

Auteur: A.K. Boerma
Datum: 14-10-2013
Titel: Versioneren zaaktype
JIRA: <https://excellencejira.atlassian.net/browse/MO-54>

1. Aanleiding

In het Excellence ZaakSysteem 5.1 wordt het zaaktype, inclusief het bijbehorende behandelproces, geversioneerd. Dit is noodzakelijk om nieuwe versies van een behandelproces te kunnen modelleren / implementeren terwijl nog lopende zaken afgehandeld worden volgens het behandelproces dat in gebruik was ten tijde van het aanmaken van de zaak. In deze opzet is een bepaalde versie altijd in één van de volgende toestanden:

- *Gedeactiveerd:* de proces(sen) die in het verleden actief zijn geweest maar niet langer gebruikt worden.
- *Actief:* het proces dat gebruikt wordt om nieuwe zaken van een bepaald type mee af te handelen. Per zaaktype kan er maximaal één actief behandelproces zijn. Dit proces kan *gedeactiveerd* worden, verwijderen daarentegen is niet toegestaan.
- *Draft:* het proces dat in de toekomst eventueel gebruikt kan gaan worden om nieuwe zaken van een bepaald type mee af te handelen. Dit proces wordt gebruikt om aanpassingen / uitbreidingen op te ontwikkelen / testen zonder dat deze direct van toepassing zijn op nieuwe zaken. Per zaaktype kan er maximaal één 'draft' behandelproces aanwezig zijn, deze mag verwijderd worden.

In de StUF standaarden (RGBZ / ZTC 2.0) kan deze toestand vastgelegd worden via de '*datum begin geldigheid*' en '*datum einde geldigheid*' attributen van het zaaktype. De begin datum dient, logischerwijs, altijd voor de einddatum te liggen en de datum ranges van verschillende versies mogen nooit overlappen (wel mogen er leemtes in de tijdslijn zitten, een zaaktype kan immers tijdelijk gedeactiveerd zijn). Binnen het Excellence ZaakSysteem 5.1 worden de volgende conventies gehanteerd om de eerder beschreven toestand van een zaaktype af te leiden:

- Een gedeactiveerde versie van een zaaktype heeft zowel een begin als eind datum geldigheid. Er zit geen restrictie op het aantal gedeactiveerde versies van een zaaktype dat aanwezig mag zijn. Een gedeactiveerde versie kan niet verwijderd worden omdat hierdoor inconsistenties kunnen ontstaan.
- De actieve versie van een zaaktype heeft een begin datum geldigheid *voor* de huidige datum en een eind datum geldigheid *na* huidige datum (of leeg). Daarnaast geldt dat er op elke moment maximaal één actieve versie aanwezig mag zijn. Een actieve versie kan niet verwijderd worden, alleen deactiveren is toegestaan (door de eind datum geldigheid in te vullen die op of na de huidige datum valt).
- De draft versie van een zaaktype heeft geen begin en eind datum geldigheid. Daarnaast geldt dat er maximaal één draft versie aanwezig mag zijn, idealiter wordt deze regel op database niveau afgezekerd (voor de liefhebbers, dit kan middels een partiële index). Een draft versie mag altijd verwijderd worden.

Om voor beheerders een duidelijker onderscheid te kunnen maken tussen de diverse versies van een bepaald zaaktype is in het Excellence ZaakSysteem 5.0 een versienummer toegevoegd (bestaande uit twee getallen van elk maximaal twee cijfers, gescheiden door een enkele punt). Dit versienummer moet opgegeven worden door een beheerder als deze een nieuwe draft versie aanmaakt voor een zaaktype.

Een (vervelend) gevolg van het specificeren van geldigheidsduren op datum niveau is dat per dag slechts één draft versie actief gemaakt kan worden. Voor een productie omgeving is dit geen probleem omdat het aantal updates van een zaaktype daar beperkt zal zijn. Voor een test/acceptatie omgeving is dit echter onwerkbaar, het moet daar mogelijk zijn meerdere versies te doorlopen op één en dezelfde dag. Binnen de Excellence ZaakSysteem wordt dit probleem opgelost door een extra operatie toe te voegen waarmee, tegen de eerder gestelde regels in, van een actief zaaktype weer een draft gemaakt kan worden. Om vervuiling van het

zaakstelsysteem te voorkomen worden eerst de zaken van het actieve zaaktype verwijderd waarna de geldigheidsduur aangepast wordt. Door hiervoor een aparte operatie te gebruiken kunnen hier specifieke gebruikersrechten aan gekoppeld worden waarmee voorkomen kan worden dat medewerkers per ongeluk zaken verwijderen.

Om simultaan meerdere versies van hetzelfde zaaktype te kunnen ondersteunen moet de onderliggende database aangepast worden van het zaakstelsysteem. In dit document wordt beschreven hoe deze aanpassing gerealiseerd kan worden. Omdat op termijn het Excellence ZaakSysteem moet gaan voldoen aan de RGBZ en GEMMA ZTC 2.0 standaarden, inclusief een StUF interface, wordt eerst beschreven hoe geldigheden en historie hierin gemodelleerd zijn zodat hiermee rekening gehouden kan worden bij het opzetten van de zaaktype versionering.

2. Uitgangspunten

Voor het doorvoeren van versionering van het zaaktype zoals is beschreven in dit document zijn de volgende uitgangspunten gehanteerd:

1. De versionering van het zaaktype moet daar waar mogelijk conformeren aan het RGBZ / GEMMA ZTC 2.0. Dit uitgangspunt is noodzakelijk om te voorkomen dat bij het verder doorvoeren van het RGBZ / GEMMA ZTC 2.0 het zaakstelsysteem nogmaals fundamenteel aangepast moet worden.
2. Het zaaktype en het behandelproces vormen binnen het Excellence ZaakSysteem één geheel. Concreet betekent dit dat een kleine aanpassing van het zaaktype automatisch leidt tot een nieuwe (ongewijzigde) versie het behandelproces en vice-versa. Door met een draft versie te werken wordt, overigens, voorkomen dat er overmatig veel versies aangemaakt worden. Normaliter zullen een aantal wijzigingen opgespaard worden in een draft waarna die tegelijkertijd actief gemaakt worden.
3. Indien een entiteit aangepast wordt vanwege de zaaktype versionering is het verstandig te bepalen of er nog andere wijzigingen doorgevoerd moeten worden om de betreffende entiteit te conformeren aan het RGBZ / GEMMA ZTC 2.0.

Opmerkingen: fundamenteel uitgangspunt bij alle aanpassingen zou moeten zijn dat daar waar mogelijk zo snel mogelijk geconformeerd moet gaan worden aan de landelijk modellen. Concreet zorgen punten 1 en 3 er voor dat de sleutel van het zaaktype aangepast moet worden. Dit is echter een brekende wijziging met forse gevolgen waarvan de consequenties hoofdstuk 5 in uitgewerkt worden.

3. Historie en StUF

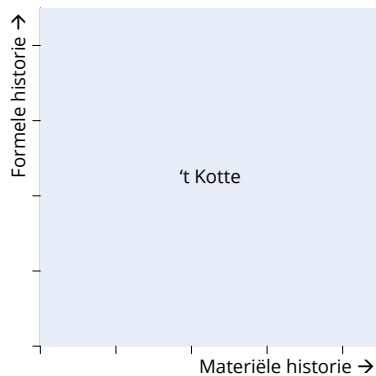
In alle KING standaarden die voorzien zijn van een StUF interface kunnen entiteiten voorzien zijn *formele* en/of *materiële* historie. De formele historie geeft aan wanneer in de administratie een verandering is verwerkt van een attribuutwaarde (bijvoorbeeld voor het registreren van correcties). De materiële historie geeft aan wanneer een verandering is opgetreden in de werkelijkheid die heeft geleid tot verandering van de attribuutwaarde (bijvoorbeeld een wijziging van het woonadres van een persoon). Kort gesteld betreft de materiële historie de *werkelijkheid* en is de formele historie *administratief*.

Omdat de formele en materiële historie van een entiteit in de tijd verweven zijn is het nuttig om, in de vorm van een voorbeeld, te demonstreren hoe deze gebruikt dienen te worden. Dit voorbeeld heeft betrekking op het woonadres van een inwoner die verhuist binnen een gemeente.

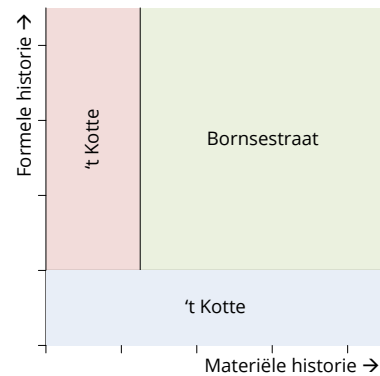
1. Inwoner woont op de 't Kotte in Hengelo (begin situatie);
2. De gemeente ontvangt een adreswijziging, het nieuwe adres is Bornsestraat;
3. Er wordt ontdekt dat de adreswijziging niet klopt, het correcte adres is Oldenzaalsestraat;
4. Er wordt nogmaals ontdekt dat de adreswijziging niet klopt, de persoon is *eerder* verhuist.

Deze eerste twee situaties zijn schematisch weergegeven in respectievelijk Figuur 1 en Figuur 2. In deze figuren zijn, in de vorm van een blokkendiagram, de twee vormen van historie in de tijd weergegeven. In de begin situatie (Figuur 1) zijn de formele en materiële historie gelijk en hebben beide geen einddatum. In Figuur 2 is te zien dat de materiële historie aangepast is, het 'oude' adres heeft een einddatum gekregen en er is een nieuw adres toegevoegd met dezelfde datum als begindatum (en geen einddatum). De formele historie blijft gelijk, het gaat hier immers *niet* om een correctie.

Om onderstaande figuren te begrijpen is het goed om in het achterhoofd te houden dat elk blok uiteindelijk als apart record opgeslagen wordt in een database. Concreet betekent dit dat een wijziging in de materiële / formele historie in sommige gevallen kan leiden tot de toevoeging van meer dan één record.

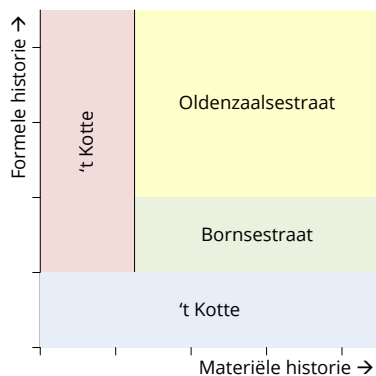


Figuur 1 - Situatie 1.

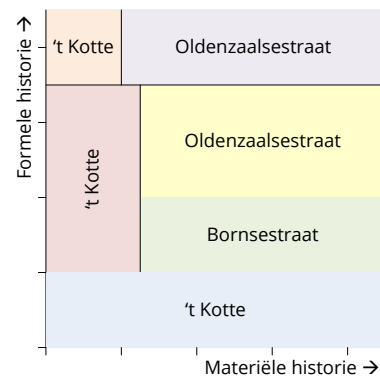


Figuur 2 - Situatie 2.

In Figuur 3 en Figuur 4 zijn schematisch de laatste twee situaties weergegeven waarin de ellende nog een tandje verder opgevoerd is.



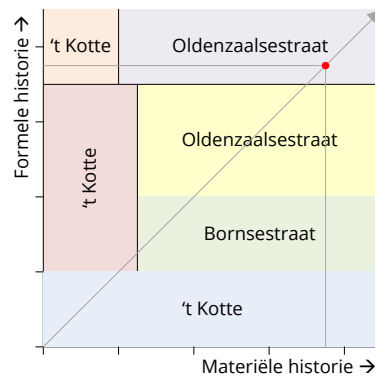
Figuur 3 - Situatie 3.



Figuur 4 - Situatie 4.

In Figuur 3 is te zien dat de formele historie aangepast is, er is een nieuw blok toegevoegd waarmee de foutieve materiële historie gecorrigeerd is. In Figuur 4 is te zien hoe de laatste, retrospectieve, aanpassing van de ingangsdatum van de voorgaande correctie verwerkt moet worden, hierdoor ontstaan twee nieuwe blokken. Voor degene die denkt dat dit een uiterst complexe manier is om wijzigingen te verwerken verwijs ik naar het "Representatie materiële en formele historie" document (deze is te vinden in Alfresco) waarin nog veel complexere scenario's beschreven worden (die allemaal ondersteund moeten worden).

Om de situatie te bepalen die op een bepaald moment gold moet in voorgaande figuren een lijn (verbinding van equivalente tijdstippen op horizontale en verticale assen) getrokken worden voor specifieke data in relatie tot de formele en materiële historie. Door een datum te prikken op deze lijn kan van links naar rechts de materiële van onder naar boven de formele historie afgeleid worden. Voor het bepalen van de exacte situatie op een bepaald moment moet dus gezocht worden naar het *enige* blok waarvoor geldt dat zowel de formele als materiële historie actueel waren (de rode punt in Figuur 5).



Figuur 5 - Bepaling historie.

Het principe zoals beschreven is in de voorgaande figuren moet uiteindelijk vertaald worden naar een database model. Voor de database modellering zijn een drietal metagegevens (i.e. standaard attributen die van toepassing zijn op elke entiteit met historie) noodzakelijk:

- `beginGeldigheid`: begin datum materiële historie (onvolledige datum). Dit metagegeven geeft aan vanaf welk tijdstip gegevens geldig zijn. Onder het geldig zijn van gegevens wordt verstaan, dat de eigenschappen van het object waarop de gegevens betrekking hebben op het begin van geldigheid de gegeven waarde hebben (c.q. hadden, indien het object op die datum ophield te bestaan). Het begin van geldigheid heeft alleen betrekking op de eigen gegevens van de entiteit en niet op eventuele relatie-entiteiten. Indien dit metagegeven niet aanwezig is wordt ervan uitgegaan dat de gegevens actueel zijn.
- `eindGeldigheid`: eind datum materiële historie (onvolledige datum). Dit metagegeven geeft aan tot welk tijdstip de gegevens geldig zijn. Onder het geldig zijn van de gegevens wordt verstaan, dat de eigenschappen van het object waarop de gegevens betrekking hebben vanaf begin van geldigheid tot eind van geldigheid de gegeven waarde hebben. Het eind van geldigheid heeft alleen betrekking op de eigen gegevens van de entiteit en niet op eventuele relatie-entiteiten. Indien dit element geen waarde heeft, dan wordt ervan uitgegaan dat de gegevens ook in de toekomst geldig zijn.
- `tijdstipRegistratie`: registratie datum/tijd formele historie. Dit metagegeven geeft aan op welk tijdstip de gegevens zijn geregistreerd door de zender van het bericht. Let op: er wordt geen aparte einddatum geregistreerd, deze is gelijk aan het `tijdstipRegistratie` van het *volgende* blok (mits aanwezig).

Bij deze metagegevens, die in de StUF berichten geretourneerd kunnen worden, dienen een aantal opmerkingen geplaatst worden die de mogelijke verwarring verderop in dit document hopelijk voorkomen. De `beginGeldigheid` en `eindGeldigheid` attributen hebben, ten eerste, een uiterst slecht gekozen naam omdat deze *niets* te maken hebben met de 'Begin datum geldigheid' en 'Eind datum geldigheid' attributen die soms voor een entiteit voorgeschreven worden (zoals zaaktype). In de StUF berichten worden deze laatste twee attributen standaard aangeduid met `ingangsdatumObject` en `einddatumObject`.

Het data type van `beginGeldigheid` en `eindGeldigheid`, ten tweede, is een *onvolledig datum*. Binnen de StUF berichten wordt dit data type gebruikt om aan te geven dat een deel van een datum feitelijk onbekend is. Dit is noodzakelijk om van een geboortedatum aan te geven, bijvoorbeeld, dat het jaar correct is maar de maand en dag niet. Hiervoor dient bij de datum (via `indOnvolledigeDatum`) één van de volgende waardes gespecificeerd te worden, waarbij 'volledig bekend' de standaard waarde is:

Waarde	Definitie
D	dag onbekend
M	maand (en dag) onbekend
J	jaar (maand en dag) onbekend
V	volledig bekend

Tabel 1 - Indicatie onvolledigheid waardes.

Bijkomend probleem is dat de onvolledige datum in een *string formaat* moet worden gespecificeerd: YYYYMMDD. Het gegeven dat jaar, maand en dag met nullen gevuld mogen worden zorgt voor een boeiend probleem; hoe kan bepaald worden of er overlap bestaat tussen twee blokken als de bijbehorende data onvolledig zijn? Wat te doen als er, bijvoorbeeld, snel op elkaar twee wijzigingen volgen met als *beginGeldigheid* de waardes 20130301 en 20130300? Wat niet in de StUF specificatie beschreven is, maar wat na lang zoeken op de StUF fora duidelijk is geworden, is dat deze situatie niet op kan treden omdat de opgegeven datum altijd een *geldige datum* moet zijn. Wat hiermee feitelijk wordt geforceerd is dat de *aanroepende partij* moet besluiten welke waardes ingevuld moeten worden voor de onbekende delen van de datum¹.

4. Versionering

De beschrijving van zaaktype in zowel het RGBZ als de GEMMA ZTC 2.0 is versie agnostisch. Hiertoe is besloten omdat versies eenvoudig kunnen worden ondersteund via de hiervoor beschreven historie. Echter, op dit punt bevatten de RGBZ en GEMMA ZTC 2.0 specificaties een ommissie; geen enkel attribuut heeft in één van beide specificaties een indicatie voor formele en/of materiële historie. Dit is op zich al vreemd, het doorlooptijd attribuut van statustype heeft materiële historie maar die van zaaktype niet. Daarnaast bevatten de StUF berichten voor het registreren / opvragen van een zaaktype de drie eerder beschreven metagegevens voor historie niet waardoor direct de vraag rijst hoe oude versies opgevraagd kunnen worden via StUF.

In de StUF fora is een voorstel gedaan in relatie tot historie voor zaaktypes. In dit voorstel staat letterlijk de volgende tekst "In het informatiemodel van de ZTC 2.0 is slechts beperkt historie gespecificeerd. Bedoeld is dat een zaaktype als geheel historie kent". Vooral de laatste zin wekt de nodige verbazing, blijkbaar kunnen entiteiten historie hebben zonder dat dit ergens in een specificatie expliciet vastgelegd is². Grote vraag is nu voor welke entiteiten binnen het RGBZ / de GEMMA ZTC 2.0 dit nog meer geldt. Geen van beide specificaties geven hier expliciet uitsluitel over, iets dat van een standaard toch verwacht zou mogen worden.

Indien formele en materiële historie moet worden toegevoegd rijst tevens de vraag of materiële historie en de geldigheidsduren (in de vorm van de 'Datum begin geldigheid zaaktype' en 'Datum einde geldigheid zaaktype' attributen) niet exact hetzelfde zijn. Indien uitgegaan wordt van een zaaktype dat als geheel van historie voorzien is lijkt deze laatste stelling waar te zijn. Waarna automatisch de volgende vraag rijst; wat is dan de toegevoegde waarde van de twee (extra) geldigheid attributen? De combinatie van deze twee attributen met historie zorgt er voor dat de 2-dimensionale blokdiagrammen van de voorgaande sectie feitelijk 3-dimensionale blokdiagrammen worden (waarbij de geldigheidsduur de derde dimensie vormt).

Concluderend kunnen we stellen dat vooral de GEMMA ZTC 2.0 specificatie vol leemtes en onduidelijkheden zit. In het algemeen geldt dat een entiteit voorzien kan worden van historie door de volgende stappen te doorlopen:

1. Toevoegen *volgnummer* aan de bestaande sleutel, deze vormt samen met de bestaande sleutel een (nieuwe) alternatieve sleutel voor de entiteit. De bestaande sleutel moet voorzien worden van een volgnummer om de diverse versies van elkaar te kunnen onderscheiden. Het volgnummer is een olopend intern nummer dat niet opgevraagd kan worden, deze is puur voor het correct houden van de interne administratie.
2. Toevoegen nieuwe numerieke sleutel (olopend in waarde) ter vervanging van de bestaande sleutel die in de voorgaande stap omgezet is naar een alternatieve sleutel.
3. Toevoegen materiële historie middels een onvolledige start- en einddatum en formele historie middels een datum/tijd.

¹ Een GBA geeft voor een geboortedatum waarvan slechts het jaar bekend is de waarde <jaar>0601 door.

² Dit soort gaten / inconsistenties leiden continu tot nieuwe vragen waardoor het schrijven van deze analyse aanzienlijk langer geduurd heeft dan gepland.

In Tabel 2 is weergegeven hoe een entiteit eruit ziet na het toepassen van deze regels. Overigens is het de bedoeling dat elke entiteit waar historie aan toegevoegd moet worden op bovenstaande wijze aangepast wordt. Het uitbreiden van de sleutel van een entiteit met een volgnummer heeft echter als consequentie dat alle tabellen die verwijzen naar deze entiteit ook aangepast moeten worden. Het grote nadeel van deze aanpassing is dan ook dat een dit soort wijzigingen 'uitvleken' naar andere entiteiten die, in potentie, weer kunnen 'uitvleken' naar weer andere entiteiten. Het toevoegen van historie aan één enkele entiteit kan grote gevolgen hebben.

Naam	Definitie	Formaat	Kard.	Opmerkingen
Id		N12	1-1	Oplopende sleutel.
<i>Sleutel</i>		...	1-1	Samengesteld alternatieve sleutel bestaande uit de originele sleutel van de entiteit en een volgnummer.
<i>Volgnummer</i>		N9	1-1	
...				
Begin geldigheid	Begin datum geldigheid materiële historie.	Date	1-1	
Indicatie onvolledigheid begin geldigheid	Indicatie onvolledigheid van de begin datum geldigheid materiële historie.	AN1	0-1	Toegestane waardes: D, M, J en V (zie Tabel 1). Moet ingevuld zijn als 'Begin geldigheid' een waarde heeft.
Einde geldigheid	Eind datum geldigheid materiële historie.	Date	0-1	
Indicatie onvolledigheid einde geldigheid	Indicatie onvolledigheid van de eind datum geldigheid materiële historie.	AN1	0-1	Toegestane waardes: D, M, J en V (zie Tabel 1). Moet ingevuld zijn als 'Einde geldigheid' een waarde heeft.
Tijdstipregistratie	Datum / tijd van registratie formele historie.	DateTime	1-1	

Tabel 2 – Entiteit met historie.

Omdat uit voorgaande beschrijving van historie in de StUF standaarden in relatie tot het RGBZ / GEMMA ZTC 2.0 niet eenduidig is gebleken dat de in Tabel 2 beschreven modellering noodzakelijk is lijkt het verstandig om eerst een variant hiervan te realiseren waarbij geen historie wordt toegevoegd. In deze opzet, zoals is weergegeven in Tabel 3, wordt alleen de geldigheidsduur gebruikt voor versieering.

Naam	Definitie	Formaat	Kard.	Opmerkingen
Id		N12	1-1	Oplopende sleutel.
<i>Sleutel</i>		...	1-1	Samengesteld alternatieve sleutel bestaande uit de originele sleutel van de entiteit en een volgnummer.
<i>Volgnummer</i>		N9	1-1	
...				
Ingangsdatum object	Begin datum geldigheid materiële historie.	Date	1-1	Let op: dit is het 'Datum begin geldigheid zaaktype' uit het RGBZ / GEMMA ZTC 2.0.
Indicatie onvolledigheid begin geldigheid	Indicatie onvolledigheid van de begin datum geldigheid materiële historie.	AN1	0-1	Toegestane waardes: D, M, J en V. Moet ingevuld zijn als 'Begin geldigheid' een waarde heeft.
Einddatum object	Eind datum geldigheid materiële historie.	Date	0-1	Let op: dit is het 'Eind begin geldigheid zaaktype' uit het RGBZ / GEMMA ZTC 2.0.
Indicatie onvolledigheid einde geldigheid	Indicatie onvolledigheid van de eind datum geldigheid materiële historie.	AN1	0-1	Toegestane waardes: D, M, J en V. Moet ingevuld zijn als 'Einde geldigheid' een waarde heeft.

Tabel 3 – Entiteit met alleen versieering.

Op de geldigheids data in deze tabel zijn de aanvullende regels zoals deze beschreven zijn in de aanleiding van deze memo van toepassing. Idealiter worden deze regels op het database niveau afgezekerd, dit om te kunnen garanderen dat deze onder alle omstandigheden afgedwongen worden.

5. Aanpassingen

Hieronder volgt een overzicht van de entiteiten, gegroepeerd op functionaliteit, die geraakt worden door de bovenstaande aanpassing. Per groep entiteiten wordt aangegeven wat de consequenties zijn en of deze al benoemd zijn in eerdere analyses / gesprekken.

5.1 Zaaktype

Naast de hiervoor beschreven aanpassingen die noodzakelijk zijn om versionering te ondersteunen voor de zaaktype entiteit zijn er nog een aantal restpuntjes die (optioneel) doorgevoerd kunnen worden om de huidige modellering te conformeren aan het RGBZ / de GEMMA ZTC 2.0:

- Aanpassen `description` kolom (zaaktype sleutel in het RGBZ) van 120 naar 80 karakters (mogelijk migratie issue).
- Toevoegen `identification` kolom (zaaktype sleutel in de ZTC 2.0). Het datatype van deze kolom is numeriek 5.

Omdat niet alle tabellen aangesloten moeten worden op de geversioneerde versie van het zaaktype moeten de volgende aanpassingen doorgevoerd worden:

- Toevoegen aparte `case_type_code` tabel waar entiteiten op kunnen koppelen met niet geversioneerde zaaktypes. Deze tabel hoeft slechts 1 kolom te bevatten `code`, `identification` of `description`. De keuze voor `code` in deze tabel heeft de minste impact, tabellen die moeten koppelen op deze tabel hoeven niet aangepast te worden, alleen de foreign keys hoeven hiervoor verlegd te worden. Nadeel hiervan is echter dat de zaaktype code een overblijfsel is van voorgaande zakenmagazijn versies. Dit attribuut staat noch in het RGBZ noch in de GEMMA ZTC 2.0. Om te kunnen voldoen aan deze modellen moet dit attribuut uiteindelijk verdwijnen. Het nu opnemen in de koppeltabel op te nemen zorgt ervoor dat er later meer werk verzet moet worden om hem te verwijderen.

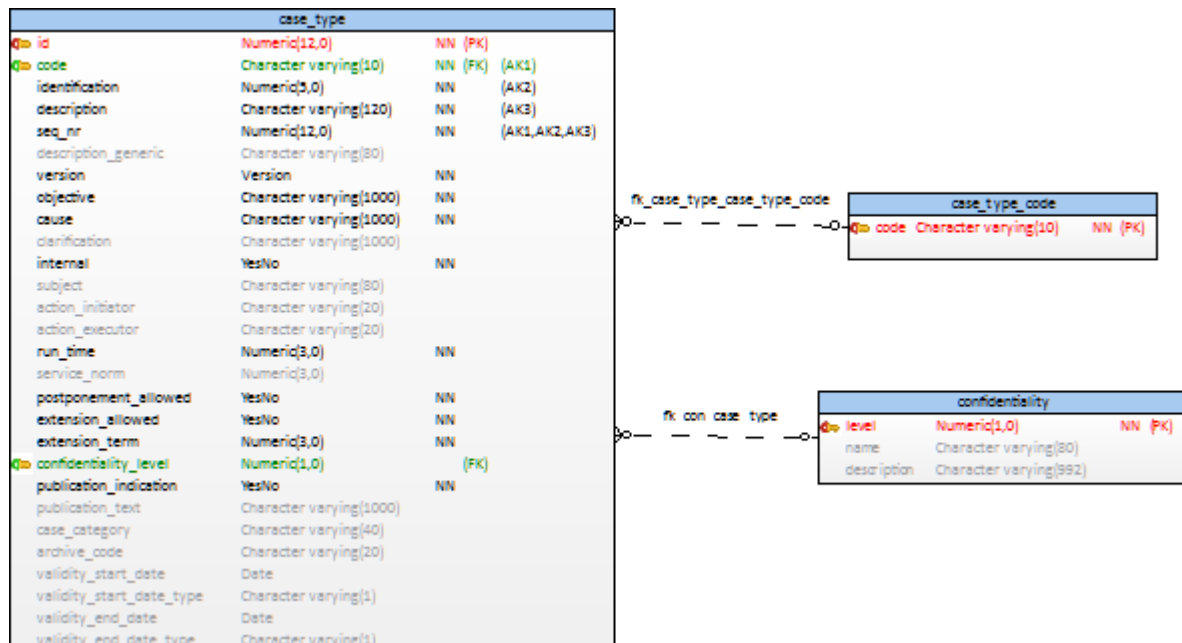
Omdat zaaktype-zaaktype relaties versie onafhankelijk zijn moeten de volgende aanpassingen gedaan worden aan zaaktype:

- Verwijderen `parent_case_type_code` deze dient afgesplitst te worden omdat zaaktype-zaaktype relaties niet versie gerelateerd zijn. De verdere uitwerking van dit punt is te vinden in sectie 5.2.

Ten slotte moet het zaaktype voorzien worden van de in sectie 4 beschreven attributen.

- Toevoegen `id` en `seq_nr` kolommen.
- Toevoegen alternatieve sleutels:
 - `code` en `seq_nr` (alternatieve sleutel Excellence);
 - `identification` en `seq_nr` (alternatieve sleutel GEMMA ZTC 2.0);
 - `description` en `seq_nr` (alternatieve sleutel RGBZ).
- Aanpassen `validity_start_date` en `validity_end_date` kolommen naar `date` (mogelijk migratie issue).
- Toevoegen `validity_start_date_type` en `validity_end_date_type` kolommen waarin aangegeven kan worden of, en welk deel van, de datum onvolledig is (waardenbereik: D, M, J, V). Deze kolommen mogen *alleen* gevuld zijn als de bijbehorende datum een waarde heeft.

De database uitwerking hiervoor is schematisch weergegeven in Figuur 6. Overigens dient bepaald te worden hoe de kolommen voor materiële en formele historie op database niveau structureel genoemd gaan worden, de hierboven gekozen namen kunnen tot verwarring leiden.



Figuur 6 – Zaaktype met versiebeheer.

De volgende gerelateerde entiteiten moeten gekoppeld worden op het versie onafhankelijk zaaktype (`case_type_code`):

- Het autorisatiemodel;
- Het externe systeem dat standaard aan het zaaktype is gekoppeld (`case_type_system`);
- De standaard uitvoerende;
- De zaakstelsysteem synchronisatie.

De volgende gerelateerde entiteiten moeten gekoppeld worden op het versie afhankelijk zaaktype:

- Eigenschapsgroepen inclusief eigenschappen;
- Staotypes;
- Resultaat types;
- Document types;
- Overige GEMMA ZTC 2.0 entiteiten:
 - Trefwoord
 - Product / Dienst
 - Formulier
 - Referentieproces
 - Verantwoordingsrelatie
- De XML naar eigenschappen conversie.

Het uitgangspunt is dat alle versie afhankelijke entiteiten gekopieerd worden van de huidige versie als er een draft versie gemaakt wordt voor een zaaktype. Voor de genoemde entiteiten kunnen in deze opzet geen nieuwe versies gemaakt worden onafhankelijk van het zaaktype. Als het zaaktype actief is zijn logischerwijs hierdoor ook alle onderliggende entiteiten actief. Er is dan ook geen noodzaak om hiervoor aparte geldigheidsduren vast te leggen.

5.2 'Ouder' zaaktype

In de GEMMA ZTC 2.0 kunnen een drietal nieuwe relaties vastgelegd worden tussen zaaktypes, analoog aan het 'ouder' zaaktype uit het Excellence ZaakSysteem 5.0. Deze relaties, inclusief hun kardinaliteit zijn weergegeven in Tabel 4. In deze tabel geven de FH en MH kolommen aan of er formele of materiele historie gespecificeerd is voor de betreffende relatie, indien er een J staat in één van deze kolommen moet de relatie zelf ook van versiehistorie voorzien worden.

Relatiennaam	Definitie	FH	MH
Zaaktype [0..1] is bronzaaktype van zaaktype [0..1]	Het zaaktype binnen de catalogus waaraan dit zaaktype is ontleend.	N	N
Zaaktype [0..*] is deelzaaktype van zaaktype [0..*]	De zaaktypen (de zaaktypen van de hoofdzaken) waaronder zaken van dit zaaktype als deelzaak kunnen voorkomen.	N	N
Zaaktype [0..*] heeft gerelateerd zaaktype [0..*]	De zaaktypen van zaken die relevant zijn voor zaken van dit zaaktype.	N	N

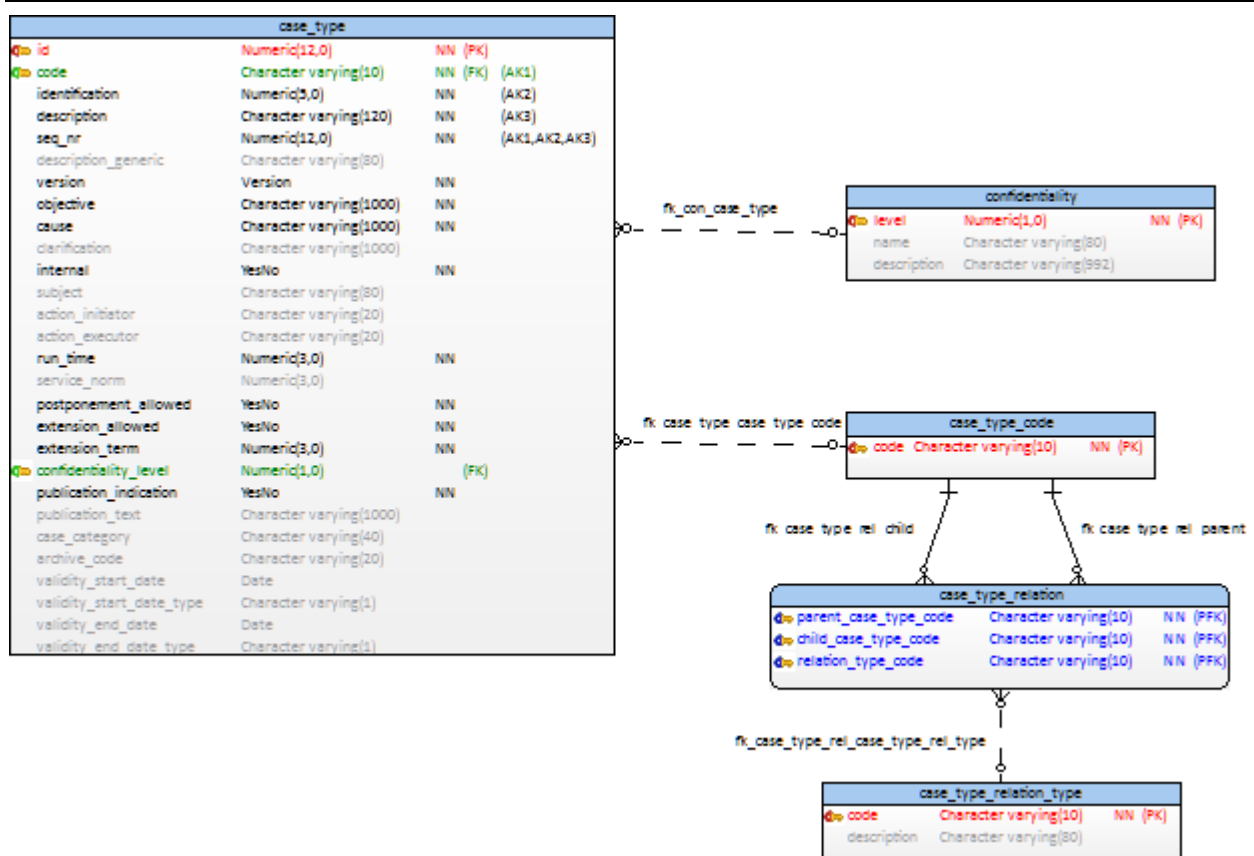
Tabel 4 – GEMMA ZTC 2.0 zaaktypen relaties.

De laatste relatie is voorzien van een aantal attributen (zie de relatieklasse ZaakTypenRelatie in de GEMMA ZTC 2.0 specificatie); deze zijn weergegeven in Tabel 5.

Naam	Definitie	Formaat	Kard.	Opmerkingen
Aard relatie	Omschrijving van de aard van de relatie van zaken van het ene zaaktype tot zaken van het andere zaaktype	AN15	1-1	
Toelichting	Een toelichting op de aard van de relatie tussen beide zaaktypen.	AN255	1-1	

Tabel 5 – Zaak typen relatie attributen.

De in Tabel 4 beschreven bron- en deelzaaktype relaties alsmede de 'ouder' zaaktype kunnen gemodelleerd worden als een zaaktypen relatie (zoals weergegeven in Tabel 5) met twee voorgedefinieerde waardes voor aard / toelichting. In Figuur 7 is schematisch weergegeven hoe dit in de database opgelost kan worden.



Figuur 7 - Generieke zaaktypen relaties.

5.3 Zaak

Elke zaak moet gekoppeld worden aan een zaaktype met een specifiek volgnummer, alleen op deze wijze kan de correcte versie van het behandelproces gevonden en uitgevoerd worden. Normaliter zou voor het realiseren van deze aanpassing het zaaktype code veld vervangen worden door zaaktype id. Bij het aanmaken van de zaak hoeft de aanroeper, net zoals nu het geval is, alleen de zaaktype code invullen. In het zaakstelsel moet bij het registreren van de zaak het id van de actieve zaaktype opgezocht worden waarna deze in de database geregistreerd kan worden. Omdat de zaak entiteit in een groot aantal componenten gebruikt wordt is het onverstandig om de code te verwijderen, het voorstel is dan ook om zowel de zaaktype code als het id vast te leggen.

Bij het aanpassen van de zaak entiteit zouden tevens de volgende GEMMA ZTC 2.0 velden toegevoegd kunnen worden die momenteel nog ontbreken:

Veld	Formaat	Omschrijving
Betalingsindicatie	AN12	Domeinwaardes: <ul style="list-style-type: none"> N.v.t. (Nog) niet Gedeeltelijk Geheel
Laatste betaaldatum	datum/tijd (JJJJMMDDHHMM)	