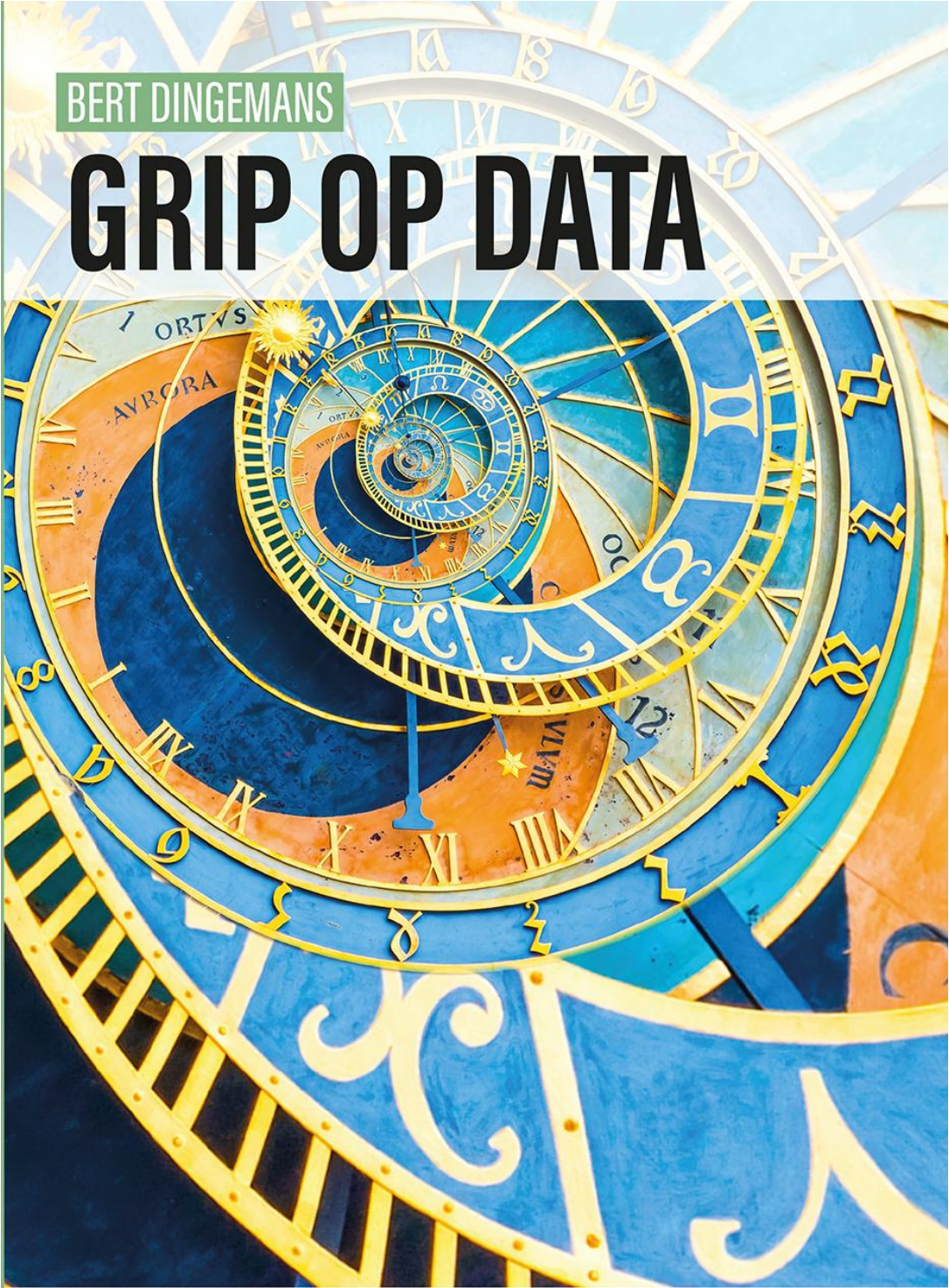




BERT DINGEMANS

GRIP OP DATA

InterActory
Grip op data

Indelen van repository content

Inleiding

Sparx Enterprise Architect is een multi-modelleertaal-tool dat gebaseerd is op een repository. Dat betekent dat de inhoud van een repository wordt opgeslagen in een onderliggende relationele database. In veel panelen van Sparx is dat duidelijk terug te zien. In een aantal panelen zie je een duidelijke detailweergave, zoals het property paneel of in een lijst weergave zoals bijvoorbeeld het relaties paneel.

Toch is met name het browser paneel in Sparx Enterprise Architect een belangrijk hulpmiddel om de inhoud van de repository voor de gebruikers te structureren in de vorm van een boomstructuur.

Regelmatig krijg ik vragen uit organisaties die ik ondersteun bij het gebruik van Sparx Enterprise Architect over hoe de repository inhoud het best ingedeeld kan worden op basis van packages (in met name de browser). Dat is een valide vraag want met name als de repository inhoud in omvang groeit of complexer wordt doordat er meerdere modelleerdomeinen worden uitgewerkt in een repository wordt de indeling relevant.

Voor het indelen van de repository content is geen eenduidige logische indeling te noemen. Dat hangt van veel factoren af in een organisatie. Wat is de structuur van de organisatie, welke rollen gebruiken de repository, welke modelleertalen worden toegepast en wat is het doel van een repository. Allemaal vragen die meespelen in het bepalen van een repository indeling die aansluit bij de stakeholders die betrokken zijn bij een modelleer repository.

In dit document ga ik in op verschillende dimensies voor het indelen van een repository en geef ik voorbeelden hoe een dergelijke indeling eruit kan zien. Naast enkelvoudige domeinindelingen zal ik ook ingaan op een meer hybride indeling van de repository inhoud.

Bij dit document hoort een filebased repository waarin de voorbeelden te zien zijn van verschillende indelingen. Deze filebased repository is te downloaden via <https://sparx-wpp.nl/upload/repositoryindeling.zip>.

Indelen repository

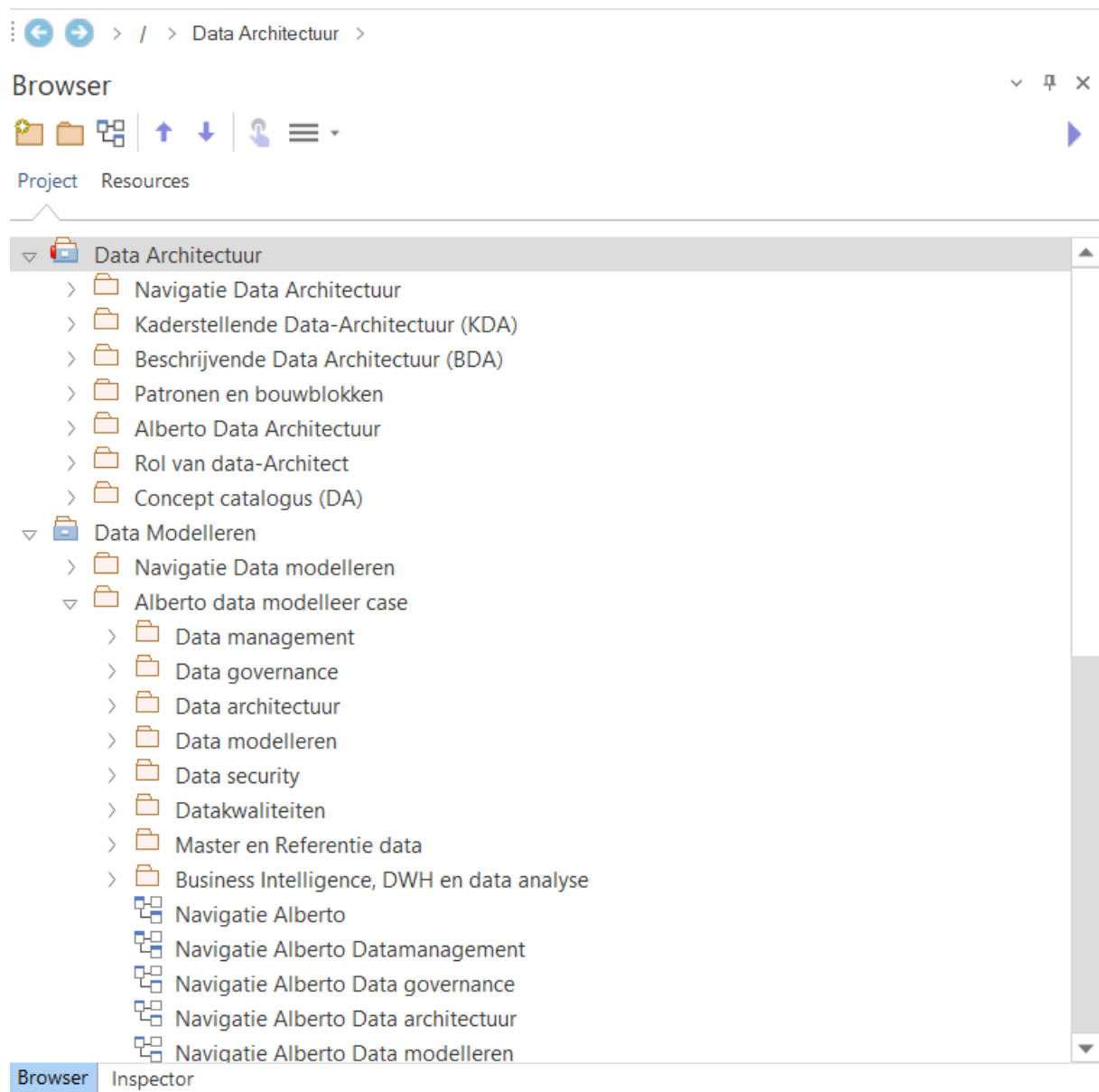
Zoals al aangegeven is de repository indeling gebaseerd op een relationele database, dus opgebouwd uit tabellen, rijen en kolommen. Toch wordt virtueel meer gewerkt op basis van een boomstructuur. Deze boomstructuur is vergelijkbaar met een filesysteem opgebouwd uit directories en files van verschillende types.

De boomstructuur in een repository heeft grote overeenkomsten met het filesysteem maar kent een aantal extra concepten, zoals documenten en diagrammen wat de indeling iets complexer maakt. Hiervoor is in Sparx Enterprise Architect het Browser paneel geïntroduceerd dat ervoor zorgt dat de database inhoud in de vorm van een verkenner weergave (boomstructuur) de inhoud van de repository op een logische wijze aan de gebruikers presenteert.

Browser

De browser is misschien wel het meest gebruikte paneel bij het gebruik van Sparx Enterprise Architect. Het biedt de logische indeling van de repository inhoud op basis van een zelf te definiëren indeling van de inhoud. Met name de modelleers zullen veelvuldig gebruik maken van de browser. De browser is ook de meest kenmerkende weergave van het onderwerp van dit document de indeling van de repository inhoud.

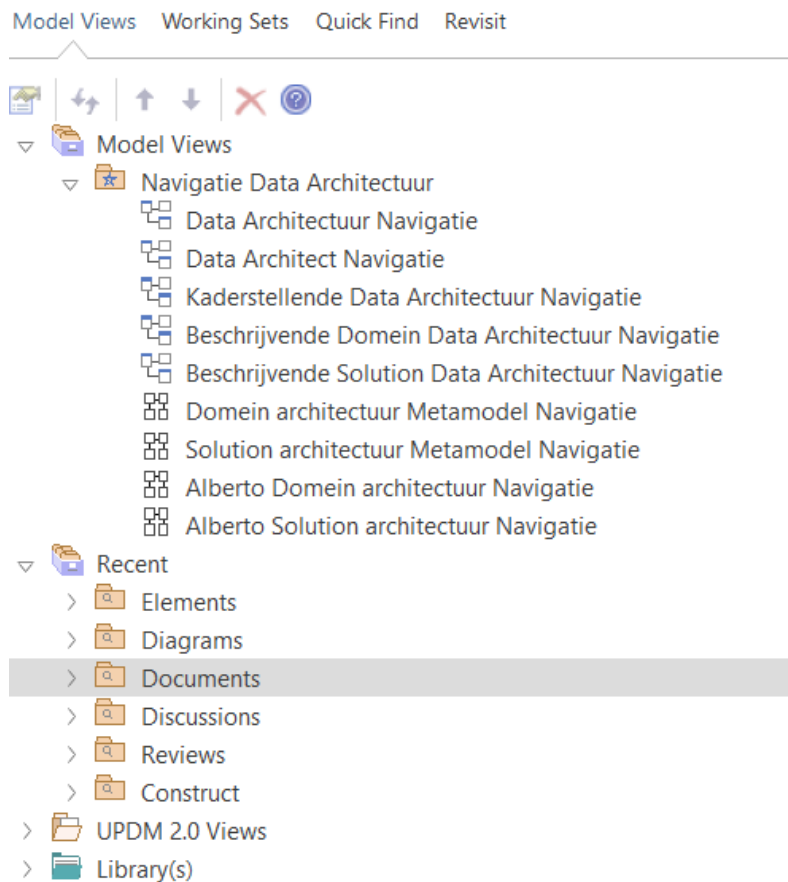
We zullen dan ook het browser paneel steeds als voorbeeld weergave gebruiken van de onderwerpen die hieronder aan de orde komen.



In de afbeelding zie je een voorbeeld van de browser weergave met daarin de packages die zorgen voor een boomstructuur om de inhoud in te delen in niveaus. Daarnaast zie je ook al een stuk inhoud namelijk een aantal diagrammen die zich in een package bevinden. Voor elementen en documenten geldt eenzelfde aanpak. In de afbeelding zie je ook een aantal knoppen in de browser om je repository inhoud anders te structureren en uit te breiden met nieuwe concepten.

Doelgroep

Focus



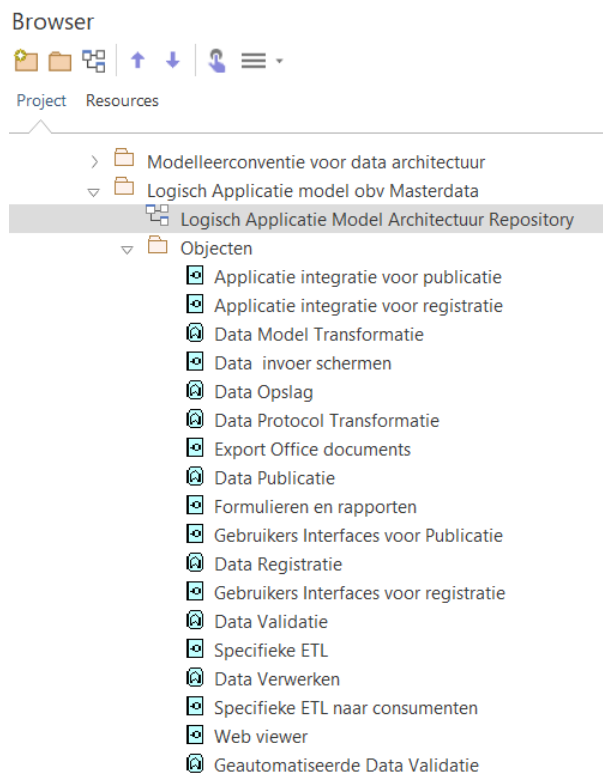
Bij het werken met een modelleer repository dien je bij het indelen van de repository inhoud rekening te houden met een aantal doelgroepen. De belangrijkste doelgroepen zijn

- **Modelleur**, dit zijn de verschillende rollen die modellen maken in de repository door diagrammen te maken en op deze diagrammen elementen en documenten te plaatsen. Vervolgens voegen ze detailcontent toe zoals relaties, attributen, beschrijvingen en andere detailconcepten. Deze doelgroep zal veelvuldig gebruik maken van de browser voor navigatie door de repository inhoud.

- **Reviewer**, deze doelgroep bekijkt de inhoud van de repository, beoordeelt deze inhoud en doet desgewenst suggesties voor verbeteringen in de modellen. Zij navigeren met name door de content, bekijken en beoordelen de voor hen relevante diagrammen en geven desgewenst commentaar op de aanwezige modellen. Zij kunnen de browser gebruik voor navigatie maar kunnen ook gebruik maken van andere navigatievormen zoals navigatie diagrammen, zoekmogelijkheden of de functionaliteiten in het Focus paneel.
- **Raadpleger**, richt zich met name op het raadplegen van de modellen zoals aanwezig in de repository. Navigeren met behulp van navigatie diagrammen, zoekmogelijkheden en model views in het Focus paneel.

Op basis van deze doelgroepen beschrijving zie je dat de modelleers de belangrijkste stakeholder zijn voor de repository indeling. Zij maken het meest gebruik van deze indeling voor navigatie door de modellen. De andere doelgroepen maken meestal gebruik van andere navigatievormen of zelfs van andere ontsluitingsvormen zoals bijvoorbeeld webbased content publicatie platformen. Vandaar dat wij in dit document de doelgroep modelleers als de belangrijkste persona beschouwen.

Drieëenheid package, diagram en element



In de browser komen in de basis slechts drie concepten voor namelijk packages, diagrammen en elementen. Wel kunnen diagrammen en elementen eigen typen en stereotypen zijn, zoals een bijzonder element het document. Maar ook dat is feitelijk een element. De packages kenmerken zich dat de inhoud van een package ook weer packages kunnen zijn naast de diagrammen en de elementen. De diagrammen zijn daarbij bijzonder omdat zij feitelijk een visuele representatie omvatten van elementen die in dit package of uit andere packages komen.

In de afbeelding zie je deze indeling terug met packages, sub packages, diagrammen en elementen.

Werk je een model uit in de repository dan begin je veelal met het aanmaken van een package. Vervolgens ga je in dat package één of meerdere diagrammen maken en op dit diagram plaats je dan elementen die je opzoekt in de repository en op het diagram plaats. Ook maak je nieuwe elementen aan in de repository door deze via de toolbox op een diagram te plaatsen. Kenmerkend is dat Sparx Enterprise Architect elementen die je nieuw aanmaakt, via de toolbox, in dezelfde package plaats als waar het diagram staat.

Hierbij komt een fundamenteel discussiepunt naar boven zet je elementen en diagrammen gezamenlijk in één package of worden de elementen en diagrammen in eigen packages geplaatst, zoals het voorbeeld in de afbeelding.

Hoe meer je repository gebaseerd gaat modelleren des te meer reden is er om de elementen en de diagrammen van elkaar te scheiden en deze in eigen packages te plaatsen. Redenen zijn:

- Elementen wil je hergebruiken in meerdere diagrammen. De diagrammen zijn dan verspreid over en opgenomen in andere packages. Het is dan onlogisch waarom het element dan bij één diagram staat.
- De repository heeft ook een registerfunctie voor elementen. Daarom wil je elementen groeperen naar type en in een eigen package plaatsen.
- Diagrammen zijn een visuele representatie en vragen daarom veelal om een eigen navigatiestructuur, bijvoorbeeld op basis van een packagestructuur. De registraanpak van element vraagt om een andere navigatiestructuur op basis van packages.

Domein voor indelen

Voor het indelen van de repository content kunnen afhankelijk van de organisatie meerdere domeinen relevant zijn. Dit is van een aantal factoren afhankelijk. Denk hierbij aan:

- Structuur van de organisatie en domein indeling relevant voor de organisatie.
- Modelleerdomeinen in gebruik binnen de organisatie zoals architectuur en business analyse
- Gebruikte modelleertalen binnen de organisatie

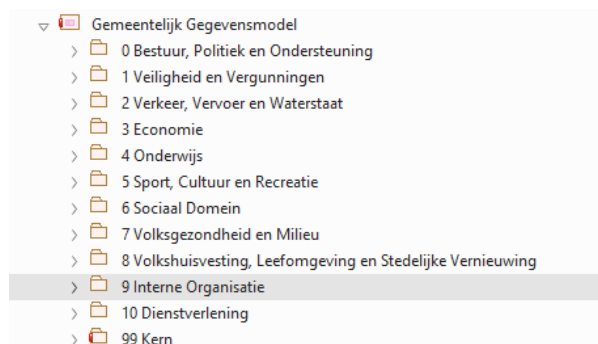
Daarnaast zijn er andere domeinen mogelijk afhankelijk van de context van de organisatie. Daarnaast kunnen hybride domeinindelingen ontstaan zeker in meer omvangrijke organisaties of een complexere context.

Organisatie domeinen

Veel organisaties kennen een domeinindeling van de inrichting van de bedrijfsprocessen en het organogram van de organisatie. Een en ander is afhankelijk van de omvang van de organisatie maar ook van de sector waar de organisatie in acteert.

Denk hierbij aan bijvoorbeeld een domein indeling voor de energiesector:

- Asset Management
- Service en Operations
- Klanten en markten



Dit is een hele gangbare eerste indeling voor de modellen die uitgewerkt worden. Vervolgens kan een detaillering gemaakt worden voor architectuur en of bedrijfsprocessen etc. De eerste laag van de packages in de browser (ook wel modellen genoemd) kunnen dan de namen krijgen van deze domeinen.

In de afbeelding van een package indeling zie je een voorbeeld van een dergelijke aanpak met packages voor de domeinen. In dit voorbeeld de domeinen voor een gemeente.

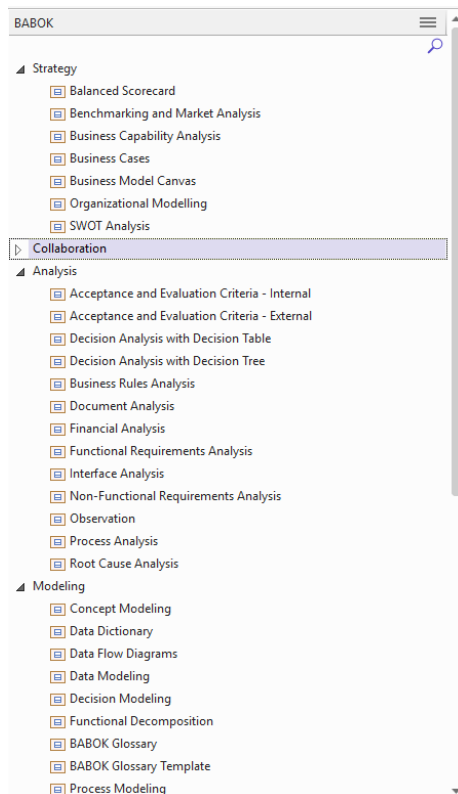
Net als in de afbeelding wordt deze package indeling gecombineerd met een andere domein indeling als boven- of onderliggende laag van de repository inhoud.

Modelleerdomeinen

Een logische indeling van de repository inhoud is het volgen van de modelleerdomeinen die uitgewerkt worden. Denk bijvoorbeeld aan

- Architectuur
- Data
- Software

- Bedrijfsprocessen
- Business analyse



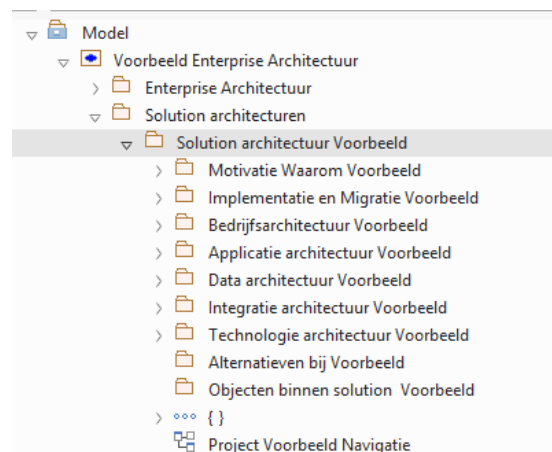
Ieder modelleerdomein kent vaak een eigen domeinindeling. Denk bijvoorbeeld bij architectuur aan de bedrijfslaag, applicatie laag en technische laag. Bij data wordt vaak een indeling gebruikt met conceptuele, logische en fysieke datamodellering gebruikt. Ook voor bedrijfsprocessen en business analyse kan een dergelijke indeling goed ingezet worden. Zie bijvoorbeeld de package en diagramindeling die gebruikt wordt bij de Business Analysis Body of Knowledge in de afbeelding.

Vaak zie je dat voor deze werkwijze een nadere detaillering gebruikt wordt op basis van de gebruikte modelleertalen. Daarnaast is een combinatie met de organisatie domeinindeling een veel gebruikte domeinindeling.

Modelleertalen

De indeling van de repository indelen op basis van de gebruikte modelleertalen is een veelgebruikte en logische indelingswijze. De metamodellen van modelleertaal ArchiMate kent een structurering op basis van lagen en aspecten voor de concepten. Het is dan natuurlijk een hele logische indeling om deze indeling te volgen.

Zie bijvoorbeeld de package indeling voor een referentie architectuur in de afbeelding die de lagen en aspecten van ArchiMate volgt. Deze gelaagde indeling zie je veelal terug in de indeling van de diagrammen maar ook van de elementen. Daarmee wordt een repository op basis van de modelleertaal vanzelf logisch ingedeeld.



Ook BPMN, UML en datamodelleren kennen een gelaagde indeling die als logisch geldt voor de indeling van de repository inhoud. Worden er binnen een repository meerdere domeinen gemodelleerd op basis van meerdere modelleertalen dan is het logisch om verschillende delen van de repository anders in te delen dan anderen. De indeling volgt dan de indeling die de modelleertaal voorstelt. Zie bijvoorbeeld de indeling van het enterprise architectuur en het data architectuur domein en de verschillende indelingen.

Hybride domeinindelingen

Zeker bij grotere organisaties zal kiezen voor één indelingsvorm onvoldoende blijken. Op dat moment wordt een hybride domeinindeling een logische keuze. Vaak zie je combinaties van indelingen zoals

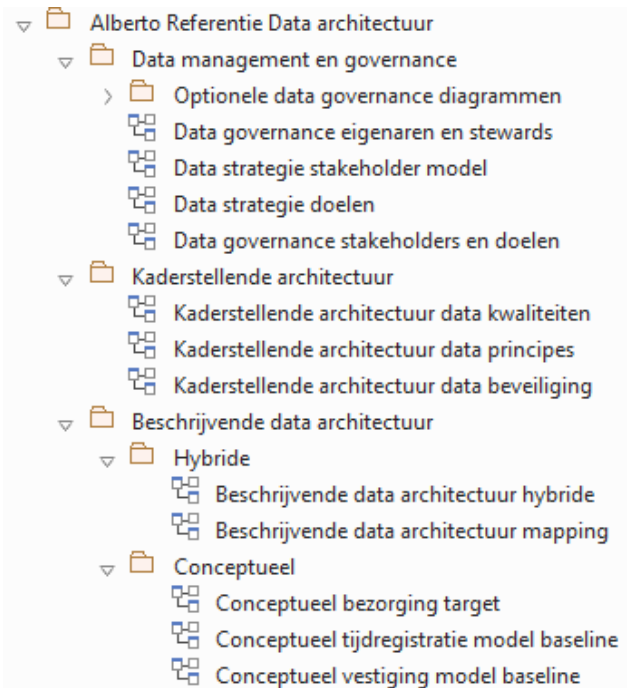
- Organisatie domein
- Modelleertaal

Of

- Modelleerdomein
- Organisatiedomein

Of

- Modelleerdomein
- Modelleertaal

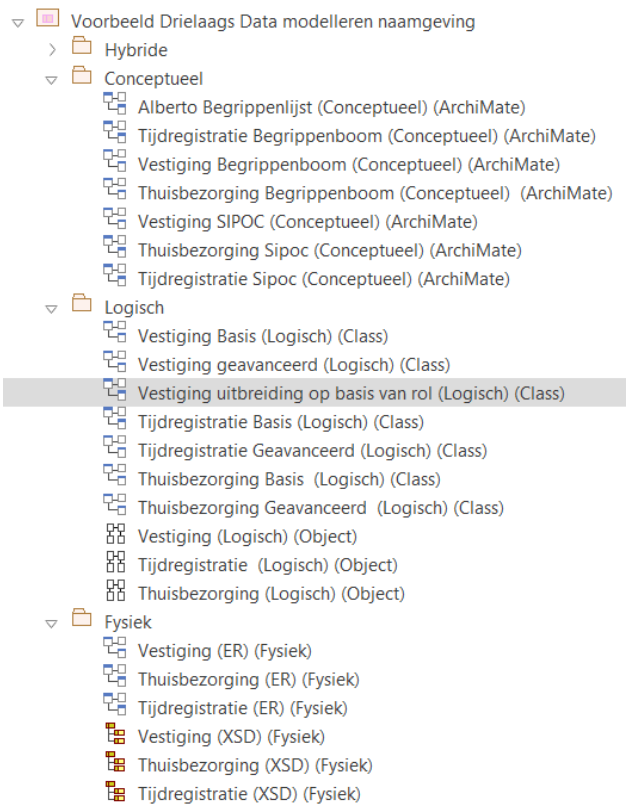


Welke indeling gekozen wordt is afhankelijk van de structuur van de organisatie, de modelleerdomeinen en -talen. Op basis hiervan kun je stellen dat dit echt afhankelijk is van de organisatie specifieke context. Mijn advies hierbij is meestal doe hier een aantal experimenten mee en betrek de verschillende modelleer stakeholders bij deze experimenten.

Naamgevingsconventies en domeinen

Voor het indelen van de repository inhoud is het werken met een boomstructuur opgebouwd uit packages en sub packages. In een aantal gevallen zie je dat die zeer fijnmazig wordt toegepast met als gevolg zeer brede maar ook zeer diepe boomstructuren.

Er is ook een andere manier om verschillende domeinen te introduceren, niet op basis van boomstructuren maar op basis van naamgevingsconventies voor packages en diagrammen. Door een codering te gebruiken bijvoorbeeld voor de organisatie domeinen maar ook voor modelleertalen of modelleerdomeinen kun je diagrammen makkelijk van elkaar onderscheiden op basis van coderingen in de namen van deze diagrammen. Hierdoor is het mogelijk om meer diagrammen in één package te plaatsen en deze te sorteren op basis van de codering die gebruikt wordt.



Zie de afbeelding waar dit wordt toegepast voor een data model domein waarin niet alleen de modelleertalen worden benoemd in de namen van een diagram maar ook de modelleerlaag. Met namen bij zoeken en groeperen in bijvoorbeeld het browser paneel met de context kun je handig groeperen

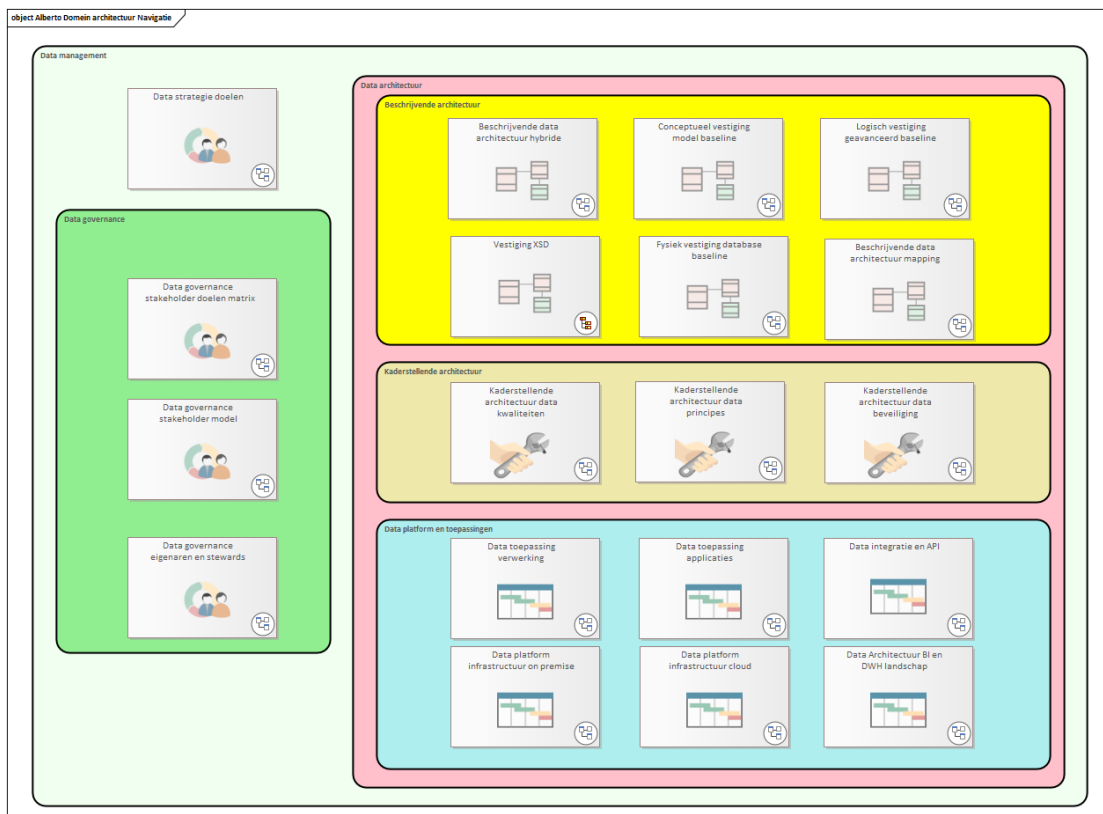
Alternatieven voor packages als indeling

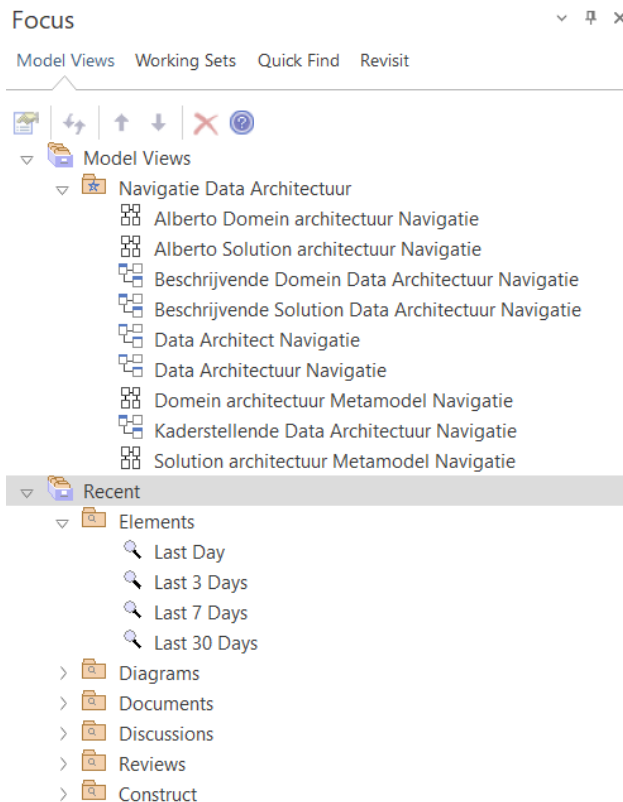
Werken met een package structuur voor het indelen van de repository is de meest gebruikte werkwijze om te structureren. Met name ter ondersteuning van de modelleers in de repository. Die zijn in deze de belangrijkste stakeholders.

Echter voor de andere stakeholders zoals reviewers en raadplegers van modellen kan de repository indeling minder bij hun behoefte aansluiten. Echter in Sparx Enterprise Architect of de webviewers zoals het Web Publicatie Platform (WPP) zijn er voldoende functionaliteiten aanwezig die de reviewers en raadplegers beter ondersteunen dan het browser paneel. Hier worden een aantal andere vormen van repository indeling en toegang kort toegelicht

Navigatie

In Sparx Enterprise Architect is er de mogelijkheid om navigatie elementen te gebruiken. Denk hierbij aan hyperlinks die linken naar diagrammen, packages of elementen aanwezig in de repository. Nog interessanter is de navigatiecel. Een algemeen element in de repository waarmee je navigatie biedt die los staat van de repository indeling. Je kunt hiermee met name voor raadplegers van de content die weinig gebruik maken van Sparx Enterprise Architect een handige navigatie bieden. Zie hieronder een afbeelding van een diagram dat is opgebouwd uit navigatiecellen waarmee je de raadplegers een mooi startpunt biedt voor navigatie door de repository inhoud.





Focus paneel

Het focus paneel is zoals de naam al zegt gericht op het bieden van inzicht op de repository inhoud vanuit een te definiëren perspectief. Naast dat je toegang krijgt tot bijvoorbeeld de laatst gewijzigde repository content kun je ook de laatst geraadpleegde inhoud snel benaderen.

Echter met name het zelf definiëren van modelviews met diagrammen over een bepaald onderwerp, een diapresentatie mogelijkheid als views zijn daarbij echt ondersteunend voor de reviewers en raadgelegers.

Werkt een raadgeleger of reviewer bijvoorbeeld vanuit een project scope in bepaalde diagrammen dan zijn ook working sets te definiëren. Hiermee maak je een collectie van diagrammen die

vanwege een bepaald onderwerp (zoals een project) bij elkaar horen en met één druk op de knop allemaal geopend worden.

Het quick search tabblad in het focus paneel maakt het mogelijk om te zoeken op trefwoord. Heel handig dat je hier zowel op elementen als op diagrammen kunt zoeken.

Zoeken

Binnen het focus paneel heb je de beschikking over de Quick Search functie. Daarnaast is er een zoekmogelijkheid in de ribbon om te zoeken naar packages. Echter ook is er een paneel voor zoeken beschikbaar. Hierin zijn een groot aantal zoekopdrachten voor gedefinieerd om op allerlei kenmerken te zoeken naar repository elementen, diagrammen en packages.

Wil je daarnaast zelf opdrachten definiëren om te zoeken in de repository dan kan dat. Je dient enige kennis te hebben van de interne structuur van de achterliggende database. Maar maak je hier gebruik van dan is dat een zeer krachtige manier om reviewers en raadgelegers te ondersteunen met het vinden van de voor hen relevante content.

Zo is er bijvoorbeeld in de open source IDEA addon een ArchiMate helper geïntroduceerd die je laat zoeken naar ArchiMate elementen op basis van de laag, aspect of het stereotype van ArchiMate elementen in de repository.

6 antipatronen voor repository indeling

Voor het indelen van de repository wordt veelal gezocht naar herkenbaarheid in de indeling. Die herkenbaarheid is zeker belangrijk. Daarnaast zie je dat de functionaliteiten van Sparx Enterprise Architect ook een bepaalde herkenbare indeling met zich meebrengen. Echter herbruikbaarheid en herkenbaarheid kunnen bij het indelen van de repository inhoud ook bijdragen aan dat gebruikers in de repository de weg kwijtraken of elementen niet eenvoudig kunnen terugvinden. In bovenstaande paragrafen zijn we ingegaan op hoe de inhoud te structureren. Hieronder beschrijven we een aantal indelingswijzen die je beter kunt vermijden.

Gestandaardiseerde package indeling

Vaak wordt geprobeerd om een gestructureerde package indeling te gebruiken. Binnen één modelleerdomein is dit zeker een goed idee. Echter probeer geen gestandaardiseerde package indeling te introduceren die voor meerdere modelleerdomeinen geldt. Ieder modelleerdomein heeft behoefte aan een eigen package indeling.

Lege packages

Heb je een herbruikbare package indeling gemaakt die je aanbiedt als een template voor deelmodellen in je repository die modelleers kunnen kopiëren en vullen met hun eigen modellen dan introduceer je standaardisatie. Dat is positief. Echter afhankelijk van dit template zie je vaak dat er na het uitwerken van het model er een groot aantal packages leeg blijft. Er staan dus geen diagrammen en elementen in deze packages. Dit maakt veel navigatie overbodig. Voorkom dus dat er veel packages zijn in je repository waar niets in staat. Maak hier werkafspraken over in je modelleerconventies.

Diepe gelaagdheid

Je kunt bij je package indeling een diepe gelaagdheid introduceren. Echter een diepe gelaagdheid biedt weinig overzicht aan de gebruikers. Wil je de repository indeling fijnmazig detailleren dan werkt een brede gelaagdheid beter. Daarmee behoudt een gebruiker eenvoudiger het overzicht.

Diagrammen en elementen in één package

Standaard zet Sparx Enterprise Architect nieuwe elementen die je aanmaakt op een diagram in dezelfde package als waar het diagram staat. Hiermee zie je dat er met name bij elementen die veel hergebruikt worden, en dus op meerdere diagrammen voorkomen, voor verwarring. Daarom is het handiger om elementen en diagrammen in verschillende packages te plaatsen en dus te scheiden. De repository is voor elementen een register en voor diagrammen een modelleeromgeving. Zorg dat je modelvormen niet met elkaar vermengt. Met name bij veel hergebruik en standaardisatie werkt dit contraproductief.

Nesting

Standaard gedrag van Sparx Enterprise Architect is dat als je een element in een diagram binnen de contouren van een ander element plaatst deze in het package worden genest. Staan de elementen niet in hetzelfde package dan past Sparx dit nesten niet toe. Werk je

in één diagram dan kan dit nesten handig zijn. In alle andere gevallen is dit onhandig en werkt het niet consistent en schept dit verwarring voor de gebruikers. Mijn advies is gebruik dit nesten niet en zet het nesten in de instellingen ook gewoon uit.

Packages indelen op basis van nummers

Vaak zie ik dat packages genummerd worden om een bepaalde volgorde af te dwingen. Dat is niet nodig, je kunt eenvoudig weg de elementen sorteren in een eigen volgorde in het browser paneel. Die nummers van packages wijzigen regelmatig, moet je steeds de nummering aanpassen. Soms verdwijnen packages uit de volgorde en dan klopt de nummering niet meer. Allemaal redenen om nummering niet te gebruiken als het niet nodig is.

Over de auteur



Bert Dingemans is trainer op het vlak van data architectuur, datamanagement en Big Data. Hij heeft een passie voor modelleren, de modelleertool Sparx Enterprise Architect en het effectief inzetten van geautomatiseerde hulpmiddelen om modellen effectief in te zetten in de praktijk. Bert is te bereiken via bert@interactory.nl