

# Technisch Memo

**Auteur:** A.K. Boerma  
**Datum:** 24-11-2013  
**Titel:** Referentie zaaktype  
**JIRA:** ES-38

## 1. Aanleiding

Bij het implementeren van bepaalde processen in het Excellence ZaakSysteem is het soms noodzakelijk zijn om meerdere, vergelijkbare, zaaktypes in te richten. Een voorbeeld hiervan is 'melding openbare ruimte', een zaaktype dat voor verschillende soorten meldingen (huisvuil, groen, etc.) in hoofdlijnen hetzelfde afgehandeld moet worden. In de huidige versie van het Excellence ZaakSysteem moet momenteel het meldingen zaaktype, alsmede alle sub-zaaktypes, apart ingevoerd worden.

Omdat dit een niet bepaald gebruikersvriendelijk werkwijze is wordt versie 5.1 het Excellence ZaakSysteem voorzien van ondersteuning voor *referentie zaaktypes*. Een referentie zaaktype is een standaard zaaktype waarvan de definitie, op een paar instellingen na (zie uitgangspunt 3), op het moment van activatie gekopieerd wordt naar de *afgeleide zaaktypes*. Door gebruik te maken van een referentie zaaktypes kunnen snel grote aantallen zaaktypes gecreëerd en gewijzigd worden. De inregeling van een referentie zaaktype omvat minimaal de volgende gegevens:

- Autorisaties
- Eigenschapgroepen / eigenschappen (inclusief XML conversie)
- Document types
- Resultaat types met bijbehorende besluit types
- Trefwoorden
- Externe systemen

Door het gebruik van referentie zaaktypes wordt tevens een ander probleem opgelost; het *wijzigen* van het type van een zaak. Naarmate er meer zaaktypes ingericht worden binnen het Excellence ZaakSysteem wordt het lastiger om het type van een zaak tijdens de behandeling aan te passen. Het fundamentele probleem hierbij dat het converteren van een *behandelproces* (in de vorm van eigenschappen, stappen, e.d.) niet automatisch uit te voeren is zelfs als twee zaaktypes behoorlijk op elkaar lijken. Dit zou opgelost kunnen worden door de medewerker een scherm te geven waarmee hij de conversie zelf kan inregelen; dit is echter complex en foutgevoelig. De invoering van referentie zaaktypes lost dit probleem op doordat ten allen tijde geswitched kan worden tussen zaaktypes die afgeleid zijn van hetzelfde referentie zaaktype (de behandelprocessen zijn in dit geval immers identiek).

Er zijn twee mogelijke manieren om de koppeling tussen een zaaktype en een referentie zaaktype te realiseren:

1. Verwijzen: het afgeleide zaaktype bevat een *verwijzing* naar het referentie zaaktype. Bij het opvragen van gegevens van het zaaktype moeten deze via het referentie zaaktype opgehaald worden.
2. Kopiëren: het afgeleide zaaktype is een *kopie* van het referentie zaaktype. Indien er een nieuwe versie van een referentie zaaktype ontstaat moet deze gepropageerd worden naar alle afgeleide zaaktypes.

De eerste optie is architecturaal gezien de fraaiste omdat hierbij de gegevens die opgeslagen worden bij de afgeleide zaaktypes minimaal is. Praktisch gezien is deze opzet echter een stuk minder omdat een flink aantal operaties voorzien moet worden van logica waarin gecontroleerd wordt of er een referentie zaaktype ingesteld is om vervolgens daar de relevante gegevens vandaan te halen. Dit zal vooral voor autorisaties een probleem zijn

omdat de verificatie hiervan voor de lees en zoek operaties in database queries opgelost gaat worden. Het schrijven van queries met conditionele sub-selecties is complex en leidt waarschijnlijk to performance problemen.

De tweede optie is architectureel minder gezien minder maar pragmatisch gezien veel beter. Iedere keer dat er een nieuwe versie van een referentie zaaktype geactiveerd wordt moet de definitie hiervan gekopieerd worden naar alle afgeleide zaaktypes. Belangrijkste voordeel van deze opzet is dat de bestaande operaties hiervoor niet of minimaal aangepast hoeven worden.

## 2. Uitgangspunten

1. Een referentie zaaktype is een *standaard zaaktype*. In de tijdens de bootcamps besproken opzet was dit niet toegestaan omdat hierin de afgeleide zaaktypes slecht *verwijzingen* waren naar het referentie zaaktype. Deze is opzet echter is later aangepast omdat de invoering van het autorisatiemodel hierdoor nodeloos complex werd (zie ook punt 2) waardoor dit uitgangspunt is komen te vervallen.
2. Een referentie zaaktype kan niet afgeleid zijn van een ander referentie zaaktype. Concreet betekend dit dat hiermee structuur van *twee* niveau's gedefinieerd kan worden. Opgemerkt dient te worden dat meer niveau's onnodig zijn; een zaaktype dat afgeleid is van een zaaktype dat afgeleid is van een referentie zaaktype is per definitie gelijk aan dat laatste zaaktype.
3. Een (draft) referentie zaaktype wordt bij activatie *gekopieerd* (eigenschappen, stappen, e.d.) naar alle afgeleide zaak types waardoor nieuwe versies deze zaaktypes ontstaan die, simultaan met het referentie zaaktype, actief worden. De aan te maken afgeleide zaaktypes (die zelf geen referentie zaaktype mogen zijn) moeten geselecteerd kunnen worden bij het activeren van het referentie zaaktype. Dit uitgangspunt is noodzakelijk om te voorkomen dat de afgeleide zaaktypes verschillende geldigheidsduren krijgen waardoor zaaktype conversie onmogelijk wordt (wat bijvoorbeeld te doen als het doel zaaktype van een conversie niet actief is).

Nadeel van deze opzet is dat aanpassingen van de instellingen van de afgeleide zaaktypes alleen uitgevoerd kunnen worden bij activatie van het referentie zaaktype. Dit is het enige moment om deze instellingen aan te passen omdat de afgeleide zaaktypes hierna actief zijn en niet aangepast moegen worden. De aan te passen instellingen van afgeleide zaaktypes zijn:

- Doorlooptijd behandeling
- Servicenorm behandeling

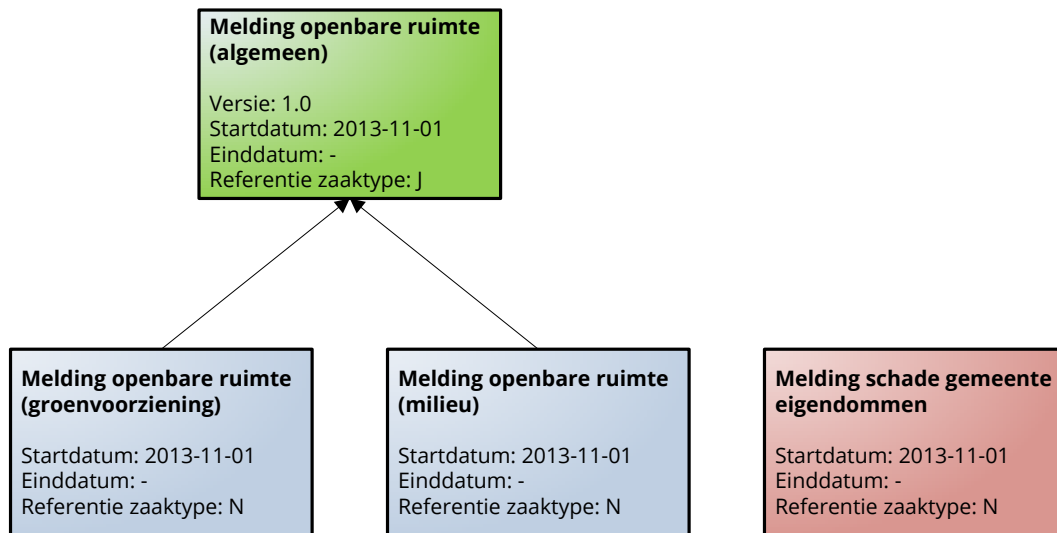
4. Een afgeleid zaaktype kan alleen aangepast worden door de volgende handelingen uit te voeren:
  - a. Ontkoppel het zaaktype van het referentie zaaktype.
  - b. Maak draft aan voor het ontkoppelde zaaktype.
  - c. Pas de draft versie van het ontkoppelde zaaktype aan.
  - d. Activeer het aangepaste ontkoppelde zaaktype.
5. Het is niet mogelijk om zaaktype later alsnog af te leiden van een actief referentie zaaktype. De enige wijze om een zaaktype toe te voegen is door een nieuw draft versie van het referentie zaaktype aan te maken om vervolgens het nieuwe zaaktype toe te voegen als deze geactiveerd wordt (zie 3). Alleen op deze manier kan gegarandeerd worden dat alle afgeleide zaaktypes dezelfde geldigheidsduur hebben.
6. Het zaaktype van een zaak mag allen verandert worden in een zaaktype dat afgeleid is van dezelfde versie van hetzelfde referentie zaaktype.

### 3. Opmerkingen

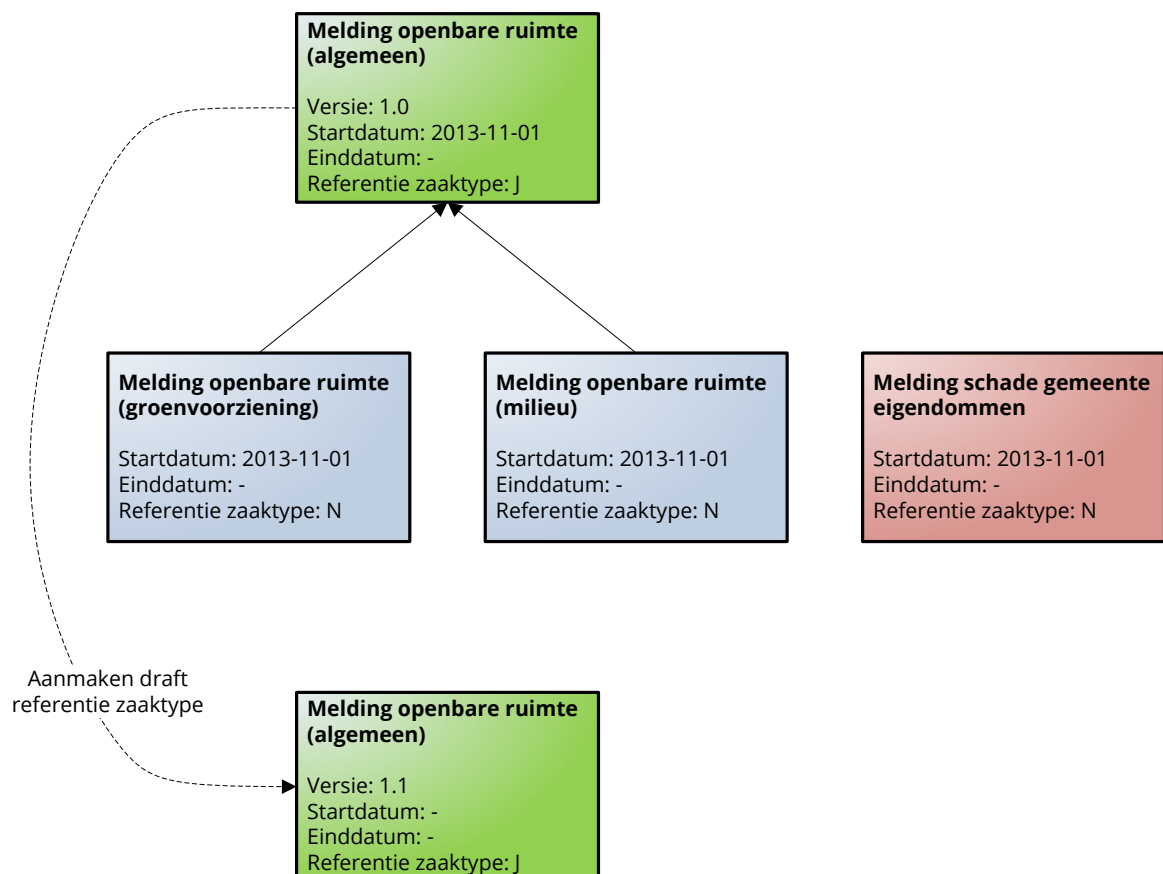
- Het zou verstandig zijn om een referentie zaaktype's als zodanig te registreren in middels een nieuw attribuut. Dit maakt controle van een aantal regels (voornamelijk 2 en 3) een stuk eenvoudiger;
- Het zou praktisch zijn als bij het activeren van een *draft* referentie zaaktype de lijst met afgeleide zaaktypes van de *actieve* versie gepresenteerd zou worden. De aan te passen instellingen krijgen hierdoor de actieve waardes als startpunt.
- De relatie referentie zaaktype – afgeleid zaaktype is *versie afhankelijk*. Dit vereist een database aanpassing omdat zaaktype relaties dat momenteel niet zijn. Deze relatie is tevens noodzakelijk om te kunnen verifiëren of een bepaalde zaaktype wijziging toegestaan is.
- De initieel verantwoordelijk is niet in te stellen bij het activeren van het referentie zaaktype, deze is namelijk in de huidige opzet voor alle versies van het zaaktype gelijk. Indien dit niet gewenst is vereist dit een aanpassing.

## 4. Illustratie

Aan referentie zaaktype J dient een nieuw afgeleid zaaktype te worden toegevoegd.  
Allereerst wordt het nieuwe zaaktype als een normaal zaaktype aangemaakt, naast het referentiezaaktype:



Vervolgens wordt een nieuwe draft versie van het referentie zaaktype aangemaakt:



Daarna kan het nieuwe afgeleide zaaktype worden gekoppeld aan het draft referentie zaaktype, waarna het referentie zaaktype kan worden geactiveerd voor gebruik:

