Inhoud

[1 Versiebeheer 1](#_Toc157090466)

[2 Distributie 1](#_Toc157090467)

[3 Conclusie 2](#_Toc157090468)

[4 Het gebruik van de repository binnen Azure DevOps 2](#_Toc157090469)

[5 Het gebruik van de Naamgevingsstandaarden 3](#_Toc157090470)

[6 De manier waarop in IDA gemapt wordt 3](#_Toc157090471)

[7 Het gebruik van de rapportagefunctie in IDA 3](#_Toc157090472)

[8 Het summiere gebruik van het tabblad [Documentation] 4](#_Toc157090473)

[9 Aanbevelingen en verzoek 4](#_Toc157090474)

[9.1 Aanbevelingen 4](#_Toc157090475)

[9.2 Verzoek 5](#_Toc157090476)

# Versiebeheer

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Datum | Wijzigingen | Door |
| 0.1 | 24-01-2024 | Concept | Claudia Watrin |
| 0.2 | 25-01-2024 | Eerste feedback verwerkt |  |
|  |  |  |  |
|  |  |  |  |

# Distributie

|  |  |  |
| --- | --- | --- |
| Versie | Datum | Naar |
| 0.1 | 24-01-2024 | Eerste review naar : Millo Korving |
| 0.2 | 25-01-2024 | Tweede review naar : Richard Hogenberg, Henri Mijnders, Henry Niessen, Alwin van der Veen, Millo Korving |
|  |  |  |
|  |  |  |

# Conclusie

Op verzoek van Erik Paardekooper heb ik gekeken naar de manier waarop het team van Hot Item (HI) IBM InfoSphere Data Architect (IDA) heeft gebruikt en ingericht bij de bouw van ZW-ARBO.

Dit team gebruikt IDA op een significante andere wijze dan dat wij binnen het Data‑IntegratieMagazijn (DIM) IDA gebruiken.

De belangrijkste verschillen die ik ben tegengekomen zijn:

1. Het gebruik van de repository binnen Azure DevOps m.b.t. de projecten in IDA (§ 4).
2. Het gebruik van de Naamgevingsstandaarden (§ 5).
3. De manier waarop in IDA gemapt wordt (§ 6).
4. Het gebruik van de rapportage-functie in IDA, waarbij een rapport uit IDA over de logische mapping wordt uitgedraaid. Dit rapport dient als basis voor een Python-script, dat een .ddl genereerd (§ 7).
5. Het summiere gebruik van het tabblad [Documentation] in de [Properties] van zowel de entiteiten, attributen en mappings (§ 8).

Meer hierover in de paragrafen hieronder.

Als laatste paragraaf zijn aanbevelingen (en een verzoek) opgenomen.

# Het gebruik van de repository binnen Azure DevOps

De DIM-ModellenRepository (DMR) is in principe per laag (Algemene map [10 DIM], [20 Bronzone], [30 Integratiezone], [40 Bedrijfszone] en [50 Gegevensvenster]) opgebouwd. In iedere map staan submappen per schema. In [20 Bronzone] komen de ontkoppelviews te staan. Deze submappen komen overeen met de projecten die in IDA zijn gebruikt.

Op deze manier kan iedereen de laatste versie van een ontkoppelview IDA binnentrekken. Of, wanneer nodig, een integratie-object vanuit [30 Integratiezone], dat op deze manier makkelijk hergebruikt kan worden.

De HI repository is op een heel andere wijze ingedeeld. Op het moment dat Hot Item begon te werken aan ZW-ARBO was de werkwijze voor het versiebeheer in Azure DevOps nog niet uitgewerkt. Integratie met DIM way of work lijkt hierdoor niet direct mogelijk, zonder extra effort. Vanuit architectureel standpunt is er een optie:

1. integratie met extra effort  
   waarbij gekeken moeten worden naar een inschatting van deze effort
2. as-is overnemen  
   met extra leertraject effort voor degenen die het onderhoud overnemen. Denk daarbij aan zowel development als deployment proces en operationeel onderhoud..

Optie 2 is het beste, indien prio 1 is op z.s.m. in productie nemen (uiteraard met inachtneming van succesvolle acceptatietest).

De indeling die het DIM gebruikt is gebaseerd op de IDA-projecten, die ieder weer specifiek voor één schema zijn.   
Binnen dit project (dat de schemanaam draagt) is alle voor dit specifieke schema, benodigde documentatie opgenomen. Mappen kan over meerdere projecten plaatsvinden. Het voordeel hiervan is dat wijzigingen aan een schema slechts op één plek bijgehouden worden. Wordt dit schema in een ander project gebruikt, dan hoeft er niet binnen meerdere projecten een update plaats te vinden.

Ook hier gelden bovenstaande opties. En ook hier zal optie 2 de meest geschikte optie zijn, wanneer ZW-ARBO z.s.m. in productie genomen dient te worden.

# Het gebruik van de Naamgevingsstandaarden

Op het gevaar af dat ik in herhaling val. Ten tijde dat HI begon met het werken aan ZW-ARBO was er nog veel onduidelijkheid over de naamgevingsstandaarden.   
Daarnaast zijn de aan Hot Item verstrekte standaarden, die voor de DIM-objecten, m.a.w. die voor de technische benamingen. Ik heb de namen langs de naamgevingsstandaarden voor de functionele namen gehouden omdat door HI vanuit het logisch model werkt. Hierdoor voldoet veel niet aan onze huidige standaarden.

Reden te meer om voor optie 2 te gaan, vanwege de verwachte effort.

# De manier waarop in IDA gemapt wordt

Zoals eerder genoemd werkt het team van HI in het logische model. Op dat niveau wordt ook gemapt.  
Het grootste verschil met de wijze waarop HI mapt, vergeleken met het DIM, ligt in het feit dat Hot Item meerdere lagen binnen één mapping verwerkt. Dit heeft tot gevolg dat dezelfde schema’s aan zowel de bronkant, als aan de doelkant worden gebruikt.

Vanuit HI is besloten om zoveel mogelijk business rules op 1 plek te implementeren en te hergebruiken, ter voorkoming van redundantie. Dit betrof de views op de al door het DIM gebouwde integratie-objecten (IMF- en UZS-objecten) en volgens HI is dit alleen in exceptionele situaties te vinden.

De logische mappings worden als input voor het python script gebruikt om extension mapping documents te genereren (fysieke mappings) voor vastlegging in IGC. Zo kan de design lineage (horizontaal) gerapporteerd worden over de data flow op fysieke niveau van bron naar target kolom op basis van het mapping ontwerp. Er wordt namelijk met generieke jobs gewerkt tijdens het operationele data proces.

# Het gebruik van de rapportagefunctie in IDA

HI maakt gebruik van de rapportagefunctie in IDA. Het team heeft een rapport gemaakt op basis waarvan, m.b.v. een Pythonscript de .ddl gegenereerd wordt.

Deze werkwijze werkt sterk af van de manier waarop het DIM werkt, waarbij het logisch model getransformeerd wordt tot een fysiek model. Dit fysieke model wordt aangevuld met technische velden en kenmerken. Zodra dit het fysieke model is aangevuld, wordt vanuit IDA een .ddl gegenereerd.

De belangrijkste reden waarom Hot Item heeft gekozen voor bovenstaande werkwijze is het terugbrengen van handmatige invoer en het scheiden van DDL per tabel.

# Het summiere gebruik van het tabblad [Documentation]

Tijdens het nalopen van de entiteiten en attributen viel het summiere gebruik van het tabblad [Documentation] in de [Properties] zowel de entiteiten, attributen en verschillende soorten mappings op.

Voorbeelden:

* In de [Documentation] voor [BDV-STG.ldm/Prestage Zaak/Datum Boeking Dienstverband] staat: *“Blijft nu bewust even leeg.”*  
  Deze opmerking werpt alleen maar vragen op:
  + Waarom blijft dit veld nu leeg?
  + Aan welke voorwaarden moet voldaan worden om dit veld te vullen?
* In [BDV-STG.ldm/Zaak UZS View] komt een attribuut met de naam [Dummy] voor. Onduidelijk is wat dit voor attribuut is. Waarschijnlijk is het een technisch veld. Graag gerichte informatie opnemen in de [Documentation].
  + Wat voor (technische beslissing) ligt ten grondslag aan het opnemen van dit veld in het logisch model?
  + Waar wordt het precies voor gebruikt?
  + Is dit gebruik context afhankelijk?
* In dezelfde entiteit wordt het attribuut [Mapping documentatie] opgevoerd. Duidelijk is waarom dit attribuut in het verleden is opgevoerd. Wanneer dit attribuut zijn functie heeft verloren, graag verwijderen.

# Aanbevelingen en verzoek

## Aanbevelingen

Naar aanleiding van bovenstaande bevindingen, heb ik de volgende aanbevelingen:

* De wijze waarop het product ZW-ARBO is opgebouwd wijkt voldoende af om een eigen repository op Azure DevOps Versiebeheer (ADV) te verdedigen. Het wijzigen van de structuur op ADV om te voldoen aan de eisen waarop het DIM met ADV omgaat levert niet voldoende op om af te dwingen.
* Hetzelfde geldt voor de manier waarop HI werkt met de projecten binnen IDA.
* Zodra ZW-ARBO wordt herbouwd door het DIM, zal de herbouw wel aan deze eisen moeten voldoen.
* Wat betreft de naamgevingsstandaarden raad ik aan om toch alles langs de lat van de DIM‑standaarden te leggen. Dit om aansluiting te krijgen op dat wat al in DIM gebouwd is en wordt.  
  De geldige standaarden zullen meegeleverd worden met dit rapport.
* Een duidelijk uitleg over de manier van mappen dient in een project opgenomen te worden. Het liefst niet als apart document buiten IDA, maar als [Other File] binnen het desbetreffende project binnen IDA.
* [Documentation] in de [Properties] van entiteiten, attributen en mappings, waarbij er geen sprake is van een 1-op-1 overgaan, moet voorzien zijn van een beknopte omschrijving en eventuele ontwerpbeslissingen.
* Indien python onderdeel is van deployment naar een target omgeving is dit een aspect om te verrijken in het deploy proces. Extension mapping documents zijn op dit moment geen onderdeel van het DIM deployment proces. Indien as-is overnemen van HI assets het geval is, dan dient ook het deployment proces helder beschreven te zijn vanuit HI.

## Verzoek

De rapportagefunctie zoals die door Hot Item gebruikt wordt, wordt door ons nog niet gebruikt. Dit is echter wel een wens van ons. Een uitleg hoe de rapportagefunctie gebruikt kan worden wordt op prijs gesteld.