gebeurtenis beschreven welke bedrijfsobjecten relevant zijn, welke gegevens (attributen) van deze objecten relevant zijn, en welke regels gelden voor de uitwisseling. De uit te wisselen informatie wordt vastgelegd in een formulier.

Oplossingsafhankelijke specificatie

- GOM* De oplossingsgerichte specificatie representeert de technische structuren. In deze specificatie beelden we de bedrijfselementen uit GDM en GIM af op gegevens-, coördinatie- en (partij-end-point)diensten specificaties die specifiek zijn voor de oplossingsarchitectuur. Het Gedeeld Oplossingsmodel (GOM) representeert daarbij de vertaling van oplossingsonafhankelijke specificatie naar een specifieke oplossingsarchitectuur. Het GOM bevat o.a. data-uitwisseling, procescoördinatie en procesafhandeling. Voor elk formulier uit het GIM zal bijvoorbeeld een bericht (als een berichtenparadigma gehanteerd wordt) of een “scherm” (bij een gezamenlijke webapplicatie) ontwikkeld worden.

Van Oplossingsonafhankelijk naar Oplossingsafhankelijk

Bij één GDM en GIM zijn meerdere oplossingsrichtingen (GOM) mogelijk. Hierbij beschouwen we de interoperabiliteitsoplossing (gespecificeerd in het GOM) dus als een oplossingsarchitectuur-specifieke vertaling van de bedrijfsmodellen (GDM en GIM). De vertaling van een GDM en GIM is afhankelijk van het gekozen oplossingsparadigma, in lijn met het Model Driven Architecture (MDA) gedachtegoed. Zoals eerder aangegeven gaan we in deze beschrijving uit van de hedendaagse praktijk, te weten berichtgeoriënteerde oplossingen op basis van XML.

4.2 Werkwijze

De MOSES methode kent de volgende (iteratieve) werkwijze:

1. Bestuderen van het probleemgebied (actoren, context, het domein) en vastleggen in een GDM.
2. Formuleren van de informatie die wordt gedeeld in dit probleemgebied en vastleggen in een GIM.
3. Bedrijfsdomein- en informatiemodellen vertalen naar een oplossings specifieke architectuur en vastleggen in een GOM.

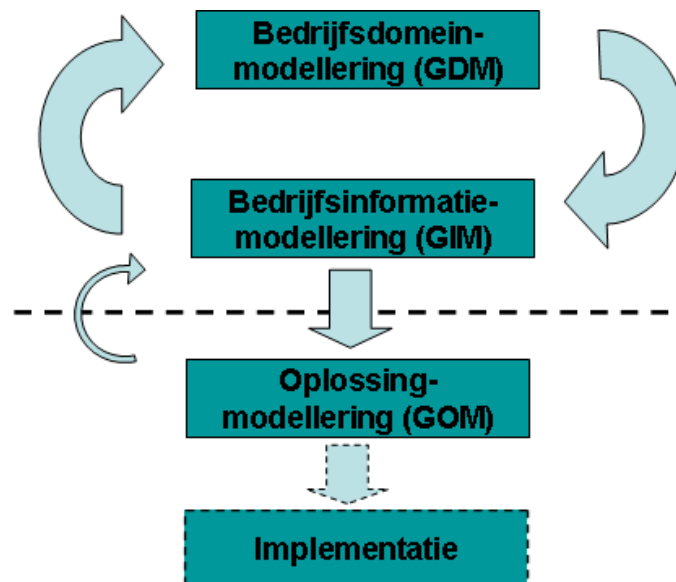
Begrippen: iteratieve werkwijze

De standaardfasering van de MOSES methode sluit aan op de samenhang tussen de specificaties, genoemd in 4.1. Standaard praten we daarbij over een middle-out-ontwikkelingsstrategie:

1. We beginnen met het bestuderen van het probleemgebied (actoren, context, het domein), en leggen dit vast in het GDM.
2. Als we hier een voldoende beeld van hebben formuleren we de informatie die we delen als een schil om het domeinmodel heen. Dit leggen we vast in het GIM.
3. Vervolgens kennen we een derde stap waarin we bedrijfsdomein- en informatiemodellen vertalen naar een oplossingsspecifieke architectuur.

*Iteratieve
werkwijze*

Hoewel dit lijkt op een lineaire werkwijze, is dit niet het geval: het opstellen van het GDM en het GIM gebeurt iteratief. Na afronding van het GDM en GIM wordt de stap gemaakt naar het GOM. Tijdens het opstellen van de GOM kan besloten worden om het GDM en/of het GIM te herzien. Eventueel (maar dat is buiten scope van deze methode), zouden zelfs op basis van implementaties, bijvoorbeeld bij het testen, nog herzieningen kunnen volgen.



De basisstappen van de MOSES methode bevatten een aantal activiteiten waarin de daadwerkelijke modellen en specificaties tot stand komen. Uitvoering van de activiteiten binnen een basisstap is volledig iteratief en worden verder beschreven in navolgende paragraaf 4.3.

4.3 Begrippen en Technieken voor domeinanalyse

De MOSES methode kent in de gebruikte modellen een aantal begrippen :

1. *Actor* : een CNO-partij in een bepaalde rol.
2. *Object* : 'een ding' uit de werkelijkheid dat in de gemeenschappelijke bedrijfsvoering een of meerdere relevante rollen speelt.
3. *Gebeurtenis* : komt overeen met iets dat gebeurt in - of relevant is voor - het gemeenschappelijke domein.

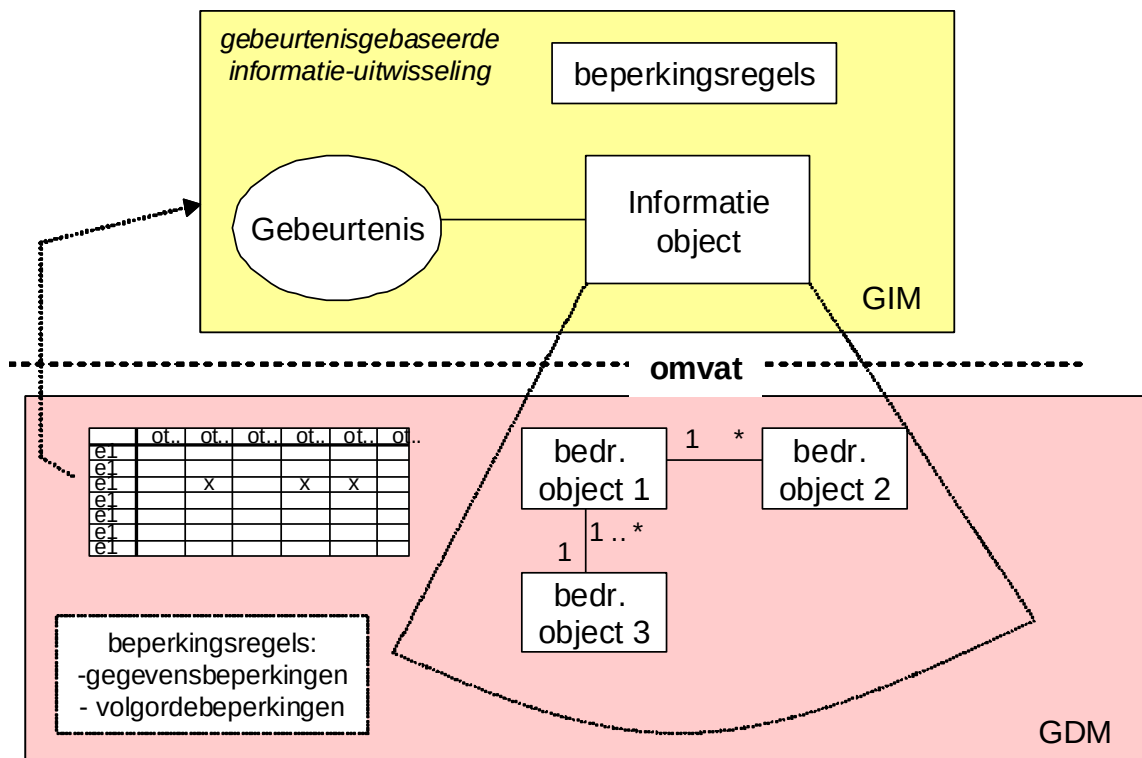
Een aanpak om domeinanalyse uit te voeren en deze begrippen te definiëren is:

1. Afbakenen van het domein en de relevante actoren.
2. Vinden van gebeurtenis- resp. objecttypen via situatiebeschrijvingen.
3. Beschouwen van de levenscyclus van objecttypen en de deelnemers aan een gebeurtenis.

Begrippen: actor, object, gebeurtenis, domeinanalyse

In het bedrijfsdomeinmodel geven we een abstracte en precieze representatie van de bedrijfsregels en het bedrijfsvocabulaire die in en rond het elektronische samenwerkingsverband gelden. Dat doen we aan de hand van de specificatie van de typen actoren, bedrijfsobjecten-, gebeurtenissen en aanvullende beperkingsregels die in dat probleemgebied van belang zijn.

In het GDM en GOM worden de verschillende aspecten van ons model vastgelegd. Onderstaande figuur geeft een overzicht van de verschillende aspecten die beschreven worden. Vervolgens zal elk van de onderdelen nader toegelicht worden, en zal ook ingegaan worden op de te hanteren technieken voor het opstellen van het model.



4.3.1 *Het begrip actor*

Actor Een actor is een CNO-partij in een bepaalde rol. Uitgangspunt hierbij is dat we in een CNO (Collaborative Networked Organisation, zie 3.1) individuele organisaties kennen die ten opzichte van elkaar meerdere rollen kunnen uitvoeren. Een organisatie kan in een CNO bijvoorbeeld zowel Afnemer als Leverancier zijn van bepaalde producten, met daarnaast bijvoorbeeld nog (geassocieerde) rollen als Crediteur en Debiteur. Het startpunt van de methode is dat we een overeengekomen en gedeeld beeld vaststellen van de verschillende deelnemende roltypen, de actoren.

4.3.2 *Het begrip object*

Object Een object is 'een ding' uit de werkelijkheid dat in de gemeenschappelijke bedrijfsvoering een of meerdere relevante rollen speelt. Een object:

- representeert een verschijnsel uit de werkelijkheid (bv: werknemer)
- kan worden beschreven met behulp van attributen en referenties (bv: leeftijd)
- heeft een identiteit (in MOSES is dit een expliciete unieke code, zie verder)
- bestaat voor een zekere periode (bv. 10 jaar, als werknemer)
- neemt deel in op zijn minst twee (creatie en terminatie) bedrijfsgebeurtenissen (bv. indiensttreden en uitdiensttreden)

Bedrijfsobjecten worden geabstraheerd tot Objecttypen, een soort van template voor het representeren van individuele objectinstanties. Een objecttype specificeert een externe, generieke en implementatieonafhankelijke kijk op de betekenis van een object. Bijvoorbeeld:

- Objecttype: Werknemer
- Bedrijfsobjecten (instanties): Jan, Piet, Marie.

4.3.3 *Het begrip gebeurtenis*

Gebeurtenis Een gebeurtenis treedt op in of rond de werkelijke gemeenschappelijke bedrijfsvoering. Een gebeurtenis:

- treedt op of wordt herkend op één moment in de tijd (en kent dus geen tijdsduur).
- komt overeen met iets dat gebeurt in - of relevant is voor - het gemeenschappelijke domein. Bijvoorbeeld: 'plaatsen order'. Als relevante gebeurtenissen beschouwen wij ook gebeurtenissen die (afgeschermd) intern bij een partij plaatsvinden maar waarover we in CNO mogelijk wel afspraken (voorwaarden) willen maken. Bijvoorbeeld: 'opmaken order'.
- is atomair en kan niet worden opgedeeld in sub-gebeurtenissen.

Net als objecten worden de gebeurtenissen ook geabstraheerd tot gebeurtenistypen, een soort template voor het beschrijven van individuele gebeurtenissen. Bijvoorbeeld:

- gebeurtenistype: kopen auto
- gebeurtenis: kopen_Ford_door_Jan_03/01/2010, kopen_Audi_Door_Marie_12/10/2010.

4.3.4 Een techniek voor domeinanalyse

Domeinanalyse

Het vastleggen van de betekenis van de begrippen volgt uit een domeinanalyse. Om deze domeinanalyse uit te voeren gaan we dus op zoek naar de actoren, gebeurtenistypen en objecttypen die binnen de reikwijdte van het beoogde CNO vallen. De zoektocht naar mogelijk relevante begrippen kan plaatsvinden op verschillende manieren. Als basis kan men bestaande situaties in het domein bestuderen. Men komt dan specifiek aanwijsbare of conceptuele instanties tegen, die een goede indruk geven van mogelijk relevante domeinverschijnselen. Een voorbeeld van deze actor- en kandidaat object- en gebeurtenistypebeschrijvingen vindt u in het volgende hoofdstuk **Fout! Verwijzingsbron niet gevonden..**

Een praktische voorbeeldaanpak voor domeinanalyse is de volgende:

1. We beginnen met een beschrijving van de relevante actoren in de CNO met hun wensen en beeld van de reikwijdte van het domein. Hierbij wordt bepaald of en welke (aanpalende) domeinen mee worden genomen in de beschouwing. Bijvoorbeeld: als uitgangspunt nemen we het domein van het bestellen van artikelen. Een vraag die dan o.m. opkomt is: is de financiële afhandeling rondom het bestellen van artikelen van belang? M.a.w., is de bank ook een actor, of alleen leverancier en afnemer?
2. Om kandidaat gebeurtenis- resp. objecttypen te vinden, is de extractie van werkwoorden en zelfstandige naamwoorden uit situatiebeschrijvingen een klassieke analysetechniek. Bijvoorbeeld: Leveranciers *leveren* de bestelde *artikelen* aan de Afnemers (leveren=gebeurtenis, artikel=objecttype). Een uitgebreider voorbeeld van deze analyse is beschreven in hoofdstuk **Fout! Verwijzingsbron niet gevonden..**
3. Bij het beschouwen van de levenscyclus van (kandidaat) objecttypen kan men ook op mogelijk (andere) relevante kandidaat gebeurtenistypen stuiten. Omdat de focus ligt op alles wat er met een objecttype gebeurt, vallen ontbrekende gebeurtenissen eerder op. Andersom kan een beschouwing van deelnemers in gebeurtenistypen duiden op mogelijk andere relevante objecttypen. Een voorbeeld van de laatste bewering: bij de gebeurtenis *samenstellen* van een *bestelling*, spelen *bestelregels* ook een rol.

4.4 Notatiewijzen

De MOSES methode hanteert een aantal notatiewijzen voor het vastleggen van de modellen:

1. Objectrelaties: een weergave van het statische, structurele gedeelte van het domeinmodel, m.b.v. objectrelatiediagrammen (UML klassendiagram).
2. Gebeurtenissen : een weergave van het dynamische gedeelte van het domeinmodel, m.b.v. een Actor- Object-Event-Table (AOET, uitbreiding op Merode).
3. Gebeurtenisvolgorde: een weergave van de volgordebependingen in het dynamische gedeelte van het domeinmodel, m.b.v. volgordediagrammen (Jackson).

Begrippen: objectrelatiediagram, AOET, volgordediagram

4.4.1 Een notatiewijze voor objectrelaties

Objectrelatie- diagram

De statische, structurele inter-objecttype aspecten worden gepresenteerd met behulp van een objectrelatiediagram. We gebruiken hiervoor de zogenaamde UML-klassendiagram notatie. Het

UML klassendiagram wordt hier verder als bekend verondersteld. Zie hoofdstuk **Fout!**
Verwijzingsbron niet gevonden. voor een voorbeeld.

We beschouwen daarbij de relaties tussen de objecttypen in overeenstemming met MERODE, d.w.z. als zogenaamde bestaansafhankelijke relaties⁵.

De actoren zien we als gewone objecttypen die dus - naast (re)actieve partijrollen - ook opgenomen worden als objecttype in het klassendiagram.

4.4.2 Een notatie voor gebeurtenissen en objectgebeurtenis relaties

Actor- Object-
Event-Table

De dynamische aspecten van de objecttypen worden gerepresenteerd in een tweetal diagrammen. Centraal in de specificatie van het GDM staat de zogenaamde A-OET, de Actor, Object, Event Table. De AOET is een uitbreiding van Merode's Object Event Table.

Een object-event table (OET) specificeert welke gebeurtenistypen de creatie, (mogelijke) modificatie en beëindiging van instanties van specifieke objecttypen veroorzaken. De OET is een tabel welke objecttypen en gebeurtenistype onderling associeert.

De uitbreiding A-OET bestaat uit het expliciet opnemen van actoren als:

- participerende objecttypen
- gebeurtenis veroorzakende partijen
- op gebeurtenis reagerende partijen

Deze uitbreiding met de actoren toont de rol die de actoren spelen bij de gebeurtenis. Een gebeurtenis kan intern plaatsvinden bij een individuele partij zonder dat deze gedeeld wordt met andere actoren. Deze gebeurtenissen nemen we wel op als ze relevant zijn voor de consistentiecontrole van het model. Bovendien kunnen er voorwaarden worden overeengekomen aangaande het optreden van de gebeurtenis. Het optreden van een gedeelde gebeurtenis kent altijd minimaal één actor die verantwoordelijk is voor het optreden van de gebeurtenis. Daarnaast kent de gedeelde gebeurtenis minimaal één actor die reactief deelneemt (en participeert) in de gebeurtenis. Een voorbeeld van een AOET is gegeven in paragraaf **Fout!**
Verwijzingsbron niet gevonden..

De objecten ontstaan niet zomaar. Instanties van de genoemde objecttypen worden altijd expliciet gecreëerd (C) en samengesteld alvorens ze eventueel participeren in opeenvolgende gebeurtenissen. Deze creatie vindt altijd plaats *Intern* bij een partij of meerdere partijen (d.w.z. achter de business-interface of end-point). De instanties kennen ook altijd een expliciete beëindiging *Intern* bij een of meerdere partijen (partij-eigen, achter de business interface, gelegen implementatie van afsluiting/archivering van bijvoorbeeld Bestelling). Indien er bij een beëindiginggebeurtenis (E, bv. afsluitenBestelling) in de tabel meerdere partijen als *Intern* staan aangemerkt dan bedoelen we hiermee dat al deze partijen een eigen, specifieke invulling van deze gebeurtenis kennen, die ze onder eigen voorwaarden en op eigen wijze invullen.

⁵ Als een objecttype A bestaansafhankelijk is van een objecttype B dan wil dit zeggen dat een instantie van A enkel een leven kan leiden binnen het leven van een - en altijd dezelfde - instantie van B. Met een leven leiden bedoelen we hier dat een object op een volgordelijke wijze deelneemt in de toegekende gebeurtenissen. Als deze volgorde relevant is dan geven we deze weer in de vorm van Jackson Volgordediagrammen of alternatief (zie verder). Als A bestaansafhankelijk is van B dan is de verzameling gebeurtenistypen waarin A deelneemt een deilverzameling van de verzameling gebeurtenistypen van B.

Alhoewel de betrokken gebeurtenissen niet direct zichtbaar zijn in het samenwerkingsverband en geen direct effect hebben over de bedrijfsgrens heen (d.w.z. in het gemeenschappelijke domein, tussen de business-interfaces) worden ze hier expliciet gemaakt zodat we verantwoordelijkheden kunnen toekennen en er eventuele voorwaarden voor uitvoering over kunnen afspreken.

Gemeenschappelijke, niet interne, gebeurtenissen (bv. plaatsenBestelling uit **Fout! Verwijzingsbron niet gevonden.**) kennen als participerende partijen altijd een partij die verantwoordelijk is voor het optreden van de gebeurtenis (V) en partij(en) die deelnemen op reactieve wijze (P). Er kunnen meerdere partijen zijn die verantwoordelijk zijn voor het optreden, d.w.z. dat verschillende partijen de initiator kunnen zijn van eenzelfde type gebeurtenis. Indien er op een rij van de tabel meerdere partijen als V staan aangemerkt dan bedoelen we hiermee dat voor elke instantie van de gebeurtenis er slechts één van deze partijen de initiator kan zijn (bv. annulerenBestelling uit **Fout! Verwijzingsbron niet gevonden.**).

4.4.3 Een notatie voor volgordebeperingen van gebeurtenissen

*Volgorde
diagram*

Volgordediagrammen zijn de tweede techniek die we gebruiken om dynamische aspecten van de objecttypen te specificeren. De volgordediagrammen worden gebruikt om volgordebeperingen tussen het optreden van gebeurtenissen te specificeren. Gebeurtenissen kunnen niet op elk willekeurig moment optreden, maar kennen een zekere volgorde. Een levering kan bijvoorbeeld pas plaatsvinden nadat er een order is geplaatst.

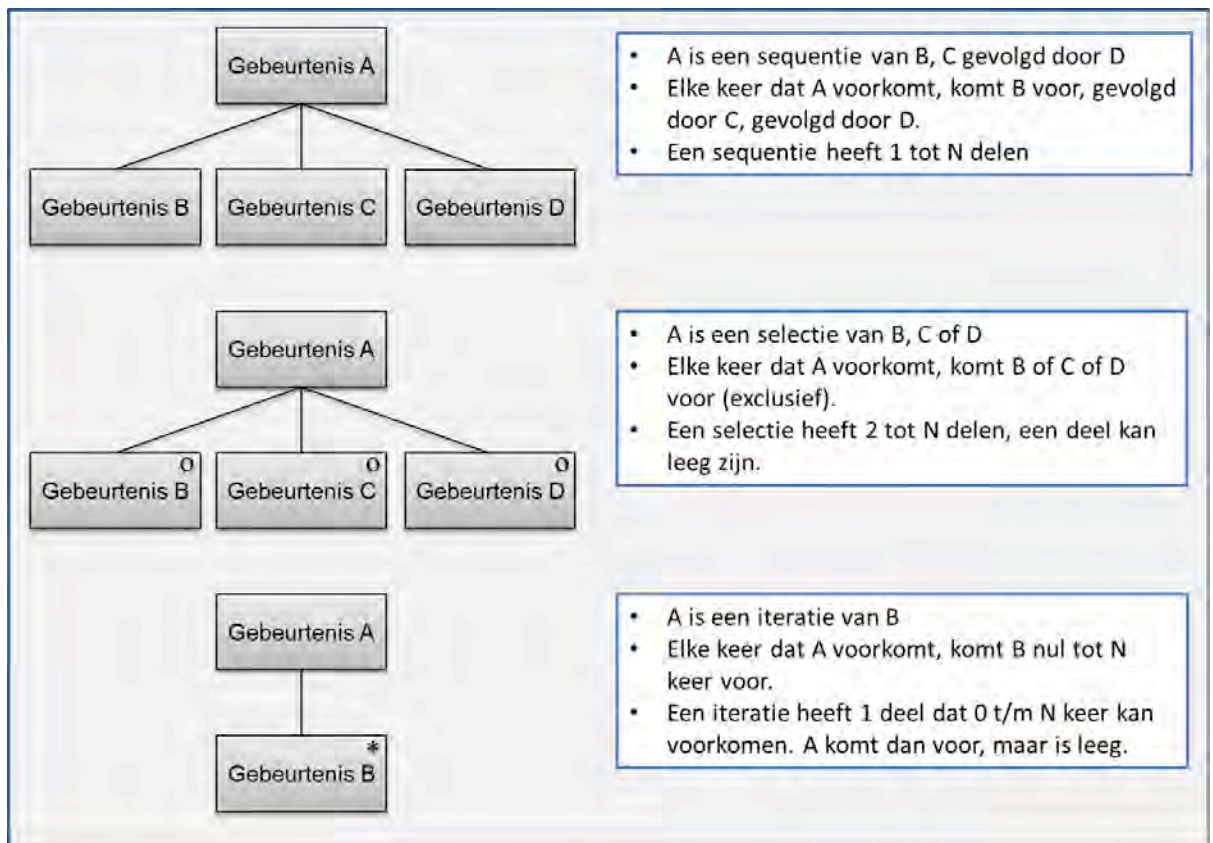
In het voorafgaande is al beschreven dat de verschillende partijen (in hun eigen administraties) een eigen kijk hebben op status, historie en gedrag van de verschillende objecten. We bekijken de processen die plaatsvinden op de objecttypen dan ook vanuit het perspectief van de afzonderlijke partijen. Voor elke partijrol ontwikkelen we een of meer volgordediagrammen per relevant objecttype. Voorbeeld: voor zowel afnemer als leverancier bestaat het begrip bestelling. De activiteiten van de afnemer (b.v. plaatsen) en leverancier (b.v. toekennen artikelen) zijn echter niet dezelfde.

We beschouwen alle zowel impliciete als expliciet gemaakte objectlevenslopen als simultane (concurrente) processen die zich afspelen bij de verschillende partijen met hun eigen taken en diensten. In de tijd kennen deze processen synchronisatiepunten in de gemeenschappelijke, gedeelde gebeurtenissen (b.v. annuleren bestelling). De processen bij elkaar geven een compleet beeld van de volgorde van de behandeling van de objecttypen. We beschouwen de processen als de specificatie van volgordebeperingen welke aangeven onder welke condities gebeurtenissen kunnen plaatsvinden in de tijd.

*Jackson's
volgorde
diagram*

In de AOET zien we al een aantal triviale volgorden uitgedrukt, de AOET toont immers de creatie, mutatie en beëindigings-gebeurtenissen van objecttypen. De volgorde creatie-mutatie(s)-beëindiging is dus vastgelegd. In veel situaties moeten echter meer stringente beperkingen geformuleerd worden ten aanzien van de mutaties, binnen de context van de in de gebeurtenissen participerende objecttypen. Voorbeeld: een bestelling (objecttype) kan pas geannuleerd worden (mutatie), nadat de bestelling is geplaatst (andere mutatie). Voor deze zogenaamde levensloopspecificatie van objecttypen en het vastleggen van volgorde van gebeurtenissen, gebruiken we Jackson's Volgorde Diagrammen.

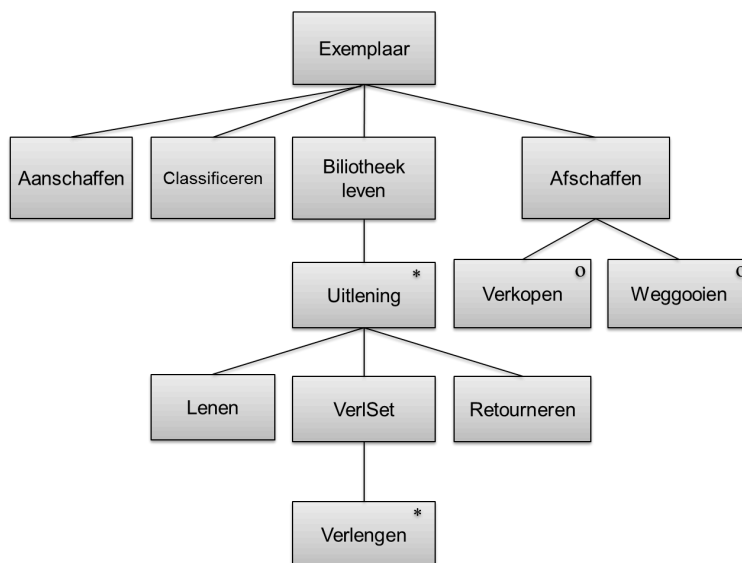
Een volgordediagram kan bestaan uit drie soorten componenten (plus het elementaire component, nl. het blad aan de boom) als volgt:



Figuur 9: Volgorde diagramnotaties volgens Jackson.

Voorbeeld:

een boek - uitleenexemplaar van een bepaalde titel - in een bibliotheek:



Figuur 10: Voorbeeld volgordediagram, een bibliotheekboek

- Een (boek)*Exemplaar* begint zijn leven met het *Aanschaffen* (start-/creatiegebeurtenis)
- Nadat het boek is aangeschaft wordt het boek geclassificeerd (*Classificeren*) en daarmee impliciet opgenomen in de bibliotheek.
- Het boek begint zijn echte “uitleenleven” in de bibliotheek. Na classificatie ondergaat het exemplaar nul tot N maal een Uitleningscyclus die begint met *Lenen* en eindigt met *Retourneren*.
- Na lening kan de uitlening nog een aantal keer verlengd (*Verlengen*) worden.
- Aan het eind van het bibliotheekleven van het exemplaar wordt het boek afgeschaft (Afschaffen, terminatiegebeurtenis) door *Verkopen* of *Weggooien*.

Meer voorbeelden zijn terug te vinden in **Fout! Verwijzingsbron niet gevonden..**

Als mogelijk (UML) alternatief kunnen State Transition Diagrams of Harel Statecharts gebruikt worden.

4.5 Een laatste opmerking rondom het GDM

Aan het eind van dit hoofdstuk willen we nog een belangrijke opmerking maken over de consistente en complete behandeling van elk opgenomen gebeurtenis- en objecttype zoals beschreven in MERODE.

In het GDM streven we niet naar een compleet beeld van elke atomaire gebeurtenis in het model zoals gehanteerd door Merode. Niet elk objecttype kent een expliciet en formeel uitgewerkte creatie of beëindiginggebeurtenis en bijbehorende specificaties van object-event-participaties (contracten in de vorm van pre- en postcondities aangaande het gedrag van een object in gebeurtenis). Het toevoegen van Orderregels aan Orders (bijvoorbeeld creatieOrderRegel als onderdeel van samenstelling Order) kent alleen relevantie voor de samenstellende partij (de Afnemer). Het eerste gedrag wat we zien en waarover we gemeenschappelijk beeld willen vastleggen is de toekenning van producten aan de verschillende Orderregels. Dit wordt uitgevoerd door de Leverancier maar is van belang om gedrag rond toekenning van artikelen aan de Order te kunnen vastleggen. Wat zijn de voorwaarden voor het toekennen van een vervangend product aan een orderregel bijvoorbeeld.

4.5.1 De techniek om van een GDM een GIM te maken

Zoals eerder aangegeven beschrijft het Gedeeld Bedrijfsinformatie Model (GIM) de relevante informatie van bedrijfsobjecten uit het GDM. Deze informatie wordt uitgewisseld bij het optreden van gemeenschappelijke gebeurtenissen uit het GDM. Bij het beschrijven van het GIM gebruiken we dezelfde notaties die ook gebruikt zijn bij het GDM.

Het delen en uitwisselen van informatie tussen de actoren aangaande de bedrijfsobjecten uit het GDM kan op een aantal verschillende wijzen plaatsvinden:

- Het optreden van een gebeurtenis bij één van de participerende actoren, die vervolgens de andere actoren informeert
- In de vorm van een vraag-antwoord dialoog, bijvoorbeeld voor een status-inspectie.

De eerste vorm is direct herleidbaar tot hetgeen in het GDM aan gebeurtenisparticipaties is gespecificeerd. Voor de vraag-antwoord dialogen geven we hier voor elk gebeurtenistype uit het

GDM een herstructurering van de gebeurtenisparameters waarbij we nu ook verzamelingen (Sets, Lists e.d.) opnemen als onderdeel van het informatieobject dat de gegevensinhoud van de gebeurtenis beschrijft. De vraag-antwoord dialogen zijn niet direct herleidbaar uit het GDM, maar zullen bepaald moeten worden in overleg met domein-experts.

Een eerste versie van het GIM kan afgeleid worden uit het GDM. Hiervoor gelden de volgende afleidingsregels:

- Elke gebeurtenis waarin meerdere partijen deelnemen (gedeelde gebeurtenis) is relevant voor het Bedrijfsinformatiemodel. Deze worden visueel gemaakt door een dikke streep te trekken in de AOE tabel waarbij alle gezamenlijke gebeurtenissen onder de streep staan.
- Voor elke gedeelde gebeurtenis wordt een formulier opgenomen in het GIM.
- In elk formulier worden gegevens opgenomen van het object dat participeert in de gebeurtenis (het basisobject⁶), al zijn attributen en relaties (de objecten waar hij bestaansafhankelijk van is). Daarnaast wordt mogelijk een lijst met objecten opgenomen die bestaansafhankelijk zijn van hem (bijvoorbeeld een lijst met artikelbestellingen binnen bestelling).

Let wel, soms kan volstaan worden met slechts een referentie (sleutel, uniek ID) vanuit een object naar het basisobject, in plaats van het hele object "mee te sturen" met al zijn afhankelijkheden. Dit is het geval indien de status van deze objecten (ook attribuutwaarden en referenties) eerder uitgewisseld, en ondertussen niet gewijzigd zijn.

Vervolgens moet in het GIM de objecten uit het GOM nader uitgewerkt worden. Eigenschappen (attributen) en invarianten (condities) worden vastgelegd. Gezamenlijke gebeurtenissen worden verder uitgewerkt in het BedrijfsInformatieModel. Hiervan leggen we de parameters vast (informatie in het bijhorende formulier, zie hierboven), pre- en postcondities.

Hoofdstuk **Fout! Verwijzingsbron niet gevonden.** zal nader ingaan op de afleiding vanuit het GDM naar het GIM.

⁶ Basisobject is het meest bestaansafhankelijke object binnen het optreden van het event.

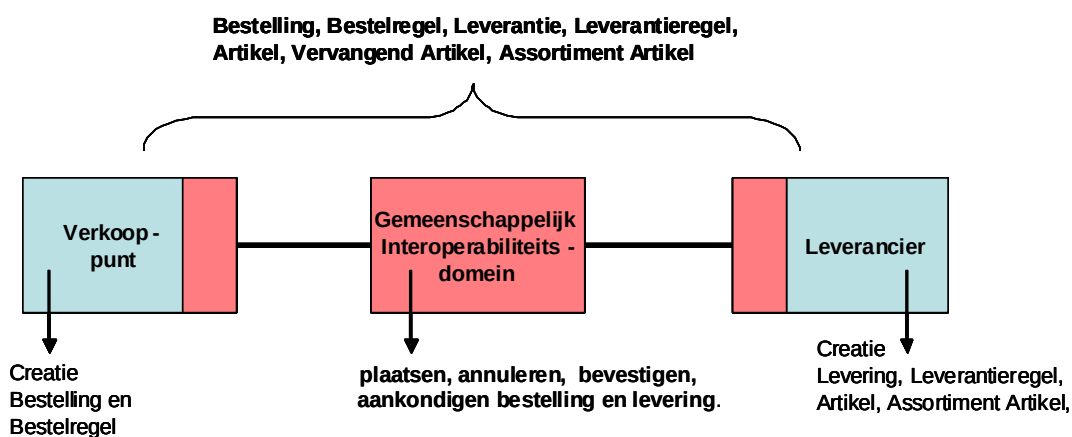
5 De methode toegepast

In dit hoofdstuk beschrijven we de MOSES methode aan de hand van een eenvoudig voorbeeld. We bekijken respectievelijk het Bedrijfsdomeinmodel en Bedrijfsinformatiemodel met hun onderdelen. Het model handelt over een algemeen, vereenvoudigd “Leverancier-Verkooppunt-Netwerk” in de juweliersbranche. Hierbij dient het ontwikkelde model de gemeenschappelijke informatievoorziening en transacties rond het bestellen en leveren van artikelen te ondersteunen.

5.1 Achtergrond en Context - Vereenvoudigd Bestellen en Leveren in de Juweliers Branche

In de branche kennen we de partijrollen Afnemer (of Verkooppunt) en Leverancier. Afnemers plaatsen bestellingen bij Leveranciers. Leveranciers leveren de bestelde artikelen aan de Afnemers. Ten behoeve van het bestellen heeft een Afnemer inzage in het assortiment van een leverancier. Bestellingen zijn samengesteld uit bestelregels waarin per regel een hoeveelheid artikelen van een gespecificeerd type worden besteld. Leveranciers kunnen - indien de bestelde artikelen niet leverbaar zijn - vervangende artikelen toekennen aan een bestelling. Een bestelling wordt bevestigd met een overzicht van geaccepteerde, geannuleerde, gewijzigde en vervangende artikelregels. Een levering is samengesteld uit leverantieregels waarin per regel een te leveren artikeltype wordt gespecificeerd met bijbehorende hoeveelheid. Een levering wordt vooraangekondigd voordat er geleverd wordt. Een levering kan deelleveringen uit meerdere onderliggende bestellingen bevatten. (Deel)Bestellingen en leveranties kunnen geannuleerd worden, wijzigingen in bestellingen, leveranties en regels doen zich in deze vereenvoudigde wereld niet voor⁷.

Een eerste analyse van de tekst geeft reeds een gevoel voor reikwijdte en relevante termen. De blauwe (cursief) woorden duiden op een gebeurtenis (werkwoord - vervoeging) en de groene (onderstreepte) woorden duiden op een object (zelfstandig naamwoord – verbuiging). We kunnen hieruit de eerste omgevingscontouren (contextdiagram, ecosysteem) schetsen.



Figuur 11: Contextdiagram Juweliersbranche

⁷ Het opnemen van 'wijzigingen in bestellingen, leveranties en geassocieerde bestel- en leverantieregels' voegt een aanzienlijk aantal gebeurtenissen en beperkingsregels toe die het model (omvang)rijker maken, maar ten behoeve van de uitleg van de methode het e.e.a. onnodig compliceren en vanuit methodisch perspectief weinig toevoegen.

5.2 Het Gemeenschappelijke Bedrijfsdomeinmodel (GDM)

Zoals beschreven in hoofdstuk 4 geven we in het bedrijfsdomeinmodel een abstracte maar precieze representatie van de bedrijfsregels en het bedrijfsvocabulaire welke in en rond het elektronische samenwerkingsverband gelden. Dat doen we aan de hand van de specificatie van bedrijfsobjecten-, gebeurtenissen en beperkingsregels.

Na interview- en modelleringssessies met belanghebbenden en verdere bestudering van het bovenstaande op relevantie en conformatie aan modelleringsregels benoemen we in onderstaande tabellen de als binnen de reikwijdte vallende, relevante, geselecteerde begrippen.

De eerste tabel beschrijft de betrokken partijrollen. Een partijrol (actor) is een classificatie van een verzameling organisaties die dezelfde rol innemen in het samenwerkingsverband.

Partijrol (Actor)	Beschrijving	Wensen en Eisen
Afnehmer	Partij welke producten bestelt en afneemt.	Bestellen vanuit kassasysteem, inzicht in assortiment leveranciers.
Leverancier	Partij welke bestellingen voor producten aanneemt en levert.	Inzicht in verkoopgegevens afnemers, eenvoudig catalogus aanleveren aan afnemers

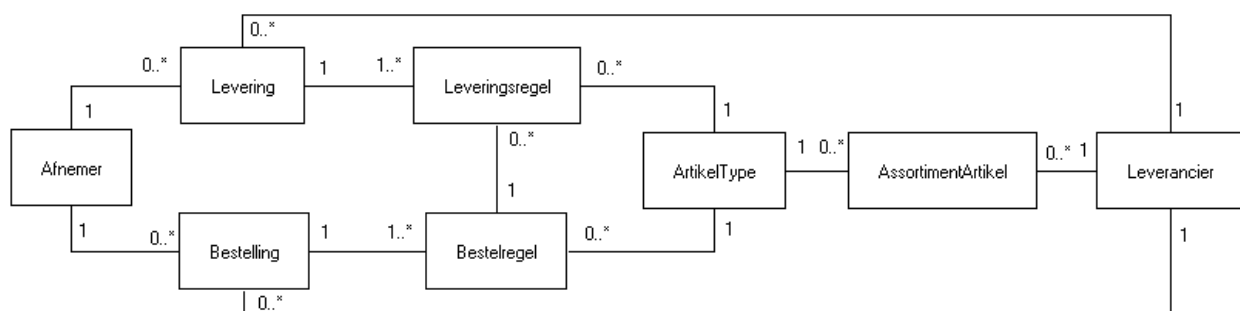
Na het beschrijven van de relevante partijrollen wordt een overzicht gegeven van de van belang zijnde bedrijfsgebeurtenistypen. Dit zijn classificaties van gebeurtenissen die optreden in het samenwerkingsverband en relevant zijn voor de verschillende organisaties.

Bedrijfsgebeurtenistype	Beschrijving	Opmerkingen
plaatsenBestelling	Het doorgeven van een bestelling met artikelen die eerder is gecreëerd.	Een bestelling bestaat uit een aantal regels waarop artikelen zijn opgenomen inclusief het gewenste aantal.
bevestigenBestelling	Het bevestigen van de geplaatste bestelling inclusief indicatie van levertijden.	Een leverancier stuurt altijd een bevestiging naar de afnemer van een bestelling. De afnemer weet hiermee ook wanneer de bestelde artikelen geleverd kunnen worden.
annulerenBestelling	Het annuleren van de bestelling bij een leverancier door de afnemer.	Een bestelling kan alleen volledig geannuleerd wordt.
vooraankondigenLevering	Het aankondigen aan de afnemer door de leverancier van de levering van de bestelde producten.	Na de vooraankondiging wordt er altijd geleverd, het annuleren van de bestelling is dan niet meer mogelijk.
annulerenLevering	Het annuleren van een levering aan een afnemer door de leverancier.	Een levering kan niet meer geannuleerd worden als deze vooraangekondigd is.

Bedrijfsobjecttype	Beschrijving	Opmerkingen
Bestelling	Verzoek (opdracht) tot het leveren van een of meerdere artikelen van dezelfde dan wel verschillende soorten.	
Bestelregel	Verzoek tot levering van een hoeveelheid van één specifiek artikeltype als onderdeel van een Bestelling	
Levering	Object dat de fysieke levering van producten representeert. Een zending van Artikelen.	
Leveringsregel	Onderdeel van een levering van een bepaalde hoeveelheid van een specifiek Artikeltype. Artikelspecifiek deel van een geconsolideerde zending.	
ArtikelType	Soort artikel dat in de juweliersbranche verhandeld wordt. (eg. horloge 123)	
AssortimentArtikel	Artikel dat is opgenomen in het assortiment van een leverancier en dus geleverd kan worden	Niet alle mogelijk artikelen maken onderdeel uit van het assortiment van de leverancier, alleen assortimentsartikelen zijn daadwerkelijk te bestellen.

In het onderstaande klassendiagram (EDG) representeren we de onderlinge relaties (bestaansafhankelijkheid) tussen de benoemde objecttypen en in de daarop volgende AOE-tabel representeren we de relaties tussen de objecttypen en de gebeurtenistypen. Deze twee modelperspectieven worden simultaan bestudeerd en opgesteld.

5.2.1 Klassendiagram - interobjecttype afhankelijkheden:



Eén Bestelling kent altijd één Afnemer en één Leverancier. Een Bestelling bestaat uit één of meerdere Bestelregel(s) waarbij een Bestelregel altijd bij één en dezelfde Bestelling behoort. Een Bestelregel refereert altijd aan één - in die regel besteld - Artikeltype (dus niet per definitie aan een artikel uit het assortiment van de betreffende leverancier - AssortimentArtikel). Eén

Levering kent altijd één Afnemer en één Leverancier. Een Levering (Zending) bestaat uit één of meerdere Leveringsregel(s) (Levering van specifiek artikel(hoeveelheid) als onderdeel van de zending). Een Leveringsregel behoort altijd bij één en dezelfde Levering en handelt altijd over het Artikeltype dat geleverd wordt en de Bestelregel (en via de Bestelregel de Bestelling) die ten grondslag ligt aan de levering van het artikel. Eén Bestelregel kan meerdere Leveringsregels tot gevolg hebben, d.w.z. deelleveringen van bepaalde hoeveelheid besteld (of vervangend) artikel. Het artikel waaraan in de leveringsregel gerefereerd wordt kan overeenkomen met het in onderliggende Bestelregel genoemde artikeltype of kan refereren aan een ander artikel. In het laatste geval is er sprake van een "Vervangend Artikel" welke niet expliciet is benoemd in het model maar afgeleid kan worden door vergelijk van Artikeltypen in Bestel- en geassocieerde Leveringsregels. Bovendien toont dit modelperspectief niet de voorwaarden waaronder een vervangend artikel (mag) worden toegekend (dit laatste zien we terug in de precondities van de gebeurtenissenspecificaties).

5.2.2 AOE_Tabel (Actor - Object - Event) - Participaties van Partijrollen en Objecttypen in Bedrijfsgebeurtenissen:

Naast boven besproken statische structuren gebeurt er van alles in en rond het gemeenschappelijk domein dat we nader dienen te bestuderen. Een belangrijk perspectief op de dynamica in het domein wordt hierbij gepresenteerd m.b.v. onderstaande AOE-Tabel. We zien hier boven geïdentificeerde partijrollen, bedrijfsobjecten en bedrijfsgebeurtenissen terug in onderlinge samenhang. De tabel toont de (soorten) participaties van de objecten in de gebeurtenissen. Iedere gevulde cel geeft een participatie van het betreffende object in betreffende gebeurtenis weer.

	Afnemer	Leverancier	ArtikelType	Bestelling	Bestelregel	Levering	Leveringsregel	
aanmakenBestelling	I			O/C				Partij - interne - voorbereiding, coördinatie en afhandeling
afsluitenBestelling	I	I		O/E				
opnemenBestelregel	I		A/M	A/M	O/C			
afsluitenBestelregel	I	I	A/M	A/M	O/E			
aanmakenLevering		I				O/C		
afsluitenLevering	I	I				O/E		
toekennenBesteldArtikel		I	A/M	A/M	A/M	A/M	O/C	
toekennenVervangendArtikel		I	A/M, A/M	A/M	A/M	A/M	O/C	
afsluitenLeveringsregel		I	A/M	A/M	A/M	A/M	O/E	
verzenden		I				O/M		
plaatsenBestelling	V	P		O/M				Gemeenschappelijke gebeurtenissen voor 2 (of meer) partijen
bevestigenBestelling	P	V		O/M				
annulerenBestelling	V/P	P/V		O/M				
vooraankondigenLevering	P	V				O/M		
annulerenLevering	P	V				O/M		
annulerenLeveringsregel	P	V	A/M	A/M	A/M	A/M	O/M	

Samenwerkende partij(rollen) deelnames en verantwoordelijkheden

legenda:

I = Partij-intern optreden en afhandeling van niet gedeelde gebeurtenis.

V = Verantwoordelijke partij voor optreden en inhoud gedeelde gebeurtenis

P = Participerende partij in optredende gebeurtenis

C = Creatie van een exemplaar van het betreffende objecttype

E = Beëindiging van een exemplaar van het betreffende objecttype

M = Mutatie van een exemplaar van het betreffende objecttype

O = Basisdeelname (Owned) deelname in gebeurtenis door meest bestaansafhankelijk objecttype voor gebeurtenistype

A = Verkregen (Acquired) deelname in gebeurtenis door propagatie van betreffende gebeurtenistype (optioneel)

We zullen in het kader van dit document niet iedere regel en kolom in deze tabel bespreken. Ter verduidelijking zullen we een drietal voorbeelden geven van de wijze waarop wij de tabel en daarin gemaakte keuzes interpreteren.

- *Voorbeeld 1, gebeurtenis plaatsenBestelling*

In één verschijningsvorm (optreden) van *plaatsenBestelling* participeert één instantie (exemplaar) van *Afnemer*, één instantie van *Leverancier* en één instantie van *Bestelling*. Afnemer is de verantwoordelijke (actieve) partij (V) voor het plaatsen van de bestelling en Leverancier de participerende (reactieve) partij (P). Het plaatsen van de bestelling verandert (M) de status van de reeds gecreëerde bestelling. Bestelling is het basisobjecttype (O, Owner) voor deze gebeurtenis. Bestelling is het meest bestaansafhankelijke object dat deelneemt in de gebeurtenis en is daarom het basisobjecttype (O, Owner) voor deze gebeurtenis. De partijen (Afnemer en Leverancier) nemen deel via propagatie van de gebeurtenis.

- *Voorbeeld 2, toekennenVervangendArtikel*

In één verschijningsvorm van *toekennenVervangendArtikel* participeert één instantie van *Leverancier* (I) en wordt één instantie van Leveringsregel gecreëerd (C). Bovendien neemt de (reeds bestaande) Leverantie (M), waar de Leveringsregel onderdeel van gaat uitmaken, deel in de gebeurtenis evenals de Bestelregel (M) en achterliggende Bestelling (M) waar de Leveringsregel uit voorkomt. Er nemen twee instanties van ArtikelType deel in de gebeurtenis, nl. het originele bestelde artikel (via Bestelregel) en het afwijkende, vervangende artikel dat daadwerkelijk geleverd gaat worden (M). Afnemer neemt niet deel aan de gebeurtenis, de gebeurtenis vindt plaats binnen de interne bedrijfsvoering van de Leverancier (I). Dit laatste wil niet zeggen dat de gebeurtenis niet relevant is voor het formaliseren van de samenwerkingsrelatie, immers we kunnen voorwaarden overeenkomen op basis waarvan een vervangend product mag worden toegekend (bv. ArtikelType.prijscategorie = Laag en Bestelregel. vervangingToegestaan = Waar). De detailspecificaties van de ArtikelType- en Bestelregel-participaties in toekennenVervangendArtikel representeren de plaatsen om deze voorwaarden te specificeren. De gecreëerde Leveringsregel is het basisobjecttype (O) voor deze gebeurtenis de andere deelnemingen zijn verkregen (A, Acquired) door propagatie van de gebeurtenis vanuit het basisobjecttype.

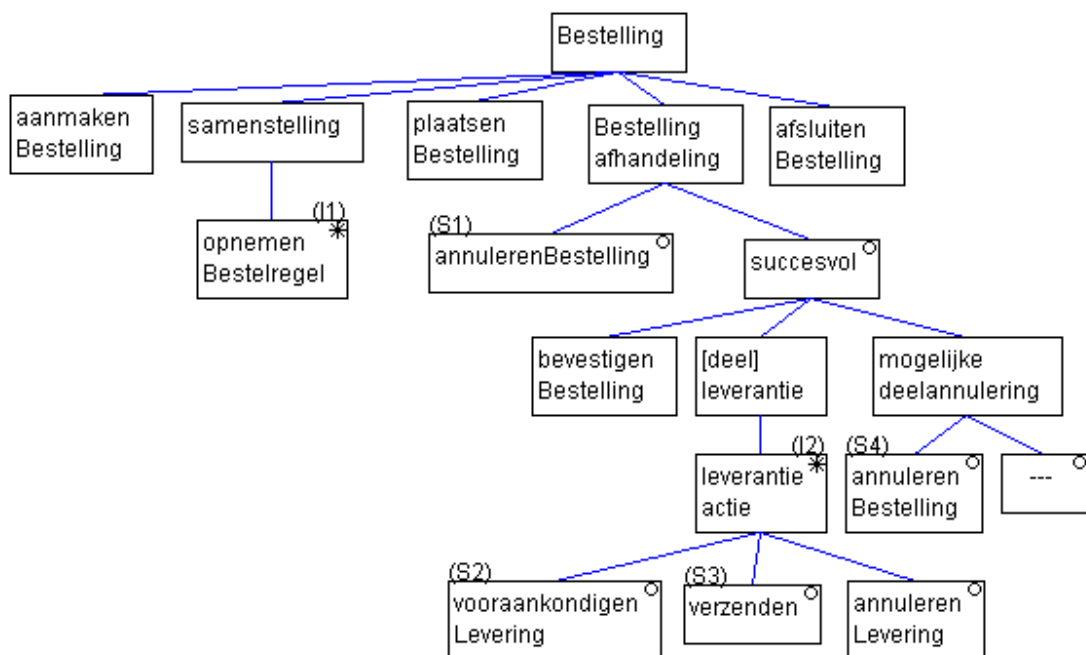
- *Voorbeeld 3, afsluitenBestelregel*

Het afsluiten van een Bestelregel betekent het in een eindstatus (E) brengen van de Bestelregel betreffende een bepaald ArtikelType (M) in een Bestelling (M). Er gebeurt hierna niets meer met een Bestelregel die relevant is voor het gemeenschappelijke domein. Afnemer (I) en Leverancier (I) kennen ieder een eigen interne realisatie van het afsluiten van de regel maar komen overeen wat het afsluiten van een Bestelregel in het gemeenschappelijke domein betekent, onder welke voorwaarden dit een geldige actie is en wat het gevolg van de actie is. Bestelregel is het basisobjecttype voor de gebeurtenis (O), de overige deelnames zijn verkregen (A) door propagatie.

5.2.3 Bestel- en Leverantieprocessen in de keten en de levenscyclus van de objecttypen

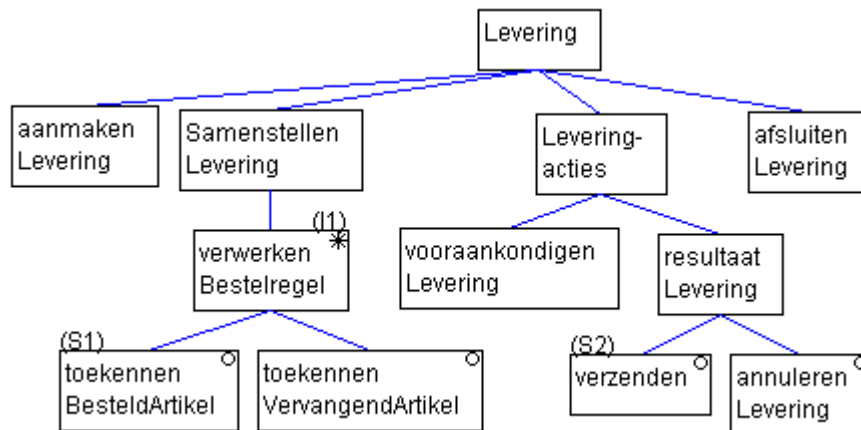
De verschillende actoren hebben in hun eigen administraties een eigen kijk op status, historie en gedrag van de verschillende objecten en realisatie van de gebeurtenissen. In de ontwikkeling van de standaard echter dienen de partijen overeenstemming te bereiken over betekenis, voorwaarden en gevolgen van de gebeurtenissen in het gemeenschappelijk domein. In het gemeenschappelijke domein kennen ze een gedeelde modelstatus. In het navolgende tonen we achtereenvolgens de levenscycli van Bestelling en Levering. Uit deze volgordes destilleren we de mogelijke waarden voor de 'statussen van de objecttypen' die gebruikt worden bij de formulering van beperkingsregels (voorwaarden voor geldigheid van optreden gebeurtenis).

Volgordelijke multi-party afhandeling van Bestelling:



aanmakenBestelling:	Bestellingstatus = "voorbereiding"
plaatsenBestelling:	Bestellingstatus = "geplaatst"
bevestigenBestelling:	Bestellingstatus = "bevestigd"
annulerenBestelling:	Bestellingstatus = "geannuleerd"

Volgordelijke multi-party afhandeling van Levering:



aanmakenLevering: Leveringsstatus = "inVoorbereiding"
 vooraankondigenLevering: Leveringsstatus = "vooraangekondigd"
 verzenden: Leveringsstatus = "verzonden"
 annulerenLevering: Leveringsstatus = "geannuleerd"

5.2.4 Detaillering GDM:

Onderstaand geven we van een aantal kernobjecten een voorbeeld van verdere detaillering door de opname van attributen. De detaillering van de gebeurtenissen hebben we hier niet opgenomen maar vindt plaats bij het opstellen van het GIM.

Objecttype: Bestelling

```

// referenties
afnemer : Afnemer_Ref                               // afkomstig van Fout! Verwijzingsbron
niet gevonden.
leverancier : Leverancier_Ref                         // afkomstig van Fout! Verwijzingsbron
niet gevonden.

// attributen
status : Bestellingstatus                            // afkomstig van Fout! Verwijzingsbron niet gevonden.
bestelnr : Number
opmerkingen : Text
plaatsings_tsk : TimeDate
uiterste_ontvangst_tsk : TimeDate

// invarianten
notNull [afnemer]
notNull [leverancier]
notNull [bestelnr]
notNull [plaatsings_tsk]
unique [afnemer, leverancier, bestelnr]              // combinatie is uniek ter identificatie
notNull [plaatsings_tsk] & plaatsings_tsk >= CurrentTimeDate
uiterste_ontvangst_tsk >= plaatsings_tsk
  
```

Objecttype: Bestelregel

// referenties

bestelling : Bestelling_Ref

artikel : Artikel_Ref

//attributen

status : Bestelregelstatus

regelnr : Number

gewenste_hoeveelheid : Number

vervanging_toegestaan : Boolean // default: true

opmerkingen : Text

// Invariant

notNull [bestelling]

notNull [artikel]

notNull [gewenste_hoeveelheid]

notNull [regelnr]

unique{bestelling, regelnr}

Objecttype: Levering

// referenties

afnemer : Afnemer_Ref

leverancier : Leverancier_Ref

// attributen

status : Leveringstatus leveringnr : Number

opmerkingen : Text

geplande_levering_tsk : TimeDate

//invarianten

Objecttype: Leveringsregel

// referenties

levering : Leveringsref_Ref

bestelregel : Bestelregel_Ref

artikel : Artikel_Ref

//attributen

status : Leveringsregelstatus regelnr : Number

hoeveelheid : Number

opmerkingen : Tekst

// Invarianten

5.3 GIM - Vereenvoudigd Juweliersbranche Informatiemodel

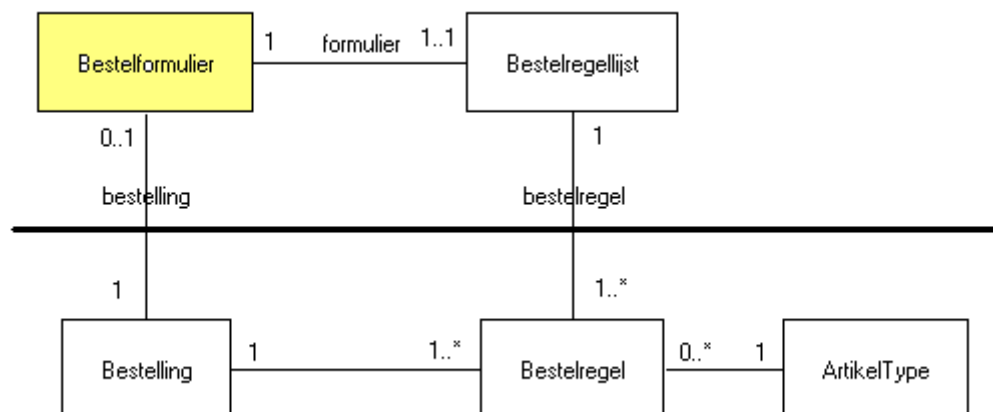
In dit hoofdstuk leiden we het GIM af uit het GDM zoals beschreven in hoofdstuk 5.2.

Als eerste beschouwen we de relevante, gemeenschappelijke *gebeurtenisgebonden informatieobjecten* (zie de gebeurtenissen 'onder de streep' uit de AOET van **Fout! Verwijzingsbron niet gevonden.** en de grijs gekleurde gebeurtenissen in de volgordediagrammen van **Fout! Verwijzingsbron niet gevonden.**). Voor elke gebeurtenis beschrijven we een geconsolideerde 'view' (een deelverzameling van relevante objecten en relaties) uit het GDM. Vervolgens mappen we het *informatieobject* (geel gemarkeerd) met zijn onderdelen op het GDM. Hierbij benoemen we de voor het formulier relevante referenties en attributen in termen van de onderliggende domeinbegrippen. Bovendien kan het hier nodig zijn om condities voor iteraties (lijsten) en selectiecriteria te bepalen. Tot slot specificeren we bij de gebeurtenis behorende, uit het GDM afgeleide *voorwaarden* af voor geldigheid van optreden, inhoud en gevolg.

Nadat we de gebeurtenisgebonden informatieobjecten hebben behandeld beschrijven we een voorbeeld van een vraag-gebonden informatieobject. Dit object beschrijft de informatie-inhoud voor een eenvoudige vraag/antwoord dialoog (aanvragenCatalogus)

5.3.1 Gebeurtenis: *plaatsenBestelling* en het Informatieobject: **Bestelformulier**

Domein-View⁸:



⁸ In dit voorbeeld duiden we eenmalig de afbeeldingswijze van GDM-gegevenstypen naar GIM-gegevenstypen overeenkomstig **Fout! Verwijzingsbron niet gevonden.** We presenteren de afbeelding als letters in de Domein-View die we vervolgens teruzien naast de attributen van het informatieobject.

Informatieobject: *Bestelformulier*

```

Afnemer : Afnemer_Ref           // afkomstig van Bestelling
Leverancier : Leverancier_Ref   //  ,,      ,,      ,,
bestelNr : Number               //  ,,      ,,      ,,
plaatsingsdatum : TimeDate      //  ,,      ,,      ,,
opmerkingen : Text              //  ,,      ,,      ,,
uiterste_ontvangst_tsk : TimeDate //  ,,      ,,      ,,
Bestelregellijst : List [Bestelregel]
  [ Bestelregel:
    Bestelregel.bestelNr : Number
    Bestelregel.regelNr : Number
    Bestelregel.gewenste_hoeveelheid : Number
    Bestelregel.vervanging_toegestaan : Boolean // default: true
    Bestelregel.opmerkingen : Text
    Bestelregel.artikel : ArtikelRef ]

```

Precondities

```

notNull [Afnemer]
notNull [Leverancier]
notNull [bestelNr]
unique [Afnemer, Leverancier, bestelNr] // combinatie is uniek - vergelijk 'compound keys'

```

Bestellingstatus= 'voorbereiding' // afkomstig van basisobjectstatus - b -

(uiterste_ontvangst_tsk = empty) or (uiterste_ontvangst_tsk >= plaatsingsdatum)

```

notEmpty [ Bestelregellijst ]
forEach [ Bestelregel] : Bestelregel.bestelNr = bestelNr
                       : unique [Afnemer, Leverancier, bestelNr, Bestelregel.regelNr]
                       : Bestelregel.gewenste_hoeveelheid >= 1
                       : notEmpty [Bestelregel.Artikel_Ref]

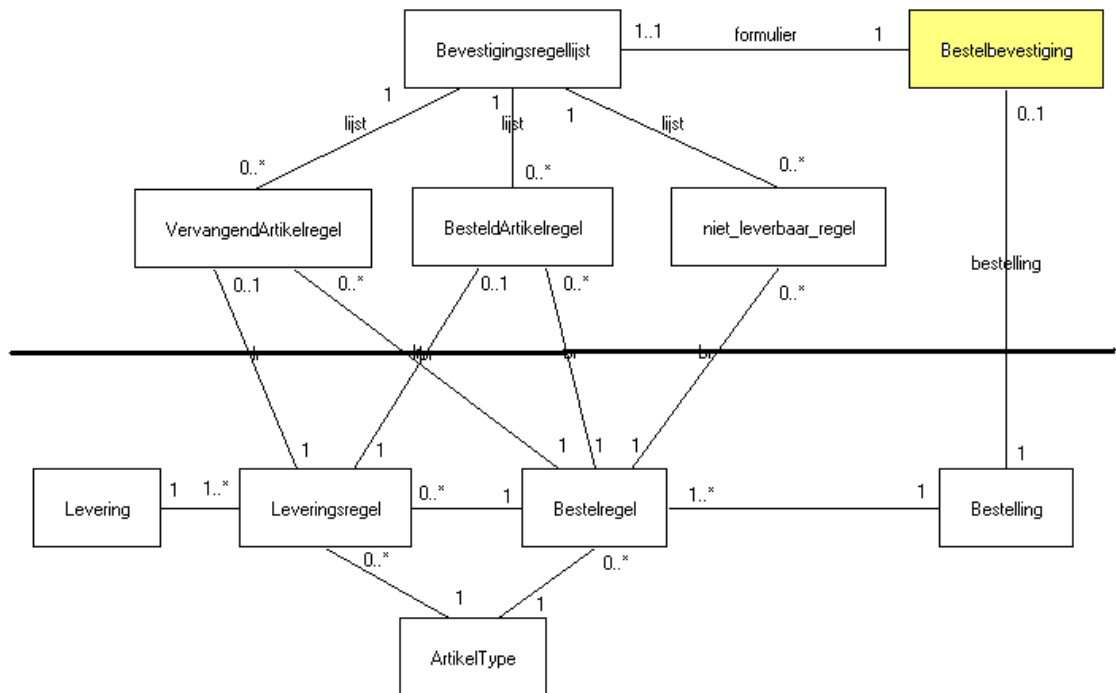
```

Postcondities (voor zover status bepaling)

Bestellingsstatus = 'geplaatst'

5.3.2 Gebeurtenis: **bevestigenBestelling** en het informatieobject: **Bestelbevestiging**

Domein-View



Informatieobject: Bestelbevestiging

Afnemer : Afnemer_Ref

Leverancier : Leverancier_Ref

Bestelnr : Number

Bevestigingsregellijst: List [VervangendArtikelregel | BesteldArtikelregel | Niet_leverbaarregel]

[VervangendArtikelregel:

VervangendArtikel-tekst : Text ("vervangend artikel levering")

Bestelregel.regelnr : Number

Leveringsregel.regelnr : Number

Leveringsregel. hoeveelheid : Number

Bestelregel.artikel : ArtikelRef

Leveringsregel.artikel : ArtikelRef]

[BesteldArtikelregel:

BesteldArtikel-tekst : Text ("besteld artikel levering")

Bestelregel.regelnr : Number

Leveringsregel.regelnr : Number

Leveringsregel. hoeveelheid : Number

Bestelregel.artikel : ArtikelRef

Leveringsregel.artikel : ArtikelRef]

[Niet_leverbaar_regel:

Niet_leverbaar_tekst : Text ("niet leverbaar")

Bestelregel.regelnr : Number
 Bestelregel.bestelde_hoeveelheid : Number
 Bestelregel.artikel : ArtikelRef]

Lijstregelsoortcondities:

ForEach Bestelregel In Bestelbevestiging.Bestelling

ForEach Leveringsregel *with* Leveringsregel.Bestelregel = Bestelregel:

Leveringsregelstatus = "toegekend_besteld" : BesteldArtikelregel

Leveringsregelstatus = "toegekend_vervanging" :

VervangendeArtikelregel

notExists Leveringsregel *with* Leveringsregel.Bestelregel = Bestelregel:

Niet_leverbaarregel

Preconditions:

notNull [Afnemer]

notNull [Leverancier]

notNull [bestelnr]

Bestellingstatus= 'geplaatst'

ForEach Bestelling.Bestelregel *there exists at least one:*

[VervangendArtikelregel | BesteldArtikelregel | Niet_leverbaarregel]

ForEach Leveringsregel *with* Leveringsregel.Bestelregel = Bestelregel and

Leveringsregel.Leveringsregelstatus = "toegekend_besteld" there

exists one: BesteldArtikelregel

ForEach Leveringsregel *with* Leveringsregel.Bestelregel = Bestelregel and

Leveringsregel.Leveringsregelstatus = "toegekend_vervanging"

there exists one: VervangendArtikelregel

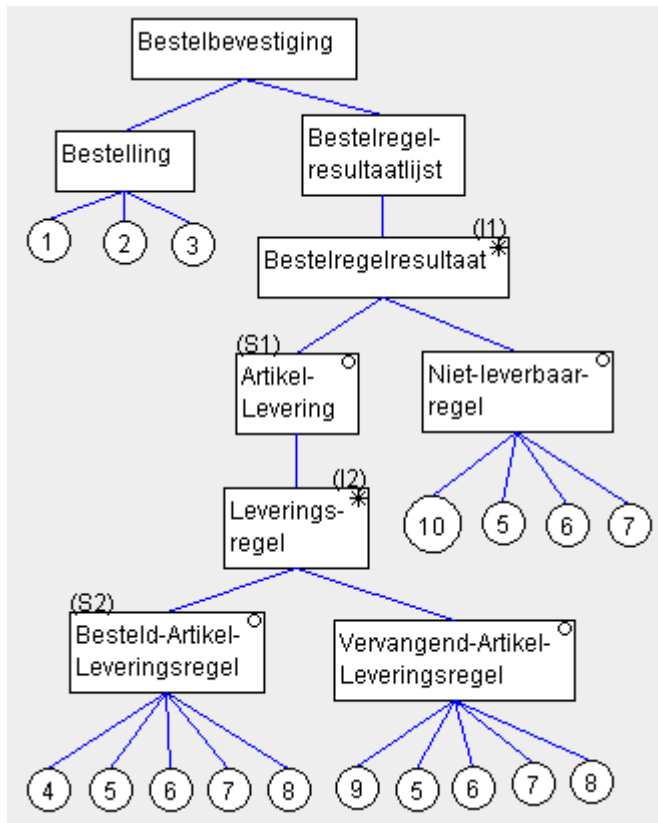
ForEach Bestelregel *without* Leveringsregel.Bestelregel there exists one:

Niet_leverbaarregel

Postconditions

Bestellingsstatus = 'bevestigd'

Opmerking: als (mogelijk transparanter) alternatief voor de presentatie van de (complexe) structuur van het informatieobject kunnen we eerder gebruikt volgordediagram gebruiken. Nu veel meer gebruikt als structuurdiagram (ter presentatie van de - mogelijk voorwaardelijke - onderdelen van het formulier) dan dat we de nadruk leggen op de volgordelijke afhankelijkheid.

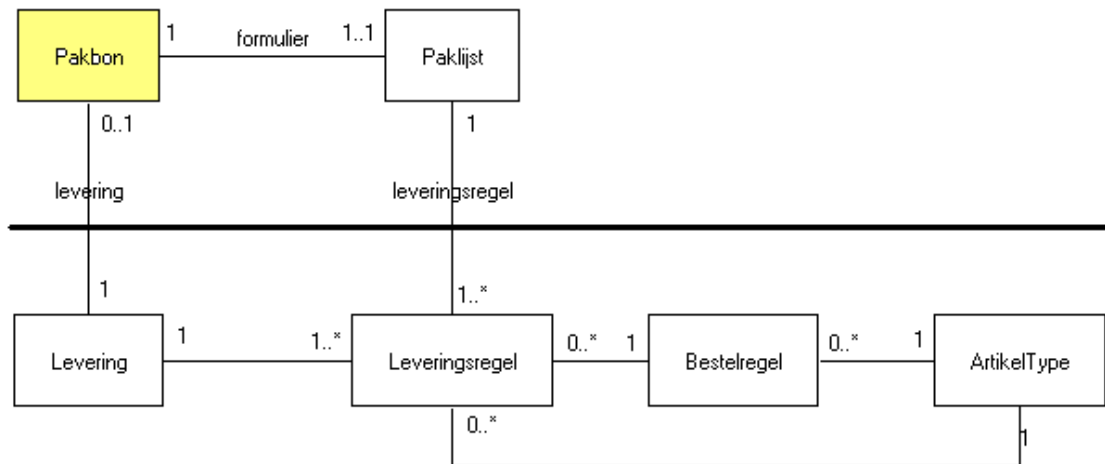


- 1: Afnemer : AfnemerRef
- 2: Leverancier : LeverancierRef
- 3: BestelNr : Number
- 4: Conformatietekst : Text ("besteldartikel levering")
- 5: BestelregelNr : Number
- 6: Artikel : ArtikelRef
- 7: Hoeveelheid : Number
- 8: Verwachte_leveringsdatum : TimeDate
- 9: VervangendArtikel-tekst : Text ("vervangend artikel levering")
- 10: Niet_leverbaar_tekst : Text ("niet leverbaar")

- I1: *ForEach* [Bestelregel in Bestelbevestiging.Bestelling]
- I2: *ForEach* [Leveringsregel with Leveringsregel.Bestelregel = Bestelregel]
- S1: *Exists* [Leveringsregel with Leveringsregel.Bestelregel = Bestelregel]
- S1': *notExists* [Leveringsregel with Leveringsregel.Bestelregel = Bestelregel]
- S2: Leveringsregelstatus = "toegekend_besteld"
- S2': Leveringsregelstatus = "toegekend_vervanging"

5.3.3 Gebeurtenis: **vooraankondigenLeveren** en het informatieobject: **Pakbon**

Domein-View:



Informatieobject: Pakbon

```

Afnemer : Afnemer_Ref          // afkomstig van Levering
Leverancier : Leverancier_Ref    //  ,,      ,,      ,,
leveringnr : Number
opmerkingen : Text
geplande_levering_tsk : TimeDate
Paklijst : List [Leveringsregel]
    [ Leveringsregel:
      Leveringsregel.Bestelregel.bestelNr : Number
      Leveringsregel.Artikel : ArtikelRef
      Leveringsregel.regelNr : Number
      Leveringsregel.hoeveelheid : Number
      Leveringsregel.opmerkingen : Text]
  
```

Preconditions

```

notNull [Afnemer]
notNull [Leverancier]
notNull [leveringnr]
notNull [geplande_levering_tsk]
Bestellingstatus= 'bevestigd'
Bestellingstatus = 'bevestigd'
notEmpty [ Paklijst ]
forEach [ Leveringregel]
    : unique [leveringsnr, regelNr]
    : hoeveelheid >= 1
    : notEmpty [artikel]
  
```

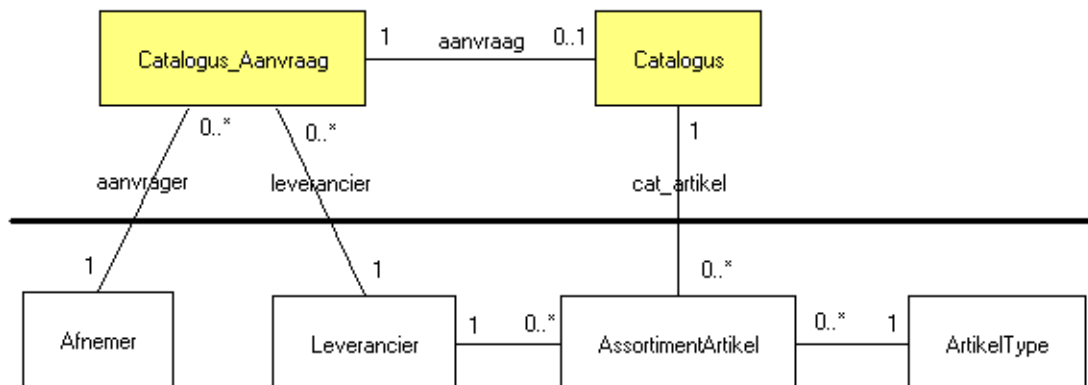
Postcondities (voor zover status bepaling)

```
Bestellingsstatus= "verwacht"
```

Bestellingsstatus_Leverancier = 'vooraangekondigd'

5.3.4 Request: *AanvragenCatalogus* met informatieobjecten: *CatalogusAanvraag* en *Catalogus*

Domein-View:



Informatieobject: *Catalogus Aanvraag*

Afnemer : Afnemer_Ref
 Leverancier : Leverancier_Ref
 Datum : Date

invarianten

notNull [Afnemer]
 notNull [Leverancier]

Informatieobject: *Catalogus*

aanvraag : CatalogusAanvraag
 opmerkingen : Tekst

Artikelen : List [AssortimentArtikelregel]
 [AssortimentArtikelregel:
 AssortimentArtikel.ArtikelType.artikelcode : Artikelcode
 AssortimentArtikel.ArtikelType.artikelbeschrijving : Tekst]

invarianten

notNull [aanvraag]
 notEmpty [Artikelen]
 forEach AssortimentArtikel where AssortimentArtikel.Leverancier =
 Catalogus_Aanvraag.Leverancier there exists one AssortimentArtikelregel

6 Modelimplementatie

In de voorgaande hoofdstukken zijn de modellen beschreven en vastgelegd die benodigd zijn om een semantische standaard vast te leggen. In dit hoofdstuk maken we een kort uitstapje naar het implementeren van deze modellen. Dit hoofdstuk geeft geen uitputtende toelichting op het implementeren van de modellen, het is alleen een eerste aanzet tot een implementatie.

6.1 Inleiding implementatie

Om de modellen die de semantische standaard definiëren in de praktijk te kunnen inzetten is een transformatie nodig. Deze paragraaf geeft een korte toelichting.

6.1.1 *Transformatie van modellen*

De meest ideale vorm van het omzetten van de modellen naar een implementatie is een automatische transformatie van de modellen naar een technische implementatie die in de praktijk gebruikt kan worden om informatie uit te wisselen. Dit kan bijvoorbeeld een database zijn waarin de informatie van alle partijen wordt opgeslagen en waaruit iedere partij de informatie kan halen die hij nodig heeft. Daarnaast zou het ook een transformatie kunnen zijn naar een opzet van berichtuitwisseling. Deze meest ideale vorm is op dit moment nog niet mogelijk voor alle mogelijke manieren van uitwisselen van informatie, de verwachting is dat dit in de toekomst wel mogelijk wordt.

6.1.2 *Standaard oplossingsarchitecturen voor realisatie van interactie*

Op het moment dat de modellen zijn vastgelegd is het noodzakelijk om na te gaan denken over hoe de informatie uitgewisseld gaat worden. Hiervoor zijn een aantal standaard oplossingsarchitecturen te gebruiken, onderstaand overzicht geeft een aantal mogelijkheden aan:

- Gebruik van een centrale database
- Uitwisseling van informatie met berichten één-op-één
- Uitwisseling van informatie met berichten via een centrale hub

6.1.3 *Gebruik van bouwstenen*

Op basis van de semantische modellen die opgesteld zijn voor de standaard is het mogelijk om basis bouwstenen te definiëren die gebruikt kunnen worden bij het uitwisselen van informatie. Door deze bouwstenen op een goede manier te herschikken is het mogelijk om bijvoorbeeld berichten te definiëren die gebruikt kunnen worden om informatie uit te wisselen. Door gebruik te maken van basis bouwstenen is het eenvoudiger om informatie uit te wisselen en kan voorkomen worden dat definities binnen de berichten uiteen beginnen te lopen.

6.2 Transformatiepatronen

Dit hoofdstuk geeft een korte inleiding op de transformatiepatronen die herkend kunnen worden bij een uitwisseling van informatie met behulp van berichten.

6.2.1 *Van bedrijfsgerichte berichtspecificaties naar technische berichtspecificaties*

Op basis van de modellen die opgesteld zijn is het mogelijk om bedrijfsgerichte berichtspecificaties op te stellen. In deze berichtspecificaties is vastgelegd welke informatie er uitgewisseld moet worden tussen partijen. Op basis van deze bedrijfsgerichte berichtspecificaties is het daarna noodzakelijk om de berichten om te zetten naar een technische berichtspecificaties, bijvoorbeeld in XML. Indien de bedrijfsgerichte berichtspecificaties in een

model zijn opgesteld dat te transformeren is naar bijvoorbeeld XML is het mogelijk om de berichten automatisch om te zetten.

6.2.2 Implementatie van dialogen

Bij het opstellen van uitwisselingspatronen is het mogelijk om de volledige berichtuitwisseling opnieuw te definiëren. Het is echter ook mogelijk om de bestaande (vaak papieren) dialogen om te zetten naar een uitwisseling van informatie met behulp van berichten. Dit zal niet in alle gevallen tot de meest ideale uitwisselingssituatie leiden omdat in sommige situaties de papieren situatie niet optimaal is, het vergt echter wel de minste inspanning op het gebied van het veranderen van bestaande processen.

6.2.3 Afbeelding van de specificatie op bestaande uitwisselingsstandaarden

Een laatste mogelijkheid bij het omzetten van de berichten naar een technische implementatie is het afbeelden van de berichtspecificaties op bestaande uitwisselingsstandaarden. In dit geval wordt de technische implementatie niet zelf bedacht, maar wordt aangesloten bij een al bestaande standaard. Om dit moment is het nog niet mogelijk om deze afbeelding automatisch uit te voeren, het is nog een handmatig proces waarbij degene die de standaard definieert zelf moet bepalen welke items passen bij de items in de bestaande standaard.

7 Semantische notaties

In de voorgaande hoofdstukken is de MOSES methode uitgewerkt om op een gestructureerde manier standaarden te ontwikkelen voor berichtenuitwisseling in een gemeenschappelijk bedrijfsdomein. In dit hoofdstuk wordt geanalyseerd in hoeverre semantische notatiewijzen kunnen worden gebruikt om de resultaten van de MOSES methode uit te drukken. Semantische notaties bieden de mogelijkheid om standaarden of berichten op te nemen in het semantische web, waardoor communicatie en distributie vereenvoudigd kan worden. Door het uitdrukken van een berichtenstandaard in een semantische taal is het in principe eenvoudiger om elementen en berichten te koppelen aan elementen uit andere domeinen. Een ander voordeel is de mogelijkheid tot hergebruik van (delen) van een met een semantische taal beschreven berichtenstandaard.

In dit hoofdstuk wordt geanalyseerd in hoeverre semantische notatiewijzen geschikt zijn om opgenomen te worden in de MOSES methode. Dit hoofdstuk bestaat uit vier paragrafen; in paragraaf 7.1 wordt de aanpak van semantische notaties in de MOSES methode beschreven; in paragraaf 7.2 wordt uiteengezet hoe de MOSES modellen worden getransformeerd naar semantische modellen; in paragraaf 7.3 wordt vervolgens de juwelierscase als voorbeeld uitgewerkt als semantisch mode; en in paragraaf 7.4 wordt de aanpak geëvalueerd.

7.1 Het gebruik van semantische notaties in MOSES

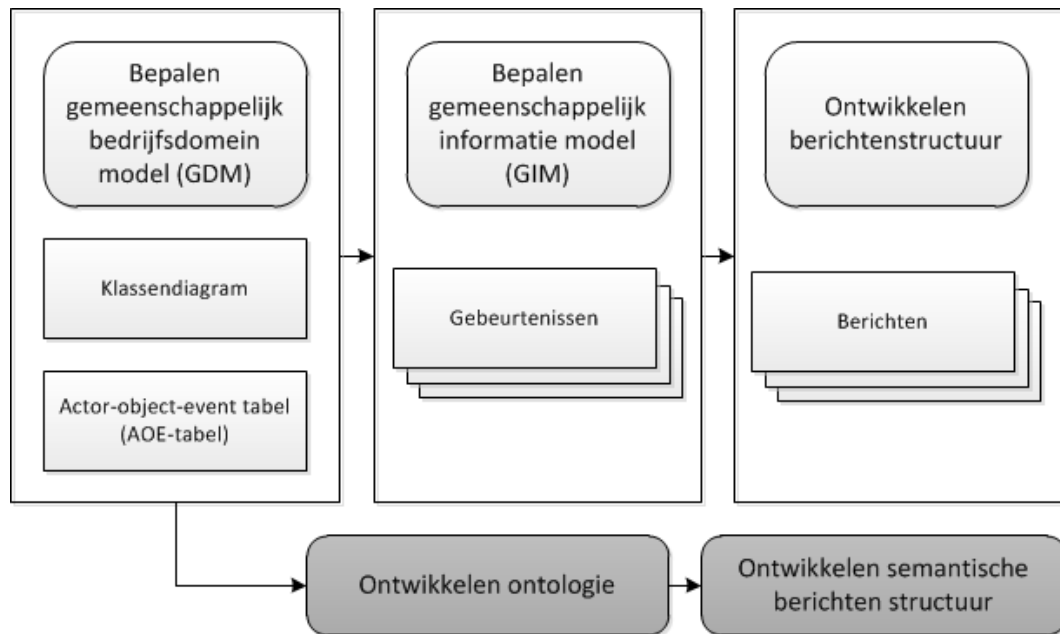
Semantische notatie is onderdeel van het semantische web (het 'web 3.0'), de volgende stap in de ontwikkeling van het internet. Waar 'web 1.0' statische webpagina's faciliteert, voegt 'web 2.0' end-user interactie toe door webpagina's uit te drukken in dynamische talen. Het semantische web legt boven op deze dynamische laag een semantische laag, waardoor automaten (internet zoekmachines, browsers, en andere intelligente software agents) zelf verbanden kunnen vinden tussen webpagina's en documenten. Het leggen van links tussen data, bestanden en internet pagina's, wat tot en met web 2.0 een handmatige activiteit is, is in web 3.0 daarmee geautomatiseerd.

Door producten van de MOSES methode uit te drukken in een semantische notatie, kunnen de ontwikkelde berichtenstandaarden ook worden opgenomen in het semantische web. Automatische redenering over, en het leggen van links van en naar concepten van een standaard kan hiermee worden gefaciliteerd.

In deze paragraaf wordt zowel de gekozen aanpak beschreven, als de keuze voor Web Ontology Language (OWL) als semantische notatietaal gemotiveerd.

7.1.1 Oplossingsrichting

In dit document is Web Ontology Language (OWL) gekozen om de mogelijkheden van het gebruik van een semantische taal te analyseren als onderdeel van de MOSES methode. Er is voor gekozen om de producten van de MOSES methode vanuit het gemeenschappelijke bedrijfsdomein (GDM) te vertalen naar semantische modellen, zie Figuur 2. Door de belangrijkste stappen in de MOSES methode te handhaven, is de analyse van toepasbaarheid van semantische talen een extensie van de huidige methode.



Figuur 2: Aanpak ontwikkelen semantische modellen in MOSES methode.

7.1.2 Aanpak

Voor deze analyse van uitbreidbaarheid van de MOSES methode met een semantische taal is gekozen om handmatig de ontwikkelde bedrijfsdomein modellen te transformeren naar semantische modellen. In praktijk betekent dit dat het klassendiagram en de actor-object-event tabel, zoals weergegeven in de eerste stap van Figuur 2, worden gebruikt als input voor een ontologie uitgedrukt in OWL. Deze ontologie is vervolgens weer bruikbaar als input voor een structuur van semantische berichten. Het ontwikkelen van de berichten zelf is geen onderdeel van de beschreven aanpak in dit hoofdstuk.

7.1.3 Web Ontology Language en Protégé

Voor de modelering is gekozen om OWL (Web Ontology Language) te gebruiken als semantische taal. OWL is een W3C-standaard voor het definiëren van web-ontologieën. In principe is OWL een verzameling van kennisrepresentatie talen die kunnen worden gebruikt om ontologieën te beschrijven. Deze verzameling wordt gekenmerkt door de formele semantiek, en de RDF/XML-gebaseerde serialisatie. In 2009 is de tweede versie van OWL gepresenteerd door de W3C werkgroep: OWL2. Deze versie is verwerkt in meerdere semantische editors, zoals Protégé, en semantische reasoners zoals Pellet, en HermiT, waarvan de laatste weer is geïmplementeerd in de Protégé tool. Er zijn drie versies van OWL beschikbaar; (1) OWL Lite, geschikt is voor classificatie hiërarchieën; (2) OWL DL (description logic), beschikt over maximale expressiviteit met behoud van computational completeness; en (3) OWL Full, met een andere semantiek dan OWL Lite en OWL DL, maar compatibel met RDF Schema (RDFS). Voor deze analyse is gekozen om OWL Full te gebruiken omdat het toestaat om aan een ontologie elementen te labelen met de betekenis van een voor gedefinieerde vocabulary (RDF of OWL). In deze analyse maken we gebruik van de Protégé software framework. Protégé is een open source ontologie editor, die ontwikkeld is aan Stanford Universiteit. De ontologieën die ontwikkeld worden met de Protégé editor kunnen worden geëxporteerd naar een breed scala formaten, zoals RDF(S), OWL, en XML Schema. De in Protégé editor kan een ontwikkelde ontologie gevisualiseerd worden met behulp van verschillende plug-ins.

7.2 Model transformatie

In deze paragraaf wordt beschreven hoe met behulp van een meta-model van de MOSES methode een gemeenschappelijk bedrijfsdomein kan worden getransformeerd naar OWL representatie. In de secties van deze paragraaf wordt ingegaan op het meta-model van de MOSES methode, de verschillende opties voor modeltransformaties, en een toelichting op het bedrijfsdomeinvoorbeeld dat wordt gebruikt in de toepassing van de beschreven aanpak.

7.2.1 MOSES meta-model

Voor dit onderzoek zijn drie variaties ontwikkeld van het MOSES meta-model. In onderstaande figuren zijn de drie variaties weergegeven, uitgedrukt in OWL, gevisualiseerd met behulp van Protegé. In appendix 9.1.1 is de bijbehorende OWL specificatie weergegeven van variant 3. De MOSES meta-modellen dient als basis voor model instanties, zoals een domein specifiek model (in dit document bijvoorbeeld de juweliersbranche).

In elk alternatief meta-model is de relatie van events, actoren, en bedrijfsobjecten impliciet of expliciet gemodelleerd. De relatie van actoren in events, of van bedrijfsobjecten in events noemen we in de MOSES methode participaties. Hiermee bedoelen we de betrokkenheid van een actor of object in een event, zoals die is weergegeven in Figuur 7. Afhankelijk van de gekozen modellering zijn in onderstaande alternatieven, zijn deze participaties dan wel expliciet of impliciet gemodelleerd. Wanneer een participatie expliciet is gemodelleerd is hiervoor een separate object opgenomen.

Objecten die onderdeel vormen van de verschillende varianten van het MOSES meta-model zijn:

ActorType: Personen en rollen in het gemeenschappelijk domein

ObjectType: Objecten in het gemeenschappelijk domein

EventType: Activiteiten / events in het gemeenschappelijk domein

ActorEventParticipation: Participatie van een actor en een event

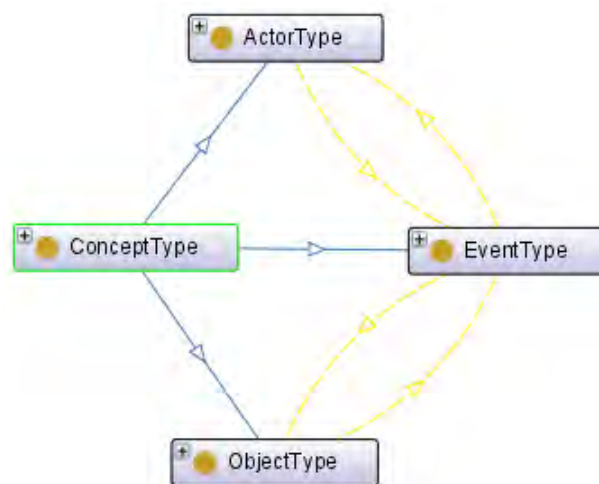
ObjectEventParticipation: Participatie van een object en een event

ActorObjectEventParticipation: Participatie van een actor en een object in een event

PropertyParticipation (niet weergegeven in de onderstaande meta-modellen): waarde die aan een ActorType, EventType, of ObjectType wordt meegegeven waarmee een relatie met een ActorEventParticipation of ObjectEventParticipation wordt gedefinieerd

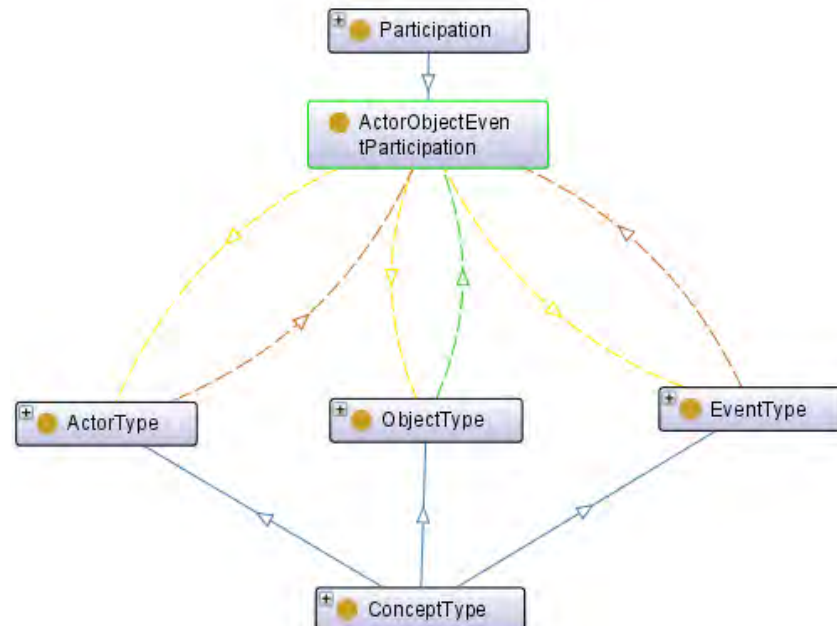
PropertyValue (niet weergegeven in de onderstaande meta-modellen): (data-)waarde die aan een ActorType, EventType, of ObjectType kan worden meegegeven.

Zoals gezegd hebben we voor de analyse van het gebruik van semantisch notaties als onderdeel van de MOSES methode drie alternatieve meta-modellen gedefinieerd. Een doel hiervan is om te analyseren welk meta-model het meest geschikt is om semantische bedrijfsdomein representaties te maken. In Figuur 3 is de eerste variant van het meta-model weergegeven. Dit model bestaat enkel uit drie concepttypen (ActorType, EventType, en ObjectType) en ParticipationProperty en ParticipationDataType (laatste twee objecten niet opgenomen in figuur). De participaties van ActorTypes in EventTypes en van ObjectTypes in EventTypes is impliciet gemodelleerd door de relaties die de objecten met elkaar hebben.



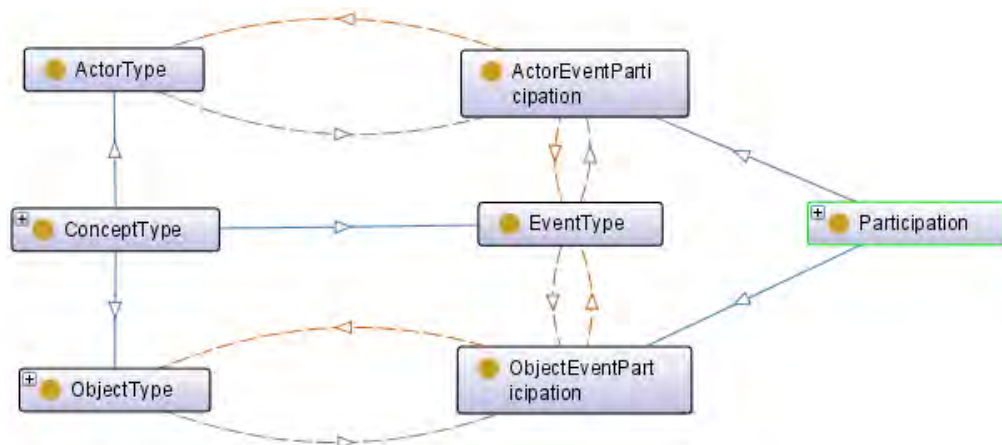
Figuur 3: MOSES meta-model, variant 1.

De relaties tussen de verschillende objecten in het meta-model zijn weergegeven met gestippelde lijnen en ononderbroken lijnen. Het laatste type lijnen impliceert een specialisatie – generalisatie relatie, zoals die onder andere bekend is uit de object-georiënteerde wereld. De gestippelde lijnen impliceren de semantische *Subject – Predicate – Object* relaties. Deze relaties worden ook wel aangeduid triples; de drie-eenheid van een onderwerp (subject), het type van relatie (predicate) en het object waar de relatie op van toepassing is (object). Een voorbeeld hiervan is (in natuurlijke taal uitgedrukt): Een Afnehmer (Subject) Participeert in (predicate) een Levering (Object).



Figuur 4: MOSES meta-model, variant 2.

De tweede variant van het meta-model is weergegeven in Figuur 4. Dit model is identiek aan het voorgaande model, met als verschil dat participaties van ActorTypes in EventTypes en ObjectTypes in EventTypes expliciet zijn gemodelleerd in één object; het Participation object ActorObjectEventParticipation. Hierdoor vervallen de onderlinge relaties die de ConceptTypes in het voorgaande figuur met elkaar deelden.



Figuur 5: MOSES meta-model, variant 3.

De derde variant van het meta-model van de MOSES methode bestaat uit de klassen ConceptType en Participation. De klassen ActorType, EventType, en ObjectType erven uit ConceptType, de klassen ActorEventParticipation en ObjectEventParticipation zijn op hun beurt specialisaties van Participation. De participatie van ObjectTypes in EventTypes en ActorTypes in EventTypes is hier dus expliciet gemodelleerd in twee afzonderlijke objecten.

7.2.2 Modeltransformatie

Op basis van het bovenstaande geformuleerde meta-model worden domeinmodellen afgeleid. Hiervoor wordt de input gebruikt die is gegenereerd in de eerste stappen, zie hiervoor Figuur 2. Voor modeltransformatie worden dus het UML klassendiagram en de Actor-Object-Event tabel gebruikt als input, specifiek voor een domein. De elementen in deze modellen worden getransformeerd naar een semantisch model door de concepten te mappen op de elementen uit het MOSES meta-model.

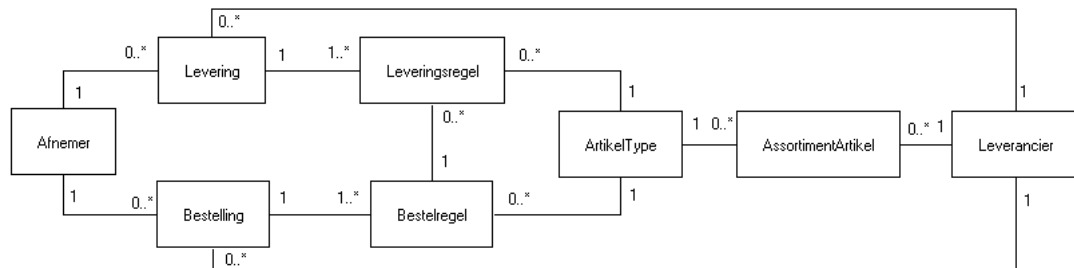
In deze analyse worden drie variaties gebruikt van het meta-model, om zo de beste configuratie te kunnen bepalen. Deze variaties verschillen in de mate waarin de participaties van Objecten, Events, en Actoren expliciet worden gemodelleerd. Participaties zijn in het actor-object-event tabel al geïdentificeerd, daarom zijn ze vrij eenvoudig te transformeren naar het semantische model.

In de eerste variatie van het meta-model zijn participaties van objecten, events, en actoren impliciet gemodelleerd in de relaties die deze elementen onderling hebben. Bijvoorbeeld een participatie van een object in een event is in deze variant weergegeven als een relatie tussen deze twee elementen. In de tweede variant zijn de participaties van objecten, events, en actoren expliciet gemodelleerd door middel van één Actor-Object-Event element. In de derde variant worden participaties ook expliciet gemodelleerd, enkel hier wordt onderscheid gemaakt tussen Object-Event participaties en Actor-Event participaties. In deze derde variant worden geen Actor-Object participaties gedefinieerd omdat een participatie altijd door een Event wordt getriggerd. In de uitwerkingen van de voorbeeld case, die in paragraaf 7.2.3 wordt behandeld, zijn de drie variaties gevisualiseerd.

7.2.3 MOSES voorbeeldcase

Het voorbeeld dat wordt gebruikt om de toepasbaarheid van semantische notatiewijzen binnen de MOSES methode te analyseren is net als in de andere hoofdstukken van dit document de juweliersbranche. In onderstaande figuren is nogmaals het UML klasse diagram en de actor-

object-event tabel weergegeven. Deze modellen vormen de basis voor de semantische modellen die in dit hoofdstuk worden ontwikkeld.



Figuur 6: UML klassendiagram; juweliersbranche

	Afnemer	Leverancier	ArtikelType	Bestelling	Bestelregel	Levering	Leveringsregel
aanmakenBestelling	I			O/C			
afsluitenBestelling	I	I		O/E			
opnemenBestelregel	I		A/M	A/M	O/C		
afsluitenBestelregel	I	I	A/M	A/M	O/E		
aanmakenLevering		I				O/C	
afsluitenLevering	I	I				O/E	
toekennenBesteldArtikel		I	A/M	A/M	A/M	A/M	O/C
toekennenVervangendArtikel		I	A/M, A/M	A/M	A/M	A/M	O/C
afsluitenLeveringsregel		I	A/M	A/M	A/M	A/M	O/E
verzenden		I				O/M	
plaatsenBestelling	V	P		O/M			
bevestigenBestelling	P	V		O/M			
annulerenBestelling	V/P	P/V		O/M			
vooraankondigenLevering	P	V				O/M	
annulerenLevering	P	V				O/M	
annulerenLeveringsregel	P	V	A/M	A/M	A/M	A/M	O/M

Figuur 7: Actor-object-event tabel; juweliersbranche

7.3 Juweliersstandaard uitgedrukt in OWL

In deze paragraaf worden de grafische weergaven van de OWL modellen weergegeven die het resultaat zijn van het gebruik van semantische notatiewijzen in de MOSES methode. Omdat de weergave van het hele model slecht leesbaar zou worden, is gekozen om vijf events te kiezen waarmee het gehele semantische model van de juweliersbranche in delen is geknipt. De vijf events uit Figuur 7 die hier worden behandeld zijn: *het annuleren van een bestelling*, *het annuleren van een levering*, *het plaatsen van een bestelling*, *het vooraankondigen van een levering*, en *het bevestigen van een bestelling*. In deze paragraaf is er voor gekozen om een segmentatie te maken van het semantische model van de juweliersbranche naar Events. Deze keuze is vanzelfsprekend arbitrair en zou ook gemaakt kunnen worden naar bijvoorbeeld Objecten of Actoren.

In de figuren die de semantische modellen representeren in deze paragraaf zijn implementaties gemaakt van het meta-model zoals dat in Figuur 5 is weergegeven. De volgende implementatie keuzen zijn gemaakt:

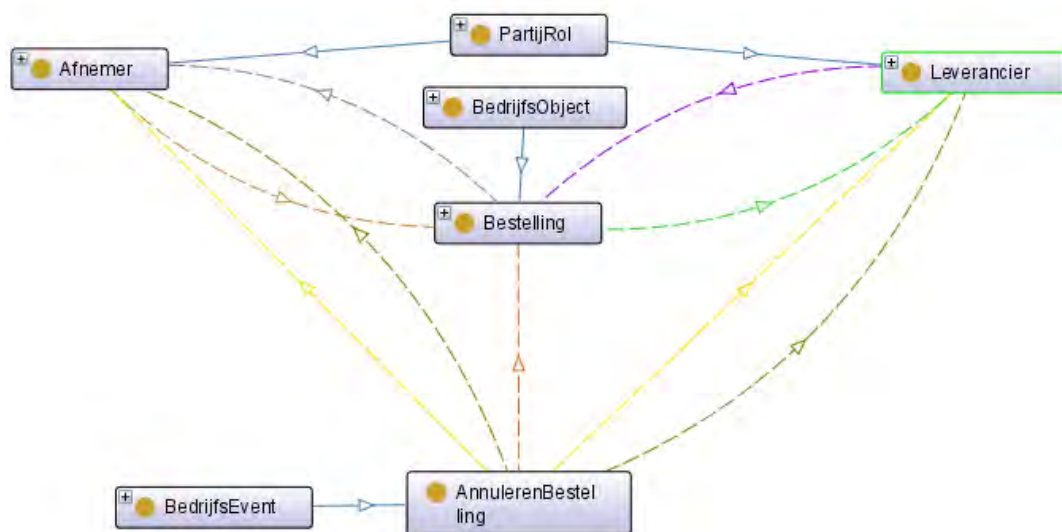
- PartijRol: implementatie van ActorType

- **BedrijfsObject**: implementatie van **ObjectType**
- **BedrijfsEvent**: implementatie van **EventType**
- **ActorEventParticipation** en **ObjectEventParticipation** zijn rechtstreeks uit het meta-model overgenomen
- **ActorObjectEventParticipatie** is een implementatie van de combinatie van **ActorEventParticipatie** en **ObjectEventParticipatie**
- In sommige modellen zijn de objecten **ActorEventParticipatie** en **ObjectEventParticipatie** uit het meta-model impliciet geïmplementeerd; de participaties zijn in deze modellen verwerkt in de relaties tussen de andere objecten die participeren in deze participaties.

In de onderstaande secties worden de vijf events die het semantische model van de juweliersbranche vormen gemodelleerd op basis van de drie meta-modellen die in paragraaf 7.2.1 zijn behandeld. In elke sectie worden daarom drie varianten weergegeven van het semantische model rondom een event.

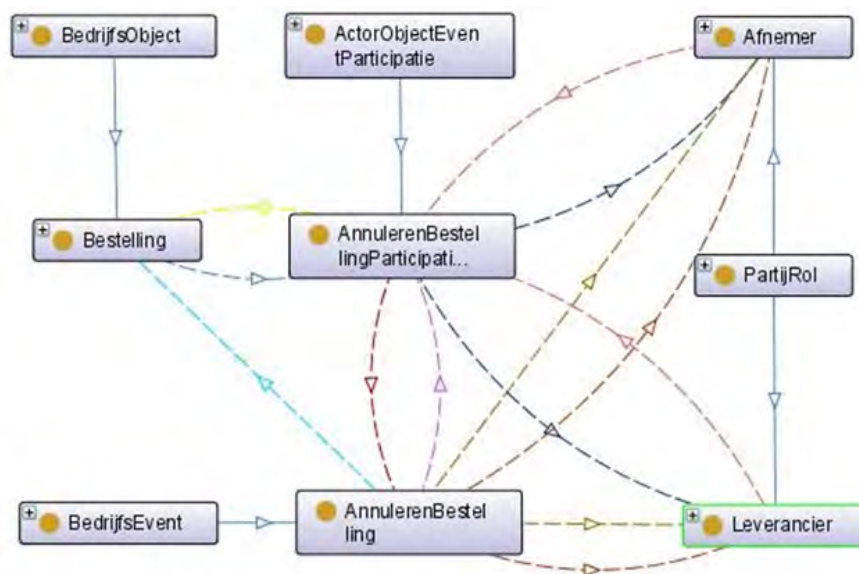
7.3.1 Annuleren bestelling

In Figuur 8 tot en met Figuur 10 is het semantisch model rondom het event object *AnnulerenBestelling* op drie manieren gevisualiseerd. In alle drie de figuren zijn de partij rollen: *Afnemer* en *Leverancier*, het bedrijfsobject *Bestelling*, en het bedrijfsevent *AnnulerenBestelling* opgenomen met bijbehorende onderlinge relaties. *Leverancier* en *Afnemer* zijn specialisaties van het type *PartijRol*, welke een instantiatie van het object *ActorType* uit het meta-model. *Bestelling* is een specialisatie van het type *BedrijfsObject*, welke een instantiatie is van het object *ObjectType* uit het meta-model. *AnnulerenBestelling* is een specialisatie van het type *BedrijfsEvent* welke weer een instantiatie is van het type *EventType* uit het meta-model.



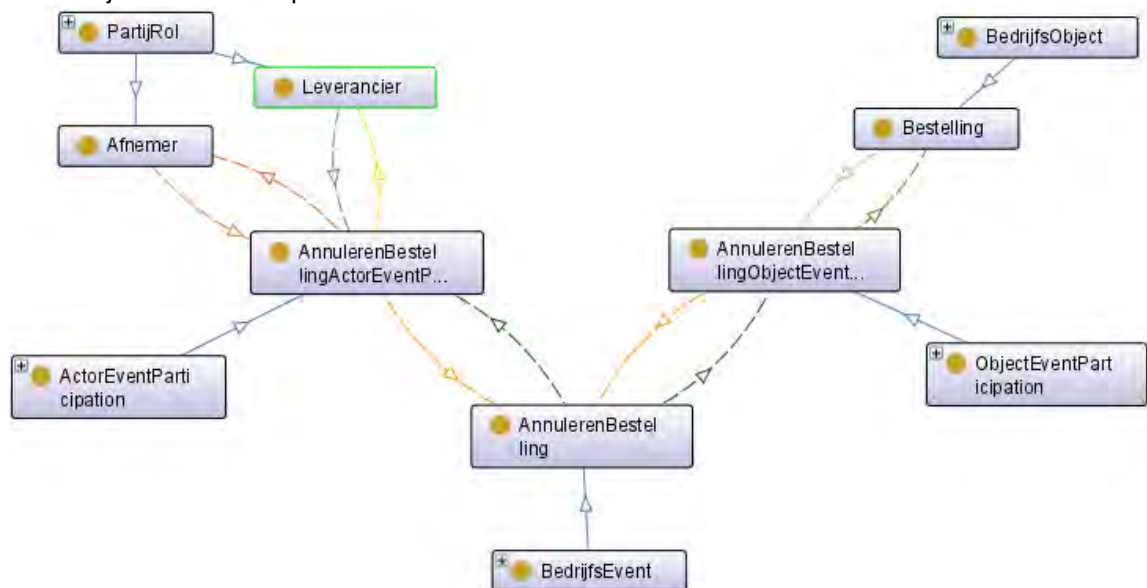
Figuur 8: Semantisch model: annuleren bestelling; alternatief 1.

In Figuur 8 is de participatie van de PartijRollen *Afnemer* en *Leverancier*, en het *BedrijfsObject* *Bestelling* met het *BedrijfsEvent* *AnnulerenBestelling*, impliciet gemodelleerd. Dit is te zien door de directe relaties die de elementen hebben in het model.



Figuur 9: Semantisch model: annuleren bestelling; alternatief 2.

In Figuur 9 is tweede variant van het meta-model gebruikt. In deze variant zijn de participaties van de PartijRollen, BedrijfsEvent, en BedrijfsObject expliciet gemodelleerd in één ActorObjectEventParticipatie.



Figuur 10: Semantisch model: annuleren bestelling; alternatief 3.

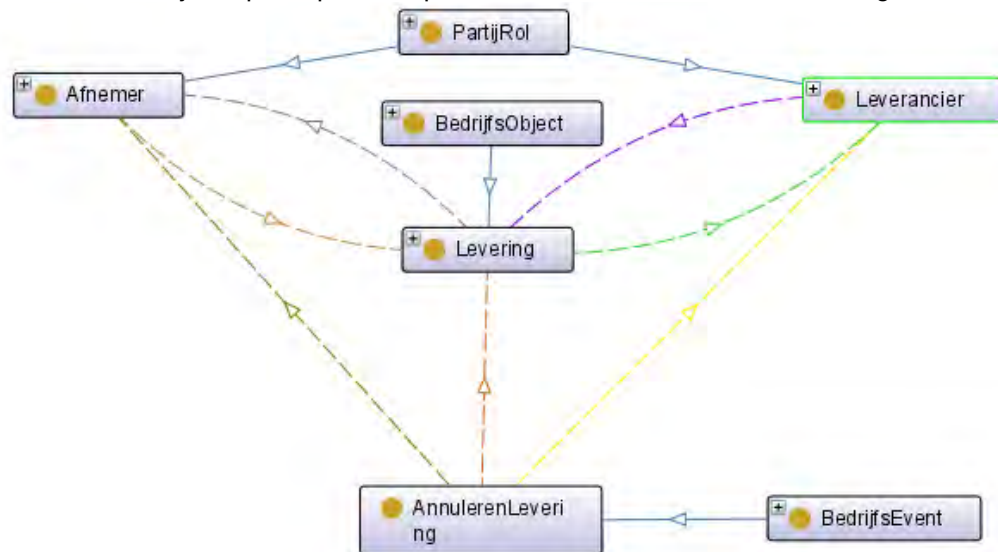
In de derde variant van het meta-model is een participatie expliciet gemodelleerd, en in twee afzonderlijke participaties, een ActorEventParticipatie en een ObjectEventParticipatie. Figuur 10 geeft de visualisatie van het semantische model rondom het bedrijfsevent AnnulerenBestelling weer.

7.3.2 Annuleren levering

In Figuur 11 tot en met Figuur 13 zijn de drie variaties op het semantische model rondom het BedrijfsEvent AnnulerenLevering gevisualiseerd. Dit deel van het semantische model komt overeen met het deel rondom het BedrijfsEvent AnnulerenBestelling, met als verschil het

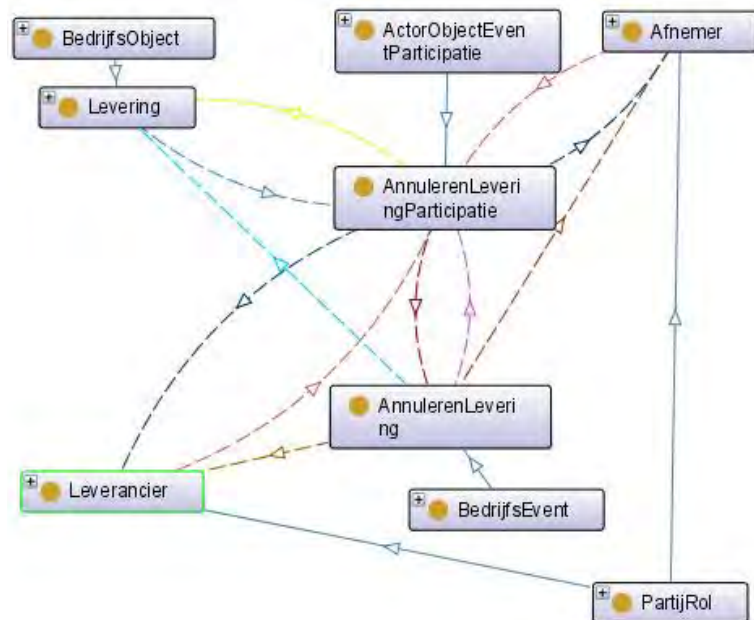
BedrijfsEvent. Ook in dit deel van het semantische model participeren de partijrollen Afnemer en Leverancier, en het bedrijfsobject Levering in het bedrijfsevent AnnulerenLevering

Figuur 11 geeft het semantische model weer dat is gebaseerd op de eerste variant van het meta-model; hierin zijn de participaties impliciet in de relaties tussen elementen gemodelleerd.

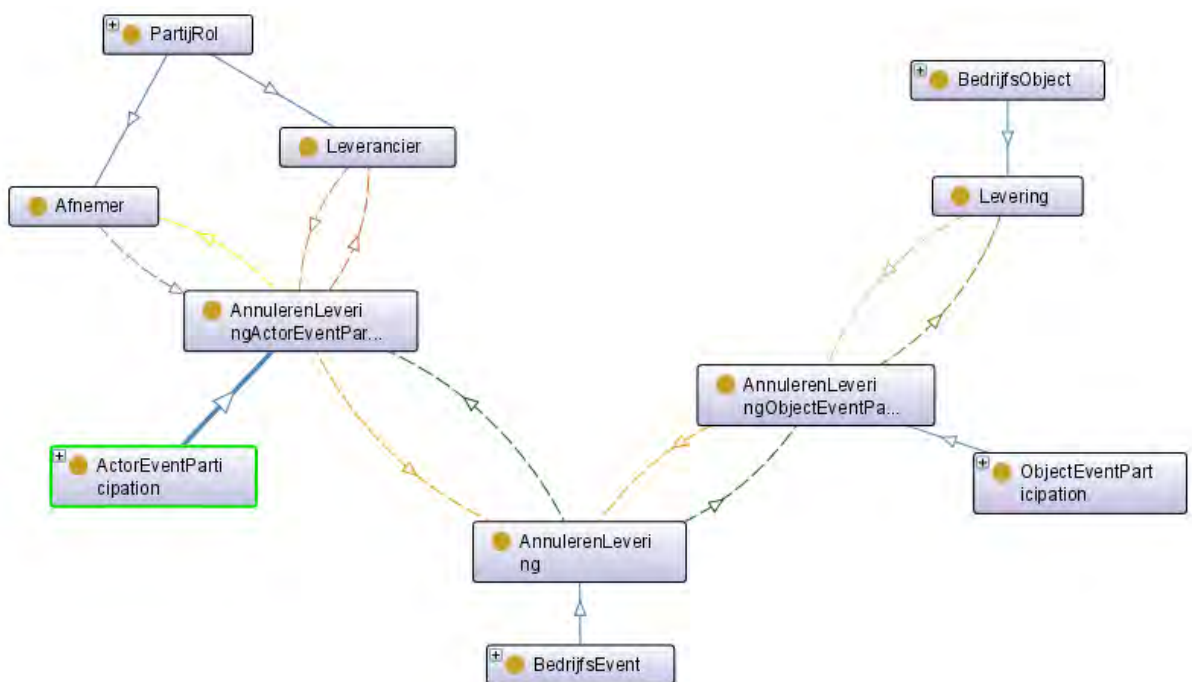


Figuur 11: Semantisch model: annuleren levering; alternatief 1.

In Figuur 12 is het semantische model weergegeven dat is afgeleid van de tweede variatie van het meta-model. De participaties zijn gemodelleerd in één element 'AnnulerenLeveringParticipatie', waarbij de PartijRollen, het BedrijfsEvent en het BedrijfsObject participeren.



Figuur 12 Semantisch model: annuleren levering; alternatief 2.

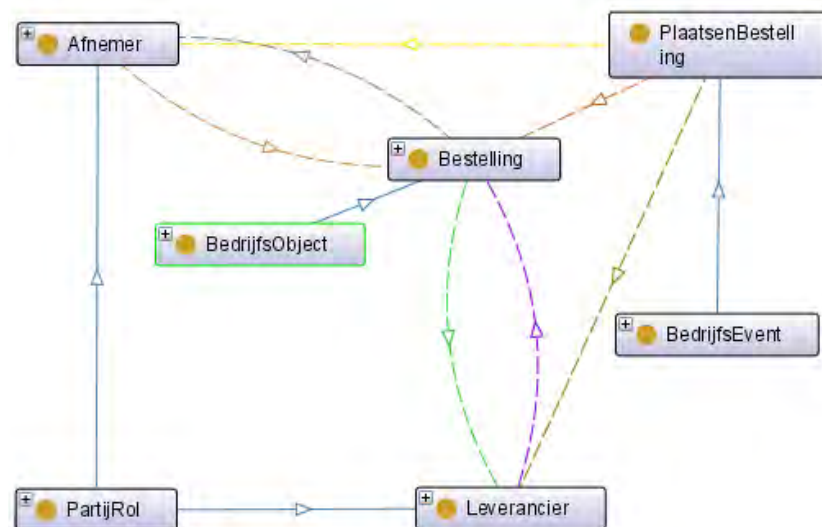


Figuur 13 Semantisch model: annuleren levering; alternatief 3.

In Figuur 13 is het semantische model rondom het BedrijfsEvent AnnulerenLevering weergegeven dat is afgeleid van de derde variant van het meta-model. In deze variant zijn de ObjectEventParticipaties en ActorEventParticipaties expliciet gemodelleerd.

7.3.3 Plaatsen bestelling

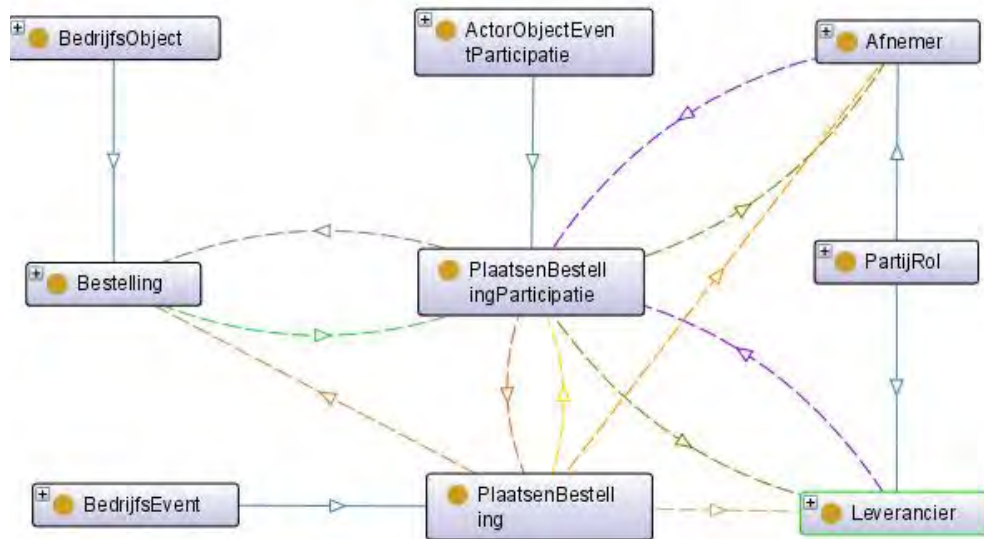
Figuur 14 tot en met Figuur 16 geven een visualisatie weer van het semantische model rondom het BedrijfsEvent PlaatsenBestelling. Ook hier zijn weer drie varianten van het meta-model gebruikt om dit deel van de semantiek te modelleren. Net als in de eerder besproken delen van het semantisch model, participeren twee partijrollen (Afnemer en Leverancier) en een bedrijfsobject (Bestelling) in het bedrijfsevent PlaatsenBestelling.



Figuur 14: Semantisch model: plaatsen bestelling; alternatief 1.

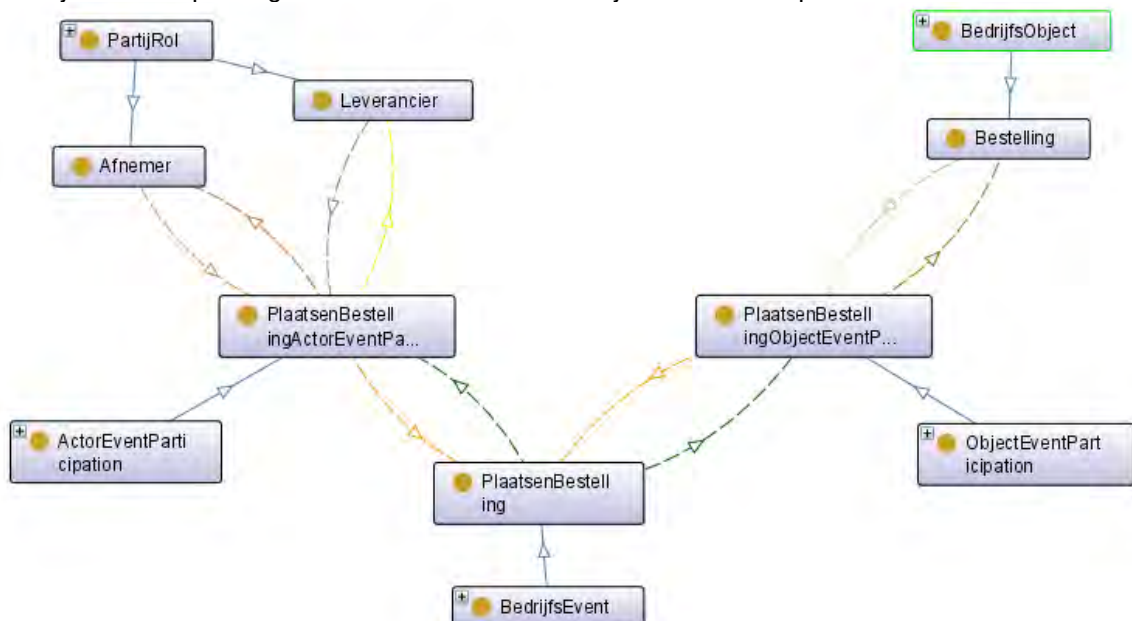
In bovenstaande figuur is het eerste alternatief voor het MOSES meta-model gebruikt om het semantische model rondom het BedrijfsEvent PlaatsenBestelling te modelleren. Net als de

andere behandelde delen van het semantische model zijn in deze variant de participaties impliciet gemodelleerd.



Figuur 15: Semantisch model: plaatsen bestelling; alternatief 2.

In Figuur 15 is de tweede variant van het MOSES meta-model gebruikt om een semantisch model te maken. In deze versie zijn de participaties van het BedrijfsObject, PartijRollen, en het BedrijfsEvent expliciet gemodelleerd in een ActorObjectEventParticipatie.



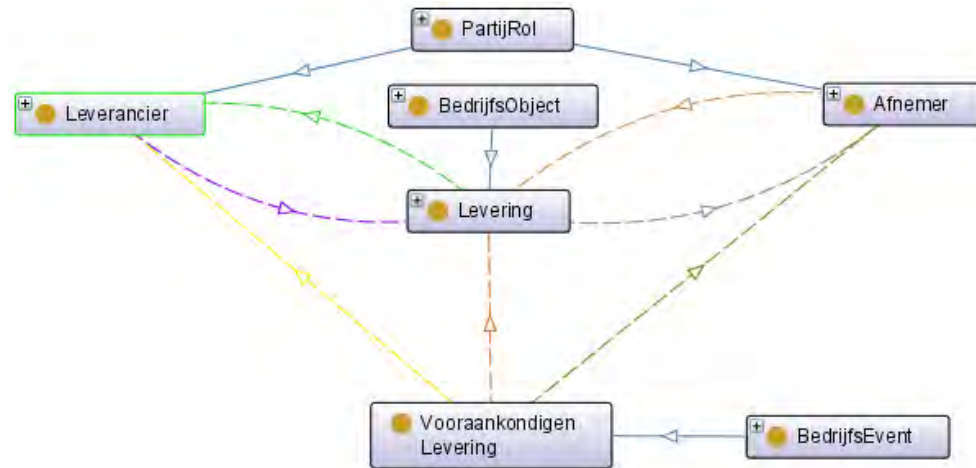
Figuur 16: Semantisch model: plaatsen bestelling; alternatief 3.

De derde variant van het meta-model van de MOSES methode is gebruikt om hetzelfde deel van het semantische model van de juweliersbranche te modelleren als Figuur 14 en Figuur 15. In Figuur 16 zijn twee expliciete participaties gemodelleerd; een ActorEvent participatie en een ObjectEventParticipatie.

7.3.4 Vooraankondigen levering

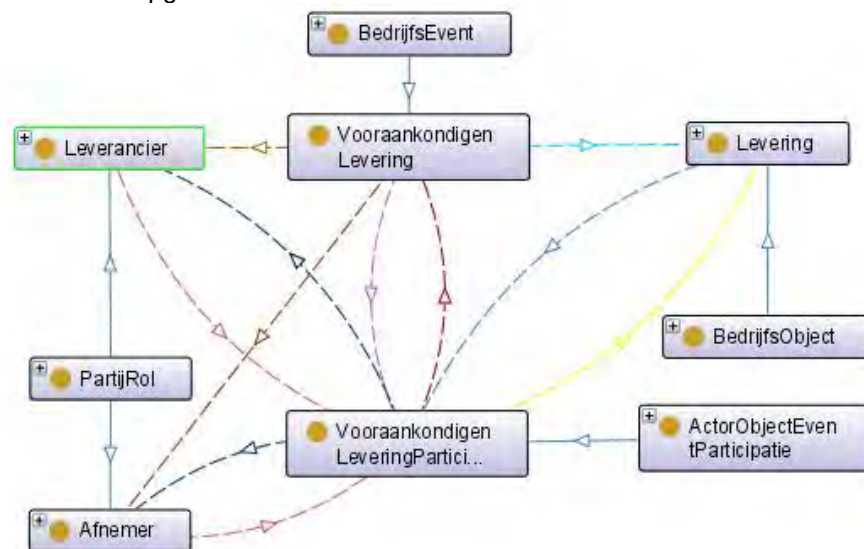
In Figuur 17 tot en met Figuur 19 is het deel van het semantische model van de juweliersbranche rondom het BedrijfsEvent VooraankondigenLevering weergegeven. Net als

met de andere behandelde delen van het semantische model zijn ook hier weer drie alternatieven weergegeven voor hetzelfde semantische model. In dit deel van het semantische model participeren de Partijrollen Leverancier en Afnemer samen met het BedrijfsObject Levering in het BedrijfsEvent VooraankondigenLevering.



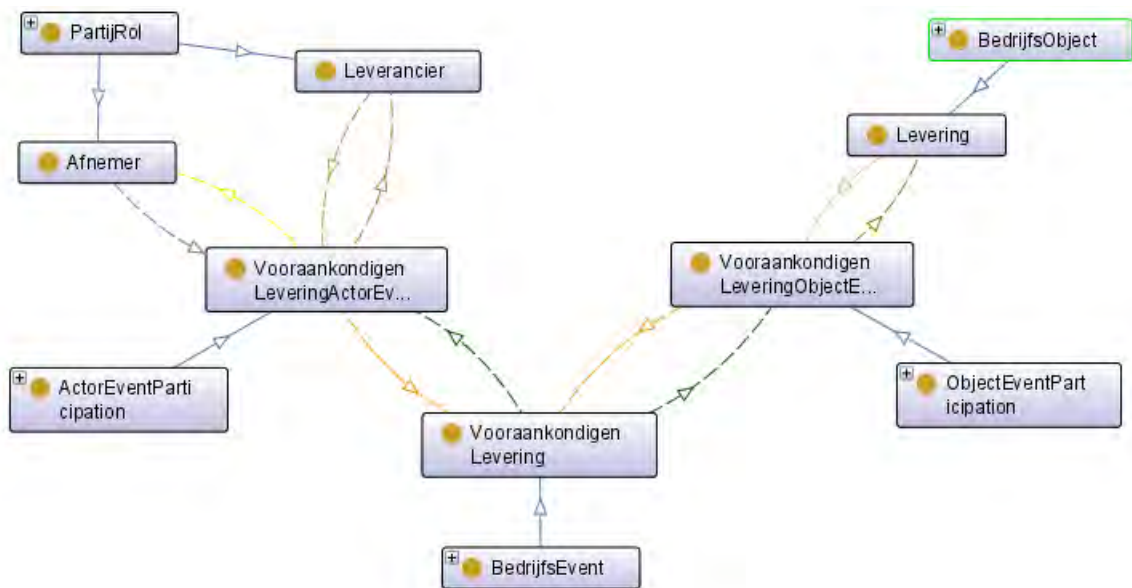
Figuur 17: Semantisch model: Vooraankondigen levering; alternatief 1.

In Figuur 17 is de variant van het semantisch model getoond dat is gebaseerd op het meta-model waarin participaties impliciet zijn gemodelleerd. Er zijn dan ook geen expliciete participatie objecten in het model opgenomen.



Figuur 18: Semantisch model: Vooraankondigen levering; alternatief 2.

In Figuur 18 zijn de participaties van Partijrollen en BedrijfsObject in het BedrijfsEvent expliciet gemodelleerd in een ActorObjectEventParticipatie.



Figuur 19: Semantisch model: Vooraankondigen levering; alternatief 3.

Figuur 19 laat een implementatie zien van de derde variant van het MOSES meta-model. Ook hier is weer het deel van het semantische model rondom de BedrijfsEvent VooraankondigenLevering gemodelleerd. In dit semantische model zijn twee participaties opgenomen; een ActorEventParticipatie, en een ObjectEventParticipatie. Beide participaties hebben een relatie met het BedrijfsEvent VooraankondigenLevering.

7.3.5 Bevestigen bestelling

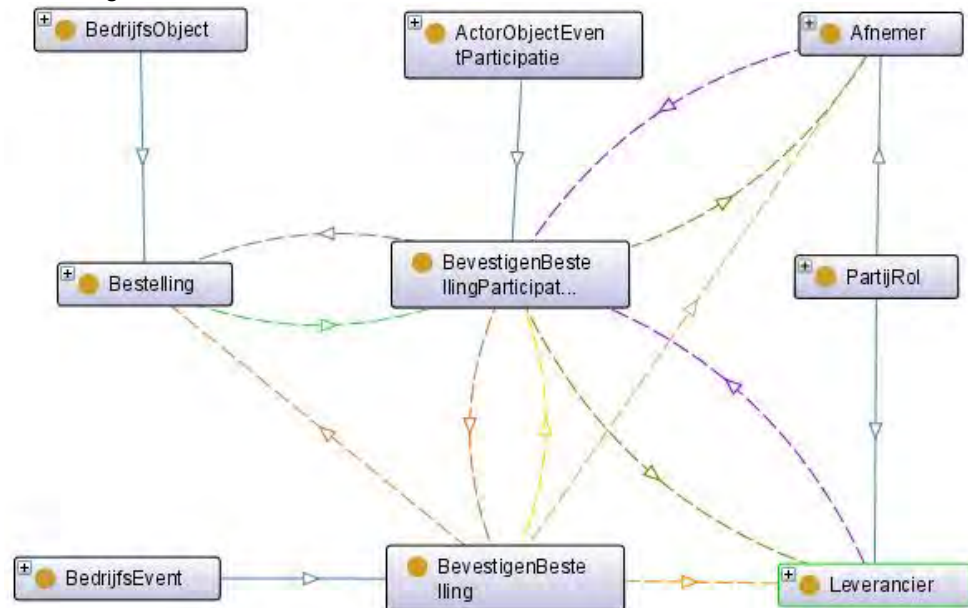
In deze sectie wordt het laatste deel van het (beknopte) semantische model van de juweliersbranche behandeld. Net als bij de voorgaande onderdelen van dit model, zijn weer drie variaties van het MOSES meta-model gebruikt om het semantische model van dit bedrijfsdomein te modelleren.

In Figuur 20 is de eerste variant van het meta-model gebruikt om de participaties rondom het BedrijfsEvent BevestigenBestelling te modelleren. In deze variant zijn participaties impliciet gemodelleerd in de relaties die de PartijRollen Afnemer en Leverancier hebben met het BedrijfsEvent BevestigenBestelling, en het BedrijfsObject Bestelling.



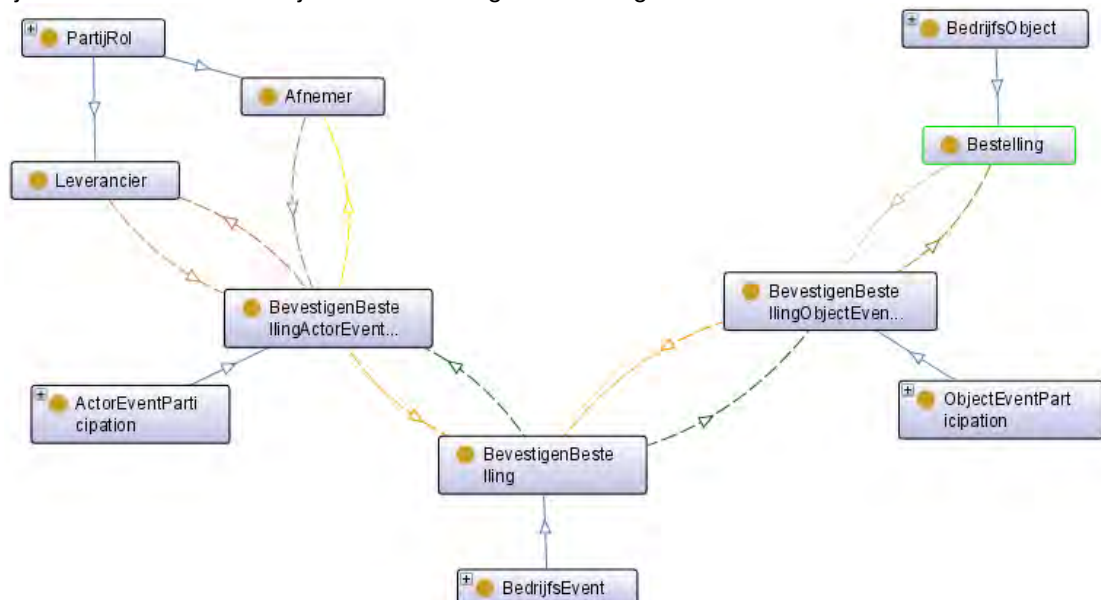
Figuur 20: Semantisch model: Bevestigen bestelling; alternatief 1.

In Figuur 21 zijn de participaties van de PartijRollen, het BedrijfsEvent en het BedrijfsObject expliciet gemodelleerd volgens de tweede variant van het MOSES meta-model, waarbij alle participaties expliciet worden weergegeven als een ActorObjectEventParticipatie in dit deel van de visuele weergave van het semantische model.



Figuur 21: Semantisch model: Bevestigen bestelling; alternatief 2.

In de derde en laatste variant van het meta-model zijn de participaties opgedeeld naar ActorEventParticipaties en ObjectEventParticipaties. Deze opdeling is analoog naar de derde variant van het meta-model voor de MOSES methode. In Figuur 22 is dit deel van de visualisatie van het semantische model van de juweliersbranche weergegeven dat betrekking heeft op de objecten rondom het BedrijfsEvent BevestigenBestelling.



Figuur 22: Semantisch model: Bevestigen bestelling; alternatief 3.

7.3.6 *Samenvatting*

De vijf onderdelen die in de voorgaande secties zijn besproken vormen samen het semantische model van de juweliersbranche. Gecombineerd bieden deze delen een overzicht. Gezien de complexiteit van een dergelijk groot model, weergegeven in een visualisatie in de Protegé tool-set, voegt een gecombineerde overview weinig toe. Wel is in appendix 9.1.2 de gehele OWL code weergegeven voor het semantische model van de juweliersbranche, gebaseerd op de derde variant van het MOSES meta-model.

7.4 **Evaluatie**

In deze paragraaf worden de resultaten van de gehanteerde aanpak geëvalueerd. Achtereenvolgens worden de aanpak, de variatie op het MOSES meta-model, het gebruik van OWL als semantisch taal en mogelijkheden voor toekomstige uitbreidingen besproken.

7.4.1 *Aanpak*

De aanpak in dit document om de resultaten van de MOSES methode om te zetten in een semantisch model is in de vorige paragraaf toegepast. Kort samengevat bestaat de aanpak voor de omzetting naar semantisch modellen zoals die is toegepast in dit hoofdstuk uit het ontwerpen van een meta-model van de MOSES methode. Vervolgens wordt de Actor-Object-Event-tabel (AOE-tabel) en het UML class diagram gebruikt als input voor een modeltransformatie, waarbij de resulterende modellen instanties zijn van de eerst ontwikkelde meta-modellen.

Deze aanpak is toegepast op het voorbeeld van de juweliersbranche, dat in de beschrijving van de MOSES methode zelf ook wordt gebruikt. De AOE-tabel en het UML classdiagram zoals die zijn weergegeven in Figuur 6 en Figuur 7, bieden input voor het creëren van modelinstanties op basis van de drie varianten van het meta-model. In de voorgaande paragraaf is het semantisch model van de juweliersbranche op deze manier opgebouwd (in drie varianten).

Het voordeel van deze aanpak is dat het in essentie vrij eenvoudig is om de resultaten van de MOSES methode om te zetten in semantische modellen. De informatie hiervoor is volledig aanwezig in het al gevormde classdiagram en de AOE-tabel; het enige wat rest is in feite modeltransformaties op basis van een meta-model.

Het nadeel van deze aanpak is dat het een vrij omslachtige methode is om tot enkel semantische modellen te komen. Wanneer de andere deliverables van de MOSES methode niet gewenst zijn, kan een semantisch model van een bedrijfsdomein eenvoudiger ontwikkeld worden (greenfielddaanpak, zie paragraaf 7.4.4).

Daarnaast kan de handmatige modeltransformatie een bron zijn van fouten. Op basis van de ervaringen die zijn opgedaan in dit project adviseren we dan ook om modellen in de MOSES methode automatisch te transformeren naar semantische modellen.

Wanneer enkel naar het resulterende model gekeken wordt kan worden gesteld dat het semantische model het bedrijfsdomein volledig dekt. Objecten en onderlinge relaties kunnen precies zo worden gemodelleerd als is gedefinieerd in de eerdere stappen van de MOSES aanpak. De aanpak voldoet dus aan de gestelde vraag.

7.4.2 *Meest geschikt meta-model*

In paragraaf 7.3 is het semantisch model van de juweliersbranche in vijf delen gepresenteerd waarbij de events in het domein de focusgebieden aangeven. Per deelgebied zijn drie model instanties weergegeven die zijn afgeleid van de drie variaties van het MOSES meta-model. Het

doel van deze exercitie is om te bepalen welk meta-model het meest geschikt is om het bedrijfsdomein op een inzichtelijke manier te vertalen naar een semantisch model. Gezien de expressiviteit van OWL en de visualisatiemogelijkheden in Protegé krijgt de derde variatie van het MOSES meta-model de voorkeur om te worden gebruikt voor modeltransformaties. In dit model worden object-event participaties en actor-event participaties expliciet gemodelleerd waardoor enerzijds de interactie tussen bedrijfsobjecten en bedrijfsevents en tussen partijrollen en bedrijfsevents duidelijk gespecificeerd blijft, en anderzijds het semantische model niet complex wordt.

7.4.3 Gebruik van semantische talen

De poging om semantische talen te gebruiken als extensie in de MOSES methode heeft twee doelen: ten eerste de analyse of resultaten van de MOSES methode omgezet kunnen worden naar semantische modellen, om deze zo in een semantische omgeving (zoals het semantische web) te kunnen plaatsen. In de vorige secties zijn we hier op ingegaan. Het tweede doel is de analyse in hoeverre OWL als semantische taal geschikt is om het bedrijfsdomeinmodel in het gedeelde informatiemodel uit te drukken. Deze sectie adresseert deze vraag.

Wanneer we kijken naar de 'fit' tussen OWL als taal om de producten van de MOSES methode uit te drukken zijn er twee aspecten die een rol spelen. Ten eerste is de mate van statische versus dynamische modellering van belang, ten tweede is de ondersteuning van de berichtenstructuur zoals die wordt uitgewisseld in het gemeenschappelijk domein van belang. Zoals is beschreven in sectie 7.1.3 is OWL een semantische taal voor web-ontologieën. Een web-ontologie representeert voornamelijk een statische structuur. Dat komt overeen met de MOSES methode waarin events zijn gemodelleerd als statische objecten, bijvoorbeeld het EventObject *PlaatsenBestelling*. Dat wil zeggen, events gebeuren niet, events zijn aanwezig; ze participeren in (bijvoorbeeld) ActorEventParticipaties. Echter, dynamisch gedrag kan voorkomen, bijvoorbeeld wanneer het BedrijfsObject *Bestelling* wordt geplaatst door het EventObject *PlaatsenBestelling*. Vervolgens wordt hetzelfde BedrijfsObject geannuleerd door het EventObject *AnnulerenLevering*. In principe is OWL hiervoor dus niet helemaal geschikt omdat dynamisch gedrag zoals statusverandering niet kan worden gemodelleerd. Daarbij dient te worden opgemerkt dat OWL een implementatietaal is, wat betekent dat veel high level problemen / misfits kunnen worden overkomen wanneer de taal daadwerkelijk wordt gebruikt.

7.4.4 Future research

De analyse, implementatie en evaluatie van het gebruik van semantische notaties als onderdeel van de MOSES methodologie geeft antwoord op de gestelde vragen wat betreft de toepasbaarheid, maar geeft ook aanleiding om extra onderzoek te doen. In grote lijnen identificeren we vier uitbreidingen op het gebied van het gebruik van semantische talen in de MOSES methodologie. In deze paragraaf worden de wenselijke onderzoeksrichtingen besproken.

Berichten ontwikkelen

In dit hoofdstuk is een aanpak beschreven waarmee de resultaten van de MOSES methode kunnen worden verwerkt tot modellen uitgedrukt in een semantische notatie. Op dit punt is enkel een semantisch domeinmodel ontwikkeld. Het uiteindelijke doel van het gebruik van semantische notatiewijzen is natuurlijk het kunnen uitdrukken van een berichtenstructuur voor een gemeenschappelijk bedrijfsdomein. Wanneer een semantische taal zoals OWL daadwerkelijk gebruikt gaat worden om berichten uit te wisselen in een gemeenschappelijk bedrijfsdomein, is een semantische berichtenstructuur ook gewenst. Daarom is het ontwikkelen van berichten op basis van het nu semantisch gemodelleerde bedrijfsdomein de volgende stap in dit proces.

Green field domein ontologie

De in dit hoofdstuk ontwikkelde ontologie van de juweliersbranche is gebaseerd op domeinmodellen die resultaat zijn van de MOSES methodologie, zoals de actor-object-event tabel en het class-diagram. Gedurende de analyse van deze aanpak werd het duidelijk dat deze methode omslachtig is; veel tijd wordt gestopt in de definitie van de samenhang van entiteiten in het domein, waarvan de samenhang niet direct kan worden teruggevonden in de semantische modellen. Het is daarom gewenst om te analyseren in hoeverre een semantisch model dat vanaf de grond wordt opgebouwd (green field) overeenkomt met de semantische modellen die resulteren uit de MOSES methode. Concreet stellen we voor om de resultaten van een green field aanpak te vergelijken met de resultaten van de MOSES methode, om zo efficiëntie, consistentie en volledigheid van beide resultaten te kunnen beoordelen.

Berichten op basis van meta-model

Een logische volgende stap in de MOSES methode is het ontwikkelen van berichten waarmee de ontwikkelde standaard gebruikt kan worden. Wanneer een semantisch model is ontwikkeld dat is uitgedrukt in een semantische notatietaal, kunnen berichten worden gegenereerd op basis van dit semantische model. Dit model kan dan functioneren als een meta-model waarmee berichten kunnen worden geïnstantieerd. Ook deze stap in het proces zien we als uitbreiding in de (semantische) MOSES methode.

Automatische modeltransformatie

De derde mogelijkheid tot toekomstige uitbreiding zien we in automatische modeltransformaties. Zoals aangegeven in de vorige sectie kunnen bijvoorbeeld berichten worden getransformeerd op basis van een meta-model. Het MOSES meta-model dat is ontwikkeld in sectie 7.2.1 kan dienen als model waarmee domeinmodellen, van bijvoorbeeld de juweliersbranche, eenvoudig kunnen worden getransformeerd. Hiervoor dient eerst een set van transformatieregels gedefinieerd te worden. Deze stap zien wij ook als uitbreiding van de beschreven semantische aanpak voor de MOSES methode.

8 Conclusies en mogelijke uitbreidingen

In dit document is een goede aanzet gegeven tot de MOSES methode voor model-gebaseerde ontwikkeling van semantische standaarden. De basisgedachte achter deze methode is de scheiding van probleemdomein en oplossingsdomein en het specificeren van het bedrijfsdomein als startpunt. De methode kan echter nog een aantal uitbreidingen gebruiken die in een vervolgtraject opgepakt zouden kunnen worden.

Deze uitbreidingen zien we op een aantal vlakken, te weten (1) verdere formele onderbouwing van de methodiek en integratie met semantische web- en kennis-modelleringsstandaarden (2) mogelijke werkwijzen en ontwikkelstrategieën die toegepast kunnen worden bij het opstellen van een semantische standaard met de MOSES methode.

8.1 Formelere onderbouwing en semantische notatiewijzen

De constructs en de regels die in de MOSES methode worden gebruikt zijn in de kern correct. Gegarandeerde modelconsistentie, nodig voor het kunnen garanderen van syntactische en semantische correctheid en compleetheid kent de methode echter nog niet. De onderliggende methode Merode (JSD) gebruikt voor deze garantie het zogenaamde “consistency-by-construction” en “one-model” principe. Merode is echter gericht op betekenisbeschrijving vanuit het perspectief van software-ontwikkeling (waar betekenisbeschrijving overigens core-business is) en niet direct op de ontwikkeling van interoperabiliteitsstandaarden. Een belangrijke vraag voor ons is hoe we een zelfde modelconsistentie als Merode biedt voor software-ontwikkeling kunnen inbrengen in MOSES ten behoeve van interoperabiliteitsontwerp. Dit alles om semantische correctheid en compleetheid te kunnen garanderen en de modellen interpreteerbaar voor automaten te maken.

Belangrijke onderzoeksvragen richten zich hierbij op:

- consistente en meer uitgebreide behandeling van de actoren in de (A)OET
- consistente wijze van omgaan met creatie-, beëindigings- en overige “niet-door-actoren-gedeelde” gebeurtenissen
- consistente behandeling van het statusloze karakter van gedeelde concepten in het interactiedomein en statusgerelateerde condities voor actor-activiteiten

Hiernaast willen we de methode integreren met een aantal belangrijke, hedendaagse trends in het modelleren en harmoniseren van ‘betekenis’. Ten eerste zijn dat de benaderingen en standaarden die zich ontwikkelen rond het ‘semantische web’, zoals W3C⁹’s RDF, RDFS en OWL. Deze ‘betekenisbeschrijvende talen’ richten zich immers ook op het expliciet maken van de semantiek van te delen domeinenconcepten op een wijze die voor automaten interpreteerbaar is. Ten tweede zijn dat de onder OMG¹⁰-vlag geplaatste benaderingen afkomstig uit de wereld van kennismodelleren en fact-based-modelling zoals SBVR en hieraan geassocieerde methoden als ORM. Ook deze methoden richten zich op betekenisbeschrijving met de bedoeling deze interpreteerbaar voor zowel mensen als automaten te maken. Ze richten zich hierbij sterk op het gebruik van natuurlijke taal en taalanalyse. Ten derde kennen we meer experimentele filosofieën ten aanzien van semantische harmonisatie en contextgerichtheid als Essence. Essence haar contextuele verbijzonderingstechniek komt (zoals het zich nu laat aanzien) vrijwel een-op-een overeen met MOSES (Merode’s) bestaansafhankelijkheid.

⁹ World Wide Web Consortium, RDF enz: <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>

¹⁰ Object Management Group: <http://www.omg.org/spec/SBVR/1.0/index.htm>

Bovendien kennen beide een formeel onderscheid tussen domein en functie (in MOSES gevat in GDM en GIM).

Een belangrijke toevoegde waarde van MOSES aan deze benaderingen zou mogelijk de expliciete gebeurtenisgerichtheid kunnen zijn. Bij het vastleggen van betekenis richten alle genoemde benaderingen zich op de statische structuren van een domeingebied vanuit een bepaald perspectief. MOSES voegt hier een dynamisch perspectief aan toe, de gebeurtenis en geassocieerde begrippen. Genoemde benaderingen beschrijven domeinen door concepten en hun onderlinge relatie te beschrijven (bv. deurstopper). Betekenis en context worden echter nog veel duidelijker als we tevens op een consistente wijze beschrijven wat er met de concepten gebeurt (aanschaffen, monteren) en onder welke condities (monteren na aanschaffen) dit gebeurt.

Belangrijke onderzoeksvragen hierbij zijn:

- hoe kunnen we RDF/OWL-ontologieën op een methodische wijze ontwikkelen overeenkomstig MOSES
- hoe kunnen we MOSES-gebeurtenisgerichtheid integreren met ORM's relatiegerichtheid
- hoe kunnen we taalanalysetechnieken, fact-based modelling, natuurlijke taalspecificatie uit bijvoorbeeld cogNiam en ORM integreren met MOSES
- wat voegt expliciete gebeurtenisgerichtheid daadwerkelijk toe aan genoemde benaderingen

8.2 Werkwijzen en ontwikkelingsstrategieën

Bij de aanpak van een project/programma waarin een semantische standaard moet worden ontwikkeld kunnen we uiteenlopende ontwikkelingsstrategieën worden gevolgd, zoals:

- Bottom-up: Vanuit de partijen die informatie moeten gaan uitwisselen vaststellen welke informatie er noodzakelijk is en op basis daarvan de standaard vaststellen.
- Top-down: Bij deze benadering worden eerst de algemene principes en vereisten bepaald waaraan de standaard moet voldoen. Op basis daarvan wordt vervolgens de invulling van de standaard op detailniveau bepaald.
- Middle-out: Bij deze benadering worden de bottom-up en top-down benadering gecombineerd om met invloed van beide kanten tot de meest gedragen standaard te komen. Deze lijkt de meest natuurlijke strategie voor inbedding en toepassing van MOSES.

Welke van deze drie benaderingen de beste is in welke situatie is nog een vraag.

Op het gebied van ontwikkelingsstrategieën zijn er een tweetal strategieën te benoemen die gebruikt kunnen worden bij de ontwikkeling van de standaard:

- Lineair: Een benadering waarbij de ontwikkeling van een standaard over een bepaalde tijd wordt uitgestreken op basis van het vastleggen van de uitgangspunten aan het begin van het project.
- Incrementeel, iteratief en agile: Een aanpak waarbij er telkens een deel van de standaard wordt opgeleverd die functioneel is. Hierbij wordt telkens een nieuw functioneel deel toegevoegd.

Ook hier is nog de vraag welke strategie het beste kan worden toegepast in welke situatie.

9 Appendices

9.1.1 OWL specificatie MOSES meta-model

```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.moses.com/ontologies/moses#"
  xml:base="http://www.moses.com/ontologies/moses"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">

  <owl:Ontology rdf:about="http://www.moses.com/ontologies/moses">

    </owl:Ontology>

    <!-- Object Properties -->

    <!--http://www.moses.com/ontologies/moses#hasParticipation -->
    <owl:ObjectProperty rdf:about="http://www.moses.com/ontologies/moses#hasParticipation">
      <rdfs:subPropertyOf rdf:resource="http://www.moses.com/ontologies/moses#participation"/>
    </owl:ObjectProperty>
    <!-- http://www.moses.com/ontologies/moses#hasProperty -->
    <owl:ObjectProperty rdf:about="http://www.moses.com/ontologies/moses#hasProperty">
      <rdfs:subPropertyOf rdf:resource="http://www.moses.com/ontologies/moses#property"/>
    </owl:ObjectProperty>

    <owl:ObjectProperty rdf:about="http://www.moses.com/ontologies/moses#isParticipationOf">
      <owl:inverseOf rdf:resource="http://www.moses.com/ontologies/moses#hasParticipation"/>
      <rdfs:subPropertyOf rdf:resource="http://www.moses.com/ontologies/moses#participation"/>
    </owl:ObjectProperty>

    <owl:ObjectProperty rdf:about="http://www.moses.com/ontologies/moses#isPropertyOf">
      <owl:inverseOf rdf:resource="http://www.moses.com/ontologies/moses#hasProperty"/>
      <rdfs:subPropertyOf rdf:resource="http://www.moses.com/ontologies/moses#property"/>
    </owl:ObjectProperty>

    <owl:ObjectProperty rdf:about="http://www.moses.com/ontologies/moses#participation"/>

    <owl:ObjectProperty rdf:about="http://www.moses.com/ontologies/moses#property"/>

    <!-- Classes -->

    <owl:Class rdf:about="http://www.moses.com/ontologies/moses#ActorEventParticipation">
      <rdfs:subClassOf rdf:resource="http://www.moses.com/ontologies/moses#Participation"/>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#hasParticipation"/>
          <owl:allValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#ActorType"/>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#hasParticipation"/>
          <owl:allValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#EventType"/>
        </owl:Restriction>
      </rdfs:subClassOf>
    </owl:Class>

    <owl:Class rdf:about="http://www.moses.com/ontologies/moses#ActorType">
      <rdfs:subClassOf rdf:resource="http://www.moses.com/ontologies/moses#ConceptType"/>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#hasProperty"/>
          <owl:someValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#PropertyValue"/>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#isParticipationOf"/>
          <owl:someValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#ActorEventParticipation"/>
        </owl:Restriction>
      </rdfs:subClassOf>
    </owl:Class>

  </rdf:RDF>
```

```

        </owl:Restriction>
      </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#hasProperty"/>
        <owl:someValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#PropertyParticipation"/>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

  <owl:Class rdf:about="http://www.moses.com/ontologies/moses#EventType">
    <rdfs:subClassOf rdf:resource="http://www.moses.com/ontologies/moses#ConceptType"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#isParticipationOf"/>
        <owl:someValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#ActorEventParticipation"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#hasProperty"/>
        <owl:someValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#PropertyParticipation"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#isParticipationOf"/>
        <owl:someValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#ObjectEventParticipation"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#hasProperty"/>
        <owl:someValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#PropertyValue"/>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

  <owl:Class rdf:about="http://www.moses.com/ontologies/moses#ObjectEventParticipation">
    <rdfs:subClassOf rdf:resource="http://www.moses.com/ontologies/moses#Participation"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#hasParticipation"/>
        <owl:allValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#EventType"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#hasParticipation"/>
        <owl:allValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#ObjectType"/>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

  <owl:Class rdf:about="http://www.moses.com/ontologies/moses#ObjectType">
    <rdfs:subClassOf rdf:resource="http://www.moses.com/ontologies/moses#ConceptType"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#hasProperty"/>
        <owl:someValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#PropertyParticipation"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#isParticipationOf"/>
        <owl:someValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#ObjectEventParticipation"/>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

```

```

<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#hasProperty"/>
    <owl:someValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#PropertyValue"/>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="http://www.moses.com/ontologies/moses#Property">
  <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>

<owl:Class rdf:about="http://www.moses.com/ontologies/moses#PropertyParticipation">
  <rdfs:subClassOf rdf:resource="http://www.moses.com/ontologies/moses#Property"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#isPropertyOf"/>
      <owl:someValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#ObjectType"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#isPropertyOf"/>
      <owl:someValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#EventType"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#isPropertyOf"/>
      <owl:someValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#ActorType"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="http://www.moses.com/ontologies/moses#PropertyValue"/>
</owl:Class>

<owl:Class rdf:about="http://www.moses.com/ontologies/moses#PropertyValue">
  <rdfs:subClassOf rdf:resource="http://www.moses.com/ontologies/moses#Property"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#isPropertyOf"/>
      <owl:someValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#ObjectType"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#isPropertyOf"/>
      <owl:someValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#EventType"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://www.moses.com/ontologies/moses#isPropertyOf"/>
      <owl:someValuesFrom rdf:resource="http://www.moses.com/ontologies/moses#ActorType"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

</rdf:Description>
</rdf:RDF>

```

9.1.2 OWL specificatie – Juweliersbranche

```

<?xml version="1.0"?>

<rdf:RDF
  xmlns="http://www.juwelier.com/ontologies/juwelier.owl#"
  xml:base="http://www.juwelier.com/ontologies/juwelier.owl"

```

```

xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:juwelier="http://www.juwelier.com/ontologies/juwelier.owl#"
xmlns:owl="http://www.w3.org/2002/07/owl#">

<owl:Ontology rdf:about="http://www.juwelier.com/ontologies/juwelier.owl"/>

<!-- Object Properties -->

<!-- http://www.juwelier.com/ontologies/juwelier.owl#actorEventParticipation -->
  <owl:ObjectProperty rdf:about="&juwelier;actorEventParticipation"/>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#eventParticipation -->
  <owl:ObjectProperty rdf:about="&juwelier;eventParticipation"/>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#hasAquired -->
  <owl:ObjectProperty rdf:about="&juwelier;hasAquired">
    <owl:propertyDisjointWith rdf:resource="&juwelier;hasOwnedMutated"/>
    <rdfs:subPropertyOf rdf:resource="&juwelier;objectEventParticipation"/>
  </owl:ObjectProperty>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#hasEvent -->
  <owl:ObjectProperty rdf:about="&juwelier;hasEvent">
    <rdfs:subPropertyOf rdf:resource="&juwelier;eventParticipation"/>
    <owl:propertyDisjointWith rdf:resource="&juwelier;isEventOf"/>
  </owl:ObjectProperty>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#hasOwnedMutated -->
  <owl:ObjectProperty rdf:about="&juwelier;hasOwnedMutated">
    <rdfs:subPropertyOf rdf:resource="&juwelier;objectEventParticipation"/>
  </owl:ObjectProperty>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#hasParticipation -->
  <owl:ObjectProperty rdf:about="&juwelier;hasParticipation">
    <rdfs:subPropertyOf rdf:resource="&juwelier;actorEventParticipation"/>
    <owl:propertyDisjointWith rdf:resource="&juwelier;hasResponsibility"/>
  </owl:ObjectProperty>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#hasResponsibility -->
  <owl:ObjectProperty rdf:about="&juwelier;hasResponsibility">
    <rdfs:subPropertyOf rdf:resource="&juwelier;actorEventParticipation"/>
  </owl:ObjectProperty>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#isAquiredBy -->
  <owl:ObjectProperty rdf:about="&juwelier;isAquiredBy">
    <owl:inverseOf rdf:resource="&juwelier;hasAquired"/>
    <owl:propertyDisjointWith rdf:resource="&juwelier;isOwnedMutatedBy"/>
    <rdfs:subPropertyOf rdf:resource="&juwelier;objectEventParticipation"/>
  </owl:ObjectProperty>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#isEventOf -->
  <owl:ObjectProperty rdf:about="&juwelier;isEventOf">
    <rdfs:subPropertyOf rdf:resource="&juwelier;eventParticipation"/>
    <owl:inverseOf rdf:resource="&juwelier;hasEvent"/>
  </owl:ObjectProperty>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#isOwnedMutatedBy -->
  <owl:ObjectProperty rdf:about="&juwelier;isOwnedMutatedBy">
    <owl:inverseOf rdf:resource="&juwelier;hasOwnedMutated"/>
    <rdfs:subPropertyOf rdf:resource="&juwelier;objectEventParticipation"/>
  </owl:ObjectProperty>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#isParticipationOf -->
  <owl:ObjectProperty rdf:about="&juwelier;isParticipationOf">
    <rdfs:subPropertyOf rdf:resource="&juwelier;actorEventParticipation"/>
    <owl:inverseOf rdf:resource="&juwelier;hasParticipation"/>

```

```

    <owl:propertyDisjointWith rdf:resource="&juwelier;isResponsibilityOf"/>
  </owl:ObjectProperty>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#isResponsibilityOf -->
  <owl:ObjectProperty rdf:about="&juwelier;isResponsibilityOf">
    <rdfs:subPropertyOf rdf:resource="&juwelier;actorEventParticipation"/>
    <owl:inverseOf rdf:resource="&juwelier;hasResponsibility"/>
  </owl:ObjectProperty>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#objectEventParticipation -->
  <owl:ObjectProperty rdf:about="&juwelier;objectEventParticipation"/>

<!-- Classes -->

<!-- http://www.juwelier.com/ontologies/juwelier.owl#ActorEventParticipation -->
  <owl:Class rdf:about="&juwelier;ActorEventParticipation"/>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#Afnemer -->
  <owl:Class rdf:about="&juwelier;Afnemer">
    <rdfs:subClassOf rdf:resource="&juwelier;PartijRol"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&juwelier;hasParticipation"/>
        <owl:someValuesFrom rdf:resource="&juwelier;AnnulerenLeveringActorEventParticipation"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&juwelier;hasParticipation"/>
        <owl:someValuesFrom rdf:resource="&juwelier;VooraankondigenLeveringActorEventParticipation"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&juwelier;hasResponsibility"/>
        <owl:someValuesFrom rdf:resource="&juwelier;AnnulerenBestellingActorEventParticipation"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&juwelier;hasResponsibility"/>
        <owl:someValuesFrom rdf:resource="&juwelier;PlaatsenBestellingActorEventParticipation"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&juwelier;hasParticipation"/>
        <owl:someValuesFrom rdf:resource="&juwelier;BevestigenBestellingActorEventParticipation"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="&juwelier;Leverancier"/>
  </owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#AnnulerenBestelling -->
  <owl:Class rdf:about="&juwelier;AnnulerenBestelling">
    <rdfs:subClassOf rdf:resource="&juwelier;BedrijfsEvent"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&juwelier;isEventOf"/>
        <owl:someValuesFrom rdf:resource="&juwelier;AnnulerenBestellingActorEventParticipation"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&juwelier;isEventOf"/>
        <owl:someValuesFrom rdf:resource="&juwelier;AnnulerenBestellingObjectEventParticipation"/>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

```

```

</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#AnnulerenBestellingActorEventParticipation -->
<owl:Class rdf:about="&juwelier;AnnulerenBestellingActorEventParticipation">
  <rdfs:subClassOf rdf:resource="&juwelier;ActorEventParticipation"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;isParticipationOf"/>
      <owl:someValuesFrom rdf:resource="&juwelier;Leverancier"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;hasEvent"/>
      <owl:someValuesFrom rdf:resource="&juwelier;AnnulerenBestelling"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;isResponsibilityOf"/>
      <owl:someValuesFrom rdf:resource="&juwelier;Afnemer"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#AnnulerenBestellingObjectEventParticipation -->
<owl:Class rdf:about="&juwelier;AnnulerenBestellingObjectEventParticipation">
  <rdfs:subClassOf rdf:resource="&juwelier;ObjectEventParticipation"/>

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;hasOwnedMutated"/>
      <owl:someValuesFrom rdf:resource="&juwelier;Bestelling"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;hasEvent"/>
      <owl:someValuesFrom rdf:resource="&juwelier;AnnulerenBestelling"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#AnnulerenLevering -->
<owl:Class rdf:about="&juwelier;AnnulerenLevering">
  <rdfs:subClassOf rdf:resource="&juwelier;BedrijfsEvent"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;isEventOf"/>
      <owl:someValuesFrom rdf:resource="&juwelier;AnnulerenLeveringObjectEventParticipation"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;isEventOf"/>
      <owl:someValuesFrom rdf:resource="&juwelier;AnnulerenLeveringActorEventParticipation"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#AnnulerenLeveringActorEventParticipation -->
<owl:Class rdf:about="&juwelier;AnnulerenLeveringActorEventParticipation">
  <rdfs:subClassOf rdf:resource="&juwelier;ActorEventParticipation"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;isParticipationOf"/>
      <owl:someValuesFrom rdf:resource="&juwelier;Afnemer"/>
    </owl:Restriction>
  </rdfs:subClassOf>

```

```

<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="&juwelier;hasEvent"/>
    <owl:someValuesFrom rdf:resource="&juwelier;AnnulerenLevering"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="&juwelier;isResponsibilityOf"/>
    <owl:someValuesFrom rdf:resource="&juwelier;Leverancier"/>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#AnnulerenLeveringObjectEventParticipation -->
<owl:Class rdf:about="&juwelier;AnnulerenLeveringObjectEventParticipation">
  <rdfs:subClassOf rdf:resource="&juwelier;ObjectEventParticipation"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;hasOwnedMutated"/>
      <owl:someValuesFrom rdf:resource="&juwelier;Levering"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;hasEvent"/>
      <owl:someValuesFrom rdf:resource="&juwelier;AnnulerenLevering"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#ArtikelType -->
<owl:Class rdf:about="&juwelier;ArtikelType">
  <rdfs:subClassOf rdf:resource="&juwelier;BedrijfsObject"/>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#AssortimentArtikel -->
<owl:Class rdf:about="&juwelier;AssortimentArtikel">
  <rdfs:subClassOf rdf:resource="&juwelier;BedrijfsObject"/>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#BedrijfsEvent -->
<owl:Class rdf:about="&juwelier;BedrijfsEvent"/>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#BedrijfsObject -->
<owl:Class rdf:about="&juwelier;BedrijfsObject"/>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#BestelRegel -->
<owl:Class rdf:about="&juwelier;BestelRegel">
  <rdfs:subClassOf rdf:resource="&juwelier;BedrijfsObject"/>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#Bestelling -->
<owl:Class rdf:about="&juwelier;Bestelling">
  <rdfs:subClassOf rdf:resource="&juwelier;BedrijfsObject"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;isOwnedMutatedBy"/>
      <owl:someValuesFrom rdf:resource="&juwelier;PlaatsenBestellingObjectEventParticipation"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;isOwnedMutatedBy"/>
      <owl:someValuesFrom rdf:resource="&juwelier;BevestigenBestellingObjectEventParticipation"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</rdfs:subClassOf>

```

```

        <owl:Restriction>
            <owl:onProperty rdf:resource="&juwelier;isOwnedMutatedBy"/>
            <owl:someValuesFrom rdf:resource="&juwelier;AnnulerenBestellingObjectEventParticipation"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#BevestigenBestelling -->
<owl:Class rdf:about="&juwelier;BevestigenBestelling">
    <rdfs:subClassOf rdf:resource="&juwelier;BedrijfsEvent"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&juwelier;isEventOf"/>
            <owl:someValuesFrom rdf:resource="&juwelier;BevestigenBestellingObjectEventParticipation"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&juwelier;isEventOf"/>
            <owl:someValuesFrom rdf:resource="&juwelier;BevestigenBestellingActorEventParticipation"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#BevestigenBestellingActorEventParticipation -->
<owl:Class rdf:about="&juwelier;BevestigenBestellingActorEventParticipation">
    <rdfs:subClassOf rdf:resource="&juwelier;ActorEventParticipation"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&juwelier;isParticipationOf"/>
            <owl:someValuesFrom rdf:resource="&juwelier;Afnemer"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&juwelier;hasEvent"/>
            <owl:someValuesFrom rdf:resource="&juwelier;BevestigenBestelling"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&juwelier;isResponsibilityOf"/>
            <owl:someValuesFrom rdf:resource="&juwelier;Leverancier"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#BevestigenBestellingObjectEventParticipation -->
<owl:Class rdf:about="&juwelier;BevestigenBestellingObjectEventParticipation">
    <rdfs:subClassOf rdf:resource="&juwelier;ObjectEventParticipation"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&juwelier;hasEvent"/>
            <owl:someValuesFrom rdf:resource="&juwelier;BevestigenBestelling"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&juwelier;hasOwnedMutated"/>
            <owl:someValuesFrom rdf:resource="&juwelier;Bestelling"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#Leverancier -->
<owl:Class rdf:about="&juwelier;Leverancier">
    <rdfs:subClassOf rdf:resource="&juwelier;PartijRol"/>
    <rdfs:subClassOf>

```



```

        <owl:Restriction>
            <owl:onProperty rdf:resource="&juwelier;hasResponsibility"/>
            <owl:someValuesFrom rdf:resource="&juwelier;VooraankondigenLeveringActorEventParticipation"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&juwelier;hasParticipation"/>
            <owl:someValuesFrom rdf:resource="&juwelier;AnnulerenBestellingActorEventParticipation"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&juwelier;hasResponsibility"/>
            <owl:someValuesFrom rdf:resource="&juwelier;AnnulerenLeveringActorEventParticipation"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&juwelier;hasParticipation"/>
            <owl:someValuesFrom rdf:resource="&juwelier;PlaatsenBestellingActorEventParticipation"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&juwelier;hasResponsibility"/>
            <owl:someValuesFrom rdf:resource="&juwelier;BevestigenBestellingActorEventParticipation"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#Levering -->
<owl:Class rdf:about="&juwelier;Levering">
    <rdfs:subClassOf rdf:resource="&juwelier;BedrijfsObject"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&juwelier;isOwnedMutatedBy"/>
            <owl:someValuesFrom rdf:resource="&juwelier;AnnulerenLeveringObjectEventParticipation"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&juwelier;isOwnedMutatedBy"/>
            <owl:someValuesFrom rdf:resource="&juwelier;VooraankondigenLeveringObjectEventParticipation"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#LeveringsRegel -->
<owl:Class rdf:about="&juwelier;LeveringsRegel">
    <rdfs:subClassOf rdf:resource="&juwelier;BedrijfsObject"/>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#ObjectEventParticipation -->
<owl:Class rdf:about="&juwelier;ObjectEventParticipation"/>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#PartijRol -->
<owl:Class rdf:about="&juwelier;PartijRol"/>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#PlaatsenBestelling -->
<owl:Class rdf:about="&juwelier;PlaatsenBestelling">
    <rdfs:subClassOf rdf:resource="&juwelier;BedrijfsEvent"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&juwelier;isEventOf"/>
            <owl:someValuesFrom rdf:resource="&juwelier;PlaatsenBestellingActorEventParticipation"/>
        </owl:Restriction>
    </rdfs:subClassOf>

```

```

    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&juwelier;isEventOf"/>
        <owl:someValuesFrom rdf:resource="&juwelier;PlaatsenBestellingObjectEventParticipation"/>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#PlaatsenBestellingActorEventParticipation -->
<owl:Class rdf:about="&juwelier;PlaatsenBestellingActorEventParticipation">
  <rdfs:subClassOf rdf:resource="&juwelier;ActorEventParticipation"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;hasEvent"/>
      <owl:someValuesFrom rdf:resource="&juwelier;PlaatsenBestelling"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;isParticipationOf"/>
      <owl:someValuesFrom rdf:resource="&juwelier;Leverancier"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;isResponsibilityOf"/>
      <owl:someValuesFrom rdf:resource="&juwelier;Afnemer"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#PlaatsenBestellingObjectEventParticipation -->
<owl:Class rdf:about="&juwelier;PlaatsenBestellingObjectEventParticipation">
  <rdfs:subClassOf rdf:resource="&juwelier;ObjectEventParticipation"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;hasEvent"/>
      <owl:someValuesFrom rdf:resource="&juwelier;PlaatsenBestelling"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;hasOwnedMutated"/>
      <owl:someValuesFrom rdf:resource="&juwelier;Bestelling"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#VooraankondigenLevering -->
<owl:Class rdf:about="&juwelier;VooraankondigenLevering">
  <rdfs:subClassOf rdf:resource="&juwelier;BedrijfsEvent"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;isEventOf"/>
      <owl:someValuesFrom rdf:resource="&juwelier;VooraankondigenLeveringActorEventParticipation"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;isEventOf"/>
      <owl:someValuesFrom rdf:resource="&juwelier;VooraankondigenLeveringObjectEventParticipation"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#VooraankondigenLeveringActorEventParticipation -->

```

```

<owl:Class rdf:about="&juwelier;VooraankondigenLeveringActorEventParticipation">
  <rdfs:subClassOf rdf:resource="&juwelier;ActorEventParticipation"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;isParticipationOf"/>
      <owl:someValuesFrom rdf:resource="&juwelier;Afnemer"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;hasEvent"/>
      <owl:someValuesFrom rdf:resource="&juwelier;VooraankondigenLevering"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;isResponsibilityOf"/>
      <owl:someValuesFrom rdf:resource="&juwelier;Leverancier"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<!-- http://www.juwelier.com/ontologies/juwelier.owl#VooraankondigenLeveringObjectEventParticipation -->
<owl:Class rdf:about="&juwelier;VooraankondigenLeveringObjectEventParticipation">
  <rdfs:subClassOf rdf:resource="&juwelier;ObjectEventParticipation"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;hasEvent"/>
      <owl:someValuesFrom rdf:resource="&juwelier;VooraankondigenLevering"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&juwelier;hasOwnedMutated"/>
      <owl:someValuesFrom rdf:resource="&juwelier;Levering"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<!-- General axioms -->

<rdf:Description>
  <rdf:type rdf:resource="&owl;AllDisjointClasses"/>
  <owl:members rdf:parseType="Collection">
    <rdf:Description rdf:about="&juwelier;AnnulerenBestellingActorEventParticipation"/>
    <rdf:Description rdf:about="&juwelier;AnnulerenLeveringActorEventParticipation"/>
    <rdf:Description rdf:about="&juwelier;BevestigenBestellingActorEventParticipation"/>
    <rdf:Description rdf:about="&juwelier;PlaatsenBestellingActorEventParticipation"/>
    <rdf:Description rdf:about="&juwelier;VooraankondigenLeveringActorEventParticipation"/>
  </owl:members>
</rdf:Description>

<rdf:Description>
  <rdf:type rdf:resource="&owl;AllDisjointClasses"/>
  <owl:members rdf:parseType="Collection">
    <rdf:Description rdf:about="&juwelier;ActorEventParticipation"/>
    <rdf:Description rdf:about="&juwelier;BedrijfsEvent"/>
    <rdf:Description rdf:about="&juwelier;BedrijfsObject"/>
    <rdf:Description rdf:about="&juwelier;ObjectEventParticipation"/>
    <rdf:Description rdf:about="&juwelier;PartijRol"/>
  </owl:members>
</rdf:Description>

<rdf:Description>
  <rdf:type rdf:resource="&owl;AllDisjointClasses"/>
  <owl:members rdf:parseType="Collection">
    <rdf:Description rdf:about="&juwelier;AnnulerenBestelling"/>
    <rdf:Description rdf:about="&juwelier;AnnulerenLevering"/>
  </owl:members>
</rdf:Description>

```

```
<rdf:Description rdf:about="&juwelier;BevestigenBestelling"/>
<rdf:Description rdf:about="&juwelier;PlaatsenBestelling"/>
<rdf:Description rdf:about="&juwelier;VooraankondigenLevering"/>
</owl:members>
</rdf:Description>

<rdf:Description>
  <rdf:type rdf:resource="&owl;AllDisjointClasses"/>
  <owl:members rdf:parseType="Collection">
    <rdf:Description rdf:about="&juwelier;AnnulerenBestellingObjectEventParticipation"/>
    <rdf:Description rdf:about="&juwelier;AnnulerenLeveringObjectEventParticipation"/>
    <rdf:Description rdf:about="&juwelier;BevestigenBestellingObjectEventParticipation"/>
    <rdf:Description rdf:about="&juwelier;PlaatsenBestellingObjectEventParticipation"/>
    <rdf:Description rdf:about="&juwelier;VooraankondigenLeveringObjectEventParticipation"/>
  </owl:members>
</rdf:Description>

<rdf:Description>
  <rdf:type rdf:resource="&owl;AllDisjointClasses"/>
  <owl:members rdf:parseType="Collection">
    <rdf:Description rdf:about="&juwelier;ArtikelType"/>
    <rdf:Description rdf:about="&juwelier;AssortimentArtikel"/>
    <rdf:Description rdf:about="&juwelier;BestelRegel"/>
    <rdf:Description rdf:about="&juwelier;Bestelling"/>
    <rdf:Description rdf:about="&juwelier;Levering"/>
    <rdf:Description rdf:about="&juwelier;LeveringsRegel"/>
  </owl:members>
</rdf:Description>

<rdf:Description>
  <rdf:type rdf:resource="&owl;AllDisjointProperties"/>
  <owl:members rdf:parseType="Collection">
    <rdf:Description rdf:about="&juwelier;actorEventParticipation"/>
    <rdf:Description rdf:about="&juwelier;eventParticipation"/>
    <rdf:Description rdf:about="&juwelier;objectEventParticipation"/>
  </owl:members>
</rdf:Description>

</rdf:RDF>
```