



Master Thesis

Onderzoeksrapport

Architectuurprincipes: functie en formulering

Begeleider:	E. Proper
Auteur:	K.A.J. van Boekel S0552054
Opleiding:	Radboud Universiteit Nijmegen
Faculteit:	Natuurwetenschappen, Wiskunde en Informatica
Studierichting:	Informatiekunde
Plaats en datum:	Mill, februari 2009
Versie:	v1.0
Scriptienummer:	90 IK

Abstract

Deze scriptie is het resultaat van het onderzoek dat is uitgevoerd met als doel om meer inzicht te verkrijgen in de functie, eigenschappen, beschrijving en het gebruik van architectuurprincipes. De aanleiding hiervoor is het volgende:

Organisaties gebruiken verschillende instrumenten die er voor moeten zorgen dat de strategie en visie verwezenlijkt wordt. Denk hierbij aan doelen, doelstellingen, strategische uitgangspunten, eisen en wensen die de strategie en visie beschrijven en aan regels, richtlijnen en voorschriften die er voor moeten zorgen dat de strategie uitgevoerd wordt en de visie werkelijkheid wordt. De overeenkomst tussen al deze instrumenten is dat ze allemaal richting geven aan het handelen van de organisatie. Je ontwerpt er als het ware de organisatie mee. Het verschil tussen deze instrumenten is echter lastig te doorgronden. Ze worden vaak door elkaar heen gebruikt en iedereen kent er vaak net een andere functie en betekenis aan toe. Wat de één opschrijft als een uitgangspunt kan de ander goed als eis of wens opschrijven. Je zou kunnen denken; wat maakt het nu uit hoe je een uitspraak noemt, het gaat om de inhoud van de uitspraak en als het maar resultaat oplevert. Dit laatste is echter juist iets waar het vaak mis gaat. Het resultaat wordt niet bereikt of is onbekend, waardoor het nut van het volgen of ambiëren van zo'n uitspraak ter discussie wordt gesteld en mogelijk genegeerd wordt. Daarnaast dienen alle instrumenten samen voor resultaat te zorgen en is ieder instrument dus een schakel in het proces met een unieke noodzakelijke functie. Een zwakke schakel brengt het hele proces in gevaar en daarom is de functie en betekenis van de diverse instrumenten wel degelijk van belang. In deze scriptie wordt onderzoek gedaan naar de functie en betekenis van deze instrumenten en de onderlinge relaties die ze met elkaar hebben.

Een onbekendere telg in deze verzameling van richtinggevende uitspraken die bezig is met een opmars, is het architectuurprincipe. Dit instrument, afkomstig uit het vakgebied Enterprise Architectuur, is een grote belofte die er voor moet zorgen dat organisaties meer toekomstvast en betrouwbaar 'ontworpen' worden. Over de functie, definitie en juiste wijze van formuleren en gebruik zijn nog veel discussies waardoor voorlopig deze belofte nog niet is ingelost. Een paar voorbeelden van architectuurprincipes:

"Elk gegeven heeft één eigenaar."

"Onze organisatie is zó ingericht dat wij onze klanten klantvriendelijk kunnen bedienen."

"Ontwerp van processen, applicaties, services en infrastructuur gebeurt service georiënteerd."

Deze uitspraken zouden door veel mensen echter zo voor een regel, doel of wens kunnen worden aangezien. Op deze wijze kan het architectuurprincipe zich niet onderscheiden van de rest van de richtinggevende uitspraken en lijkt het weinig meer waarde te hebben.

De volgende onderzoeksvraag is geformuleerd naar aanleiding van de probleemstelling:

Welke eigenschappen en functie heeft een architectuurprincipe en hoe dient een architectuurprincipe geformuleerd te worden?

Om een beter beeld te krijgen van deze uitspraken die in de praktijk architectuurprincipes genoemd worden, is een raamwerk opgesteld waarin deze uitspraken gepositioneerd kunnen worden op basis van een aantal eigenschappen. Dit raamwerk, genaamd Proposition Framework, is in dit onderzoek gebruikt om de architectuurprincipes van Stater NV (een onafhankelijke leverancier van diensten aan hypotheekverstrekkers) te onderzoeken en te positioneren in het raamwerk. Het raamwerk is opgezet op basis van theorie en praktijkervaring van de auteurs en is door middel van dit onderzoek voor de eerste keer gebruikt en getoetst.

De bevindingen op basis van het uitvoeren van dit experiment zijn:

De architectuurprincipes van Stater NV zijn samen met de domeinexperts van Stater NV geclassificeerd. De architectuurprincipes zijn leidend geweest bij de bouw van een nieuwe ICT-oplossing. Op basis van de classificatie is vastgesteld dat deze set van architectuurprincipes vooral een gewenste situatie beschrijven. De architectuurprincipes beschrijven de functie en constructie van de te bouwen ICT-oplossing. De architectuurprincipes zijn op logisch of conceptueel niveau beschreven in natuurlijke taal en zijn richtinggevend. De architectuurprincipes hebben vooral betrekking op de ICT-oplossing zelf en weinig op de context en omgeving waarin de ICT-oplossing zal gaan worden ingezet. De impact, motivatie, issues en conflicten zijn niet opgenomen in de beschrijvingen van de architectuurprincipes.

Het Proposition Framework v0.9 is in de huidige vorm een handig overzicht van mogelijke eigenschappen die architectuurprincipes kunnen hebben. Het kan als checklist dienen bij het opstellen van architectuurprincipes en hulp bieden bij het bepalen van de functie van een architectuurprincipe. Hiermee wordt bedoeld dat een organisatie kan bepalen hoe men het architectuurprincipe als instrument wil gaan inzetten en welke eigenschappen het dan moet hebben.

Meer inzicht verkrijgen in de eigenschappen van de verschillende propositie soorten wordt in mijn ogen met de huidige versie van het raamwerk en het uitgevoerde experiment nog niet bereikt. Dit komt omdat het raamwerk nog onvoldoende is uitgekristalliseerd en van te voren niet beoordeeld is of de architectuurprincipes uit de casussen wel van voldoende kwaliteit zijn en dus als goed voorbeeld gezien mogen worden.

Op basis van de bevindingen zijn aanbevelingen gedaan aan Stater NV hoe ze in het vervolg hun architectuurprincipes anders kunnen formuleren zodat ze zorgen voor meer resultaat.

Om meer inzicht te verkrijgen in de problematiek die wordt veroorzaakt door het verkeerd gebruiken en formuleren van architectuurprincipes is een probleemanalyse uitgevoerd op het architectuurproces en de architectuurprincipes van Stater NV. Tevens zijn een aantal architectuurprincipes geherformuleerd door middel van een gestructureerd stappenplan voor het opstellen en formuleren van architectuurprincipes (Dragon1) om te achterhalen wat er aan de huidige architectuurprincipes ontbreekt. Een greep uit de misstanden die zijn geconstateerd op basis van dit onderzoek:

- De motivatie en impact van het hanteren van de architectuurprincipes is niet expliciet gemaakt en daardoor onvoldoende bekend bij de business van Stater NV. Hierdoor zijn architectuurprincipes overboord gegooid bij tijd- en geldgebrek en heeft de business van Stater NV van te voren geen oordeel kunnen vellen over de noodzaak van het hanteren van deze architectuurprincipes. Hierdoor heeft de business van Stater NV weinig op met architectuur en vormen de architectuurprincipes een zwakke schakel in het ontwerpproces van de onderneming.
- De architectuurprincipes lijken een verzameling te vormen van ontwerpregels en bouwrichtlijnen waar het label architectuurprincipes is opgeplakt. Dit leidt tot onduidelijkheid over de toegevoegde waarde van het gebruik van architectuurprincipes.

Het herformuleren van de architectuurprincipes heeft tot het inzicht geleid dat het gebruik van een gestructureerd stappenplan voor het opstellen en formuleren van architectuurprincipes een deel van de bevonden problematiek kan afvangen. Vervolgens is door middel van een literatuurstudie en gesprekken met de bedenker van Dragon1 meer inzicht verkregen in de functie, eigenschappen, beschrijving en het gebruik van architectuurprincipes. Op basis van de bevindingen tot dan toe is besloten om het stappenplan van Dragon1 verder uit te werken, met als doel de ondervonden problematiek beter te kunnen afvangen en het stappenplan scherper te formuleren. De voorgestelde wijzigingen en toevoegingen zijn door de bedenker van het stappenplan goedgekeurd en zullen worden verwerkt in het stappenplan.

Op basis van dit onderzoek is het volgende antwoord te geven op de onderzoeksvraag:

De functie van het architectuurprincipe is het vertalen van de strategie naar een ontwerp voor een oplossing die deze strategie ten uitvoer brengt. Een architectuurprincipe zorgt er voor dat strategische uitgangspunten, doelen, eisen en wensen op de juiste wijze worden doorvertaald naar (ontwerp)regels en (bouw)richtlijnen. Een architectuurprincipe geeft dus richting aan het ontwerp van de organisatie.

Een architectuurprincipe beschrijft hoe iets werkt of hoe iets is én wat dit voor resultaat heeft en hoe dit resultaat bijdraagt aan het verwezenlijken van de strategische uitgangspunten, doelen, eisen en wensen. Het architectuurprincipe beschrijft een bewezen, duurzame, betrouwbare en toekomstvaste aanpak waarvan het resultaat gegarandeerd kan worden binnen een bepaalde context. De beschrijving dient begrijpbaar te zijn voor de opdrachtgever en opdrachtnemer. Een architectuurprincipe beschrijft een (deel van een) concept dat kan worden toegepast in een context.

Hoe dient een architect op de juiste manier architectuurprincipes te formuleren en te gebruiken:

- Stel in samenwerking met de opdrachtgever de strategische uitgangspunten (doelen, eisen en wensen) vast indien dit nog niet gedaan is.
- Bepaal welke (bewezen) concepten kunnen worden gebruikt in een oplossing om deze strategische uitgangspunten te verwezenlijken.
- Beschrijf met de architectuurprincipes hoe deze concepten werken en hoe deze bijdragen aan het verwezenlijken van de strategische uitgangspunten, doelen, eisen en wensen.
- Geef invulling aan de attributen van de architectuurprincipes die beschreven dienen te worden zoals handhavingmechanisme, implementatie impact, validiteit, bedrijfsvoordelen- en nadelen, kosten / baten zodat de opdrachtgever een objectief en realistisch beeld heeft van waar men voor kiest.
- Verkrijg goedkeuring van de opdrachtgever over de geformuleerde architectuurprincipes.
- Vertaal de architectuurprincipes door naar (ontwerp)regels en (bouw)richtlijnen.
- Zoek een geschikte partij die deze oplossing kan realiseren.

Door het architectuurprincipe deze functie en beschrijving mee te geven is het architectuurprincipe van toegevoegde waarde t.o.v. van de al bestaande richtinggevende uitspraken. Sterker nog: het vormt de verbindende schakel tussen deze richtinggevende uitspraken en is daardoor een krachtig hulpmiddel om een toekomstvaste en betrouwbare organisatie of ICT-oplossing te realiseren.

Voorwoord

Dit onderzoeksrapport bevat de resultaten van het onderzoek naar architectuurprincipes uitgevoerd in het kader van de Master Thesis door Koen van Boekel. Het onderzoek vormt de afsluiting van de Master Informatiekunde aan de Radboud Universiteit Nijmegen.

Het onderzoek wordt uitgevoerd binnen het Nijmeegs Instituut voor Informatica en Informatiekunde (NIII), dat onderdeel is van de faculteit der Natuurwetenschappen, Wiskunde en Informatica (FNWI). Dhr. prof.dr. H.A. Proper zal optreden als begeleider in dit onderzoek.

De volgende personen wil ik graag bedanken voor hun medewerking: Erik Proper, Wim van Stokkum, Gerrit uit de Bosch, Mark Paauwe en de leden van de Joint NAF / ArchiMate “Business Principles” Workgroup.

Door het exploratieve karakter van dit onderzoek en de vele paden die zijn bewandeld is veel kennis opgedaan en een visie en mening gevormd over het vakgebied Enterprise Architectuur en met name architectuurprincipes. In de diverse hoofdstukken is getracht deze visie te verwerken.

Koen van Boekel

Inhoudsopgave

1. Inleiding	7
2. Onderzoekopzet	8
2.1.1 Onderzoeksvraag.....	9
2.1.2 Deelvragen	9
2.1.3 Precisie	10
2.1.4 Functionaliteit	10
2.2 Verantwoording en relevantie	10
2.3 Theoretisch kader	12
2.3.1 Verankering.....	12
2.3.2 Begrippenkader.....	12
2.3.3 Keuzes en vooronderstellingen.....	12
2.3.4 Conceptueel model	13
2.4 Methode / Aanpak.....	15
2.5 Structuur onderzoeksrapport.....	17
3. Proposition Framework	18
4. Visie op Enterprise Architectuur	20
4.1 Wat is Enterprise Architectuur?	20
4.2 Wat zijn architectuurprincipes?	21
5. Casus	23
5.1 Inleiding casus.....	23
5.2 Architectuurproces Stater NV	23
5.3 Overzicht architectuurprincipes	25
6. Classificatie architectuurprincipes Stater NV	27
6.1 Werkwijze	27
6.2 Resultaat classificatie	27
6.3 Bevindingen	29
6.4 Conclusie	30
7. Verhouding EA methoden en Proposition Framework v0.9	32
7.1 Experiment.....	32
7.2 Bevindingen	36
7.3 Conclusie	37
8. Evaluatie Proposition Framework	39
8.1 Algemene bevindingen	39
8.2 Specifieke bevindingen.....	40
8.3 Conclusies & Aanbevelingen	41
9. Aanbevelingen voor Stater NV	44

10. Analyse architectuurproces Stater NV	47
10.1 <i>Knelpunten en problematiek architectuurproces Stater NV</i>	47
10.2 <i>Herformuleren architectuurprincipes Stater NV</i>	48
11 Functie, eigenschappen en beschrijving van het architectuurprincipe	50
11.1 <i>Functie en definitie volgens Dragon1.....</i>	50
11.2 <i>Gevolg, resultaten, mogelijkheden: synoniem?</i>	51
11.3 <i>Bouwstenen en structuur (deel 1).....</i>	52
11.4 <i>Bouwstenen en structuur (deel 2).....</i>	53
11.5 <i>Soorten architectuurprincipes.....</i>	54
11.6 <i>Architectuurprincipes: valide?</i>	56
11.7 <i>Functie van een architectuurprincipe.....</i>	57
11.8 <i>Structuur van de beschrijving van een architectuurprincipe.....</i>	58
11.9 <i>Redeneerstrategie en architectuurprincipes.....</i>	59
11.10 <i>Het werken met bewezen architectuurprincipes</i>	59
11.11 <i>Het toepassen van een architectuurprincipe</i>	60
11.12 <i>Conclusie</i>	61
12 Een stappenplan voor het formuleren van architectuurprincipes.....	62
12.1 <i>Inhoud van de attributen</i>	62
12.2 <i>De relaties en volgorde van de attributen</i>	68
12.3 <i>Resultaat analyse attributen</i>	72
12.4 <i>Kandidaat attributen</i>	74
12.4.1 <i>Alternatieve representatiewijze.....</i>	74
12.4.2 <i>Principeoplossingen / aanpakken</i>	74
12.4.3 <i>Validiteit</i>	75
12.5 <i>Overzicht attributen en volgorde inclusief kandidaat attributen.....</i>	76
12.6 <i>De context van het architectuurprincipe.....</i>	77
12.7 <i>Relatie attributen met het short statement.....</i>	78
12.8 <i>Attribuut functiesoorten</i>	79
12.9 <i>Overzicht uiteindelijke attributen</i>	80
12.10 <i>Conclusie</i>	85
13 Conclusies.....	87
13.1 <i>Beantwoording deelvragen.....</i>	87
13.2 <i>Beantwoording onderzoeksvraag</i>	90
13.3 <i>Belangrijkste bevindingen.....</i>	91
13.4 <i>Vervolgonderzoek</i>	92
13.5 <i>Reflectie</i>	92
Literatuur	93
Bijlage A: Proposition Framework v0.9.....	95
Bijlage B: Uitwerking classificatie zonder commentaar en motivatie	104
Bijlage C: herformuleren architectuurprincipes Inleiding.....	127
Bijlage D: Attributen van een architectuurprincipe Dragon1	147

1. Inleiding

Binnen organisaties worden verschillende soorten richtinggevende uitspraken gehanteerd die er voor moeten zorgen dat de organisatie als een geoliede machine loopt en blijft lopen. Denk hierbij aan regels, richtlijnen, uitgangspunten, doelstellingen, eisen, wensen etc. De algemene functie van deze verzameling instrumenten is voor iedereen bekend, maar het verschil tussen deze instrumenten is voor velen moeilijk te onderscheiden. Dit onderzoek richt zich op een nieuwe telg binnen deze verzameling, die voor de nodige discussie zorgt: het architectuurprincipe. Het doel van een architectuurprincipe is om richting te geven aan een ontwerp. Dit ontwerp kan betrekking hebben op een organisatie, maar ook op een informatiesysteem. De wijze waarop en wat een architectuurprincipe nu moet beschrijven om zijn functie te kunnen uitvoeren is echter nog onderwerp van discussie. Een paar bekende voorbeelden geven een goed beeld van in welke smaken je het architectuurprincipe tegenkomt:

“Elk gegeven heeft één eigenaar.”

“Onze organisatie is zó ingericht dat wij onze klanten klantvriendelijk kunnen bedienen.”

“Ontwerp van processen, applicaties, services en infrastructuur gebeurt service georiënteerd.”

Om meer inzicht te verkrijgen in de verschillen tussen de soorten richtinggevende uitspraken is een raamwerk opgesteld waarmee de eigenschappen van deze instrumenten in kaart kunnen worden gebracht. Het raamwerk is in dit onderzoek gebruikt om de eigenschappen van architectuurprincipes die in de praktijk gehanteerd worden vast te leggen. Bij het opstellen van het raamwerk is rekening gehouden met de eerder besproken diversiteit en de verschillende soorten en smaken architectuurprincipes zouden moeten passen in het raamwerk.

De casus die gebruikt is voor dit onderzoek is afkomstig van Stater NV. Stater NV is een onafhankelijke leverancier van diensten aan hypotheekverstrekkers in verscheidene Europese landen, met een sterke positie in Nederland. De architectuurprincipes die gehanteerd zijn bij het ontwikkelen van hun nieuwe ICT-oplossing, zijn als onderzoeksobject gebruikt in dit onderzoek.

Het probleem is echter tweezijdig te noemen; enerzijds is er de vraag of het raamwerk correct is, anderzijds is er de vraag in hoeverre de architectuurprincipes van Stater NV als correct voorbeeld gezien kunnen worden. De nadruk zal echter liggen op de toetsing van het raamwerk. Het doel van dit onderzoek is te achterhalen of het raamwerk van toegevoegde waarde is bij de eerder geschetste zoektocht.

Dit onderzoeksrapport beschrijft de twee fases van het onderzoek. In fase 1 wordt de classificatie van architectuurprincipes door middel van het Proposition Framework v0.9 besproken. Fase 2 beschrijft een reeks van experimenten waarin meer inzicht verkregen is in het formuleren van architectuurprincipes.

2. Onderzoeksopzet

In dit hoofdstuk wordt een overzicht van het onderzoek weergegeven. Als eerste zal de probleemstelling worden behandeld en welke onderzoeksvragen op basis hier van zijn opgesteld. Vervolgens wordt ingegaan op de relevantie van het onderzoek en wordt het onderzoeksgebied nader besproken. Daarna zal stapsgewijs worden verteld welke aanpak en methode gehanteerd is om een antwoord te kunnen geven op de onderzoeksvragen. Als laatste wordt de structuur van dit onderzoeksrapport uiteengezet.

2.1 Probleemstelling

Met de onderstaande probleemstelling is begonnen aan het onderzoek. Deze probleemstelling vormt het startpunt. Tijdens het onderzoek zijn nieuwe deelproblemen aan het licht gekomen die de richting en resultaat van het onderzoek gestuurd en bepaald hebben:

Binnen het vakgebied Enterprise Architectuur zijn er grote verschillen in het gebruik van definities en begrippen die horen bij dit nog jonge vakgebied. Wanneer je kijkt naar de verschillende vergelijkbare definities van Enterprise Architectuur zul je verschillende 'dingen' tegenkomen die vallen onder de begrippen 'principles', 'business rules', 'rules', 'policy', 'guidelines' etc. Het verschil tussen deze begrippen is vervaagd en voor velen niet meer duidelijk te onderscheiden. Om een beter inzicht te verkrijgen in de verschillende vormen die deze 'dingen' hebben is een onderzoek gestart door het Joint NAF / ArchiMate "Business Principles" Workgroup, met als doel te achterhalen welke eigenschappen deze 'dingen' nu precies hebben. Het eerste resultaat hiervan is een raamwerk waarin deze 'dingen' gepositioneerd en geclassificeerd kunnen worden. De vraag is echter of dit raamwerk volledig en correct is en het daadwerkelijk mogelijk is om deze 'dingen' te positioneren en classificeren door middel van dit raamwerk. Deze 'dingen' worden door deze werkgroep gegroepeerd onder de verzamelnaam 'propositions'.

Het architectuurprincipe is een propositie soort waarover nog zeer veel gediscussieerd wordt. Welke eigenschappen en functie heeft deze propositie? Binnen het vakgebied Enterprise Architectuur (EA) zijn veel verschillende definities van principes en architectuurprincipes te vinden. Er zijn verschillende visies over welke eigenschappen een architectuurprincipe heeft en wanneer een architectuurprincipe een architectuurprincipe genoemd mag worden. Ook op de vraag hoe je het beste een architectuurprincipe kan formuleren en wanneer een architectuurprincipe nu goed of correct is, worden verschillende antwoorden gegeven door de diverse EA methoden.

De vraag hierbij is of het raamwerk kan helpen om hier een antwoord op te vinden. Vanuit dit oogpunt is het dan ook noodzakelijk om het raamwerk te toetsen aan de hand van real-world casussen. Indien dit een hulpmiddel blijkt te zijn kan het vaker worden toegepast en zal het helpen bij de eerder geschetste zoektocht.

Een punt van discussie dat veel stof doet op waaien in deze zoektocht is de mate waarin een architectuurprincipe een uitspraak moet zijn die altijd waar is (of waarvan men graag ziet dat deze waarheid wordt of zou moeten zijn). Hier zijn de meningen binnen het vakgebied EA nog over verdeeld. Dat dit echter wel gewenst is, zal door niemand ontkend kunnen worden (het zou de ideale situatie zijn, namelijk een beheersbare situatie). Daarnaast zijn er verschillende visies over wat een architectuurprincipe nu moet beschrijven. Moet het een werkwijze zijn, dus moet het iets zeggen over het hoe? Moet het alleen betrekking hebben op iets wat men wil of graag wenst? Door deze verschillende visies is de verschijsning waarin architectuurprincipes voorkomen zeer divers.

De verschillende methoden die er zijn op het gebied van EA geven hier geen eenduidig of duidelijk antwoord op. Een methode die hier wel veel aandacht aan besteed is Dragon1 [Dragon1/08a]. Dragon1 zegt dat een architectuurprincipe altijd waar (in ieder geval een waarschijnlijkheid van > 0.8) moet zijn en dus een bewezen aanpak zou moeten beschrijven. De vraag hierbij is hoe je architectuurprincipes kan formuleren die onderbouwd zijn en ook bewezen kan worden dat ze waar zijn. Een aanpak die hierbij uitkomst kan bieden is om bij het formuleren van architectuurprincipes gebruik te maken van de redeneerwijze van propositiologica en/of predicatenlogica. Deze vorm van correct redeneren zou houvast kunnen bieden.

In de bovenstaande probleemstelling wordt alvast de top van de ijsberg weergegeven en het bevat al een aantal inzichten die verder onderzocht zijn in dit onderzoek.

2.1.1 Onderzoeksvraag

Op basis van de probleemstelling is één (hoofd) onderzoeksvraag geformuleerd:

Welke eigenschappen en functie heeft een architectuurprincipe en hoe dient een architectuurprincipe geformuleerd te worden?

2.1.2 Deelvragen

Door de omvang en complexiteit van het vraagstuk zal de hoofdvraag niet compleet beantwoord kunnen worden door middel van deze afstudeerscriptie. Het doel is dan ook om een bijdrage te leveren aan een antwoord op deze vraag. De deelvragen en / of subvragen geven inzicht in de bijdrage op het vraagstuk die men kan verwachten in deze afstudeerscriptie:

Fase 1:

Welke eigenschappen hebben architectuurprincipes die in de praktijk gebruikt worden?

- Welke eigenschappen zijn toe te kennen aan een architectuurprincipe op basis van de classificatie (door middel van het Proposition Framework) van de architectuurprincipes van Stater NV?

Welke eigenschappen hebben architectuurprincipes volgens de theorie?

- Welke eigenschappen zijn toe te kennen aan een architectuurprincipe op basis van de theorie van architectuurmethoden- en raamwerken?

In hoeverre is het Proposition Framework correct en bruikbaar in de praktijk?

- Welke problemen zijn er bij het toepassen van het Proposition Framework op een praktijkcasus?
- Welke aanbevelingen zijn er op het Proposition Framework naar aanleiding van deze problemen?
- Hoe kan Stater NV het Proposition Framework gebruiken om betere architectuurprincipes te formuleren?

Fase 2:

Welke problemen ontstaan er in de praktijk door de wijze waarop architectuurprincipes gebruikt en geformuleerd worden?

- Welke problemen zijn te onderkennen binnen het architectuurproces van Stater NV en welke rol speelt het gebruik en de wijze van formuleren van architectuurprincipes hierin?

Kunnen deze problemen voorkomen worden door architectuurprincipes anders te formuleren?

- Zijn deze problemen te voorkomen door gebruik te maken van een gestructureerd stappenplan voor het opstellen van architectuurprincipes?

Zoals eerder aangegeven is het onmogelijk om alle antwoorden te geven op de verschillende vraagstukken. Deze afstudeerscriptie zal dan ook gedeeltelijk als doel hebben om een handvat te bieden en / of een voorbeeld aanpak te zijn om een antwoord te verkrijgen op het vraagstuk.

2.1.3 Precisie

Het domein van dit onderzoek is:

Architectuurprincipes en architectuurprocessen van bedrijven in Nederland.

2.1.4 Functionaliteit

De onderzoeksfunctie van dit onderzoek is beschrijvend, verklarend, toetsend en evaluerend.

2.2 Verantwoording en relevantie

Binnen het vakgebied EA is exploratief onderzoek gewenst. Het is een jong vakgebied waar de lijnen nog niet vast staan en men nog steeds op zoek is naar betere definities en kaders. Vooral op het gebied van architectuurprincipes zijn nog veel slagen te slaan.

De probleemstelling en onderzoeksvragen zijn relevant omdat er nog veel verschillende ideeën zijn over wat architectuurprincipes zijn en hoe deze geformuleerd zouden moeten worden en aan welke eisen een goed architectuurprincipe moet voldoen. Er is geen duidelijk overzicht van voorbeelden uit de praktijk die vallen onder de noemer architectuurprincipe en welke eigenschappen deze hebben. Daarnaast dient het raamwerk getoetst te worden in de praktijk om te controleren of deze valide is.

De onderstaande stukken uit diverse bronnen zijn geselecteerd om een beeld te geven van de relevantie van dit onderzoek. Het beschrijft de discussie over het formaliseren van architectuurprincipes. Dit is maar één van de velen gebieden waar onderzoek naar wordt gedaan op het gebied van architectuurprincipes. Het geeft echter een goed beeld weer van de discussie die alleen al gaande is over dit punt in de zoektocht naar de meest geschikte wijze van het formuleren van architectuurprincipes.

In de volgende artikelen is aandacht besteed aan het formaliseren van architectuurprincipes door middel van ORM en ORC. Aan de mogelijkheden voor het toepassen van gevolgtrekking en correct redeneren door middel van (propositie en predicaten)logica is echter nog weinig aandacht besteed.

- Formalizing Architecture Principles using Object-Role Modelling. – [CHO07]
- P. van Bommel, S.J.B.A. Hoppenbrouwers, H.A. Proper, and Th.P. van der Weide. Giving Meaning to Enterprise Architectures - Architecture Principles with ORM and ORC. – [BHPW06]

Mijn standpunt hierin is dat het formaliseren van architectuurprincipes twee doelen kan hebben:

- Het daadwerkelijk formeel beschrijven van een architectuurprincipe zodat het architectuurprincipe door iedereen op dezelfde manier geïnterpreteerd wordt.
- Het gebruiken van de redeneerstrategie / wijze van formuleren die gehanteerd wordt door formele talen om zo betere architectuurprincipes in natuurlijke taal te beschrijven.

Mijn inziens zijn beide doelen zeer interessant en hebben ze ook onderling weer diverse punten waar ze elkaar raken en elkaar versterken. Beide doelen hebben als hoofddoel de communiceerbaarheid en beschrijfbaarheid van een architectuurprincipe te optimaliseren. In dit onderzoek wordt op beide doelen ingegaan.

Serge Bouwens schrijft het volgende over eisen aan de taal voor het formuleren van architectuurprincipes [DYA08]:

Eisen aan taal voor architectuurprincipes

- *Zowel geschikt voor specificatie als communicatie.*
- *De taal moet om kunnen gaan met de vaagheid van de menselijke taal (niet dwingen tot formele specificatie). In de praktijk van requirements engineering bestaan tools waarbij eenduidig taalgebruik wordt ondersteund. Wellicht ook*

geschikt voor architecten?

Mijn visie hierop is dat één taal niet geschikt kan zijn voor specificatie en communicatie. Als een taal te veel detail en ingewikkelde constructies en notatiewijzen bevat is een taal niet meer gemakkelijk communiceerbaar.

De onderstaande conclusies zijn afkomstig uit een verslag van de resultaten van een workshop over EA [Gre2007]:

- *Vrijwel iedereen was het de tweede ronde eens met de noodzaak tot formaliseren van principes. Verklaring hiervoor was waarschijnlijk het inzicht van deelnemers dat goede definities van termen noodzakelijk is om tot overeenstemming te komen. Tijdens de workshop was er nogal wat discussie rondom terminologie. Dit is tevens een verklaring voor het feit dat niemand het de tweede stemronde meer eens was met stelling 8.*
- *Er waren de tweede stemronde toch ineens drie mensen die het gevoel hadden dat het kwantificeren van de motivatie achter een principe belangrijk is. Dit is waarschijnlijk te verklaren doordat er bij het opstellen van de principes discussie en onenigheid ontstond over het precieze nut.*

Hierboven wordt duidelijk gemaakt dat er verschillende mensen toch de noodzaak van het formaliseren van architectuurprincipes in zien. Wanneer je naar meerdere bronnen gaat kijken waar gelijke discussies worden gevoerd over het formaliseren van architectuurprincipes, kun je de conclusie trekken dat er zeer veel tegenstrijdigheden zijn binnen EA, alleen al op het gebied van architectuurprincipes en dan ook nog eens een deel hiervan. Dit laat maar eens te meer de complexiteit zien van het vakgebied en de problemen die hier binnen te vinden zijn. Architectuur moet communiceerbaar zijn en inzicht geven, duidelijkheid bieden en nog veel meer wordt er gezegd. Het is echter nog steeds onduidelijkheid over wat het nu inhoud en hoe je 'het' moet doen. Onderzoek naar wat EA is en wat het omvat is dus nog zeer gewenst en noodzakelijk.

Pieter Buitenhuis schrijft in zijn afstudeerscriptie [bui07] het volgende over voordelen van een formele modelleertaal:

“In een formele modelleertaal zijn de taalkundige eisen geformaliseerd. Dit houdt in dat de eisen zijn beschreven in een logica. Aangezien logica geen ruimte laat voor interpretatievrijheden verkleint een dergelijke formalisatieslag de kans op ambiguïteit, misinterpretatie en foutieve implementatie [ST04] van een instantie van de taal. Zo is een UML model een stuk minder strikt te interpreteren dan een ORM model. Gezien het feit dat een prescriptieve architectuurmodelleertaal vooral gebaseerd is op natuurlijke taal is dit helemaal geen luxeprobleem.

Een ander groot voordeel van een formele modelleertaal ligt in de hogere automatiseringsgraad van de taal. Wanneer een taal formeel is gedefinieerd, is deze eenduidiger om te zetten in een geautomatiseerde taal waarin bepaalde syntactische en semantische eisen en/of richtlijnen automatisch gecontroleerd kunnen worden.

Dan moeten de concepten en de taalkundige eisen dus wel formeel beschreven zijn. Dit is binnen dit onderzoek echter onmogelijk, aangezien de uitkomsten uit dit onderzoek eerst in praktijk gevalideerd dienen te worden. Het blijkt namelijk dat er geen consensus bestaat over de te voeren bouwstenen in de taal. Dan is het verspilde moeite om de taal te formaliseren. Laat de taal eerst maar in de niet-geformaliseerde vorm getoetst worden in de praktijk.”

In dit stuk staan een aantal interessante inzichten die nader onderzocht kunnen worden. Zo wordt er in dit stuk alleen gesproken over het formaliseren van architectuurprincipes door middel van visuele modelleertalen. Vandaar dat het interessant is om te kijken naar propositielogica en predicatenlogica. Dit zijn instrumenten waarmee de correctheid van systemen kunnen worden bewezen.

Daarnaast wordt er gesproken over het feit dat er nog geen consensus is bereikt over de bouwstenen van een architectuurprincipe. Deze afstudeerscriptie beschrijft ook deels een zoektocht naar de mogelijke bouwstenen. Startpunt hiervoor is de theorie die Dragon1 beschrijft op dit gebied, aangezien deze methode hier veel aandacht aan besteed en hier concrete voorbeelden bij geeft.

2.3 Theoretisch kader

2.3.1 Verankering

Het kennisgebied waar deze onderzoeksvraag betrekking op heeft is informatiekunde, enterprise architectuur en architectuurprincipes. De onderzoeksvraag is een vraagstuk wat valt binnen het domein van de informatiekundige waarbinnen enterprise architectuur een belangrijke rol speelt. Dit onderzoek zal zich binnen dit gebied richten op architectuurprincipes.

2.3.2 Begrippenkader

De diverse enterprise architectuur (EA) methoden hebben iedere hun eigen definitie van de onderstaande begrippen. De definities voor EA en architectuurprincipe zijn afkomstig van TOGAF en DYA aangezien deze zeer breed gedragen worden binnen het vakgebied en een goede weerspiegeling geven van de algemene functie die aan deze instrumenten wordt toegekend.

Enterprise Architectuur

The fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution [TOGAF]

Architectuur is een consistent geheel van principes en modellen dat richting geeft aan ontwerp en realisatie van de processen, informatievoorziening, technische infrastructuur en organisatorische inrichting. [DYA]

Principe / Architectuurprincipe

Principles are general rules and guidelines, intended to be enduring and seldom amended, that inform and support the way in which an organization sets about fulfilling its mission. A qualitative statement of intent that should be met by the architecture. Has at least a supporting rationale and a measure of importance. [TOGAF]

Architectuurprincipes zijn beleidsuitspraken die in de lijn van die visie richting geven aan de inrichting van de informatievoorziening. [DYA]

Propositie

De inhoud van een zin of bewering. In dit onderzoek wordt dit begrip gebruikt als de verzamelnaam die toegekend wordt aan begrippen als architectuurprincipe, businessrule, regel, principe, doelstelling, uitgangspunt, voorschrift, richtlijn etc. door de Joint NAF / ArchiMate "Business Principles" Workgroup.

2.3.3 Keuzes en vooronderstellingen

Keuzes

Door het gewicht en doorlooptijd van deze afstudeerscriptie zullen concessies worden gedaan wat betreft de breedte van het onderzoek. Dit onderzoek zal betrekking hebben op één praktijkcasus en er zal één architectuurmethode betrokken worden. De casus betreft de architectuurprincipes van Stater NV. De architectuurmethode betreft Dragon1. De keuze om in dit onderzoek de Dragon1 methode als gestructureerde aanpak/methode te gebruiken is omdat deze methode zeer veel aandacht besteed aan het opstellen en formuleren van architectuurprincipes. Daarnaast is deze methode uitgebreid gedocumenteerd en is er de mogelijkheid om de auteur van deze methode te benaderen om het correcte gebruik van deze methode te verifiëren.

De keuze om een casus (en architectuurprincipes) uit de praktijk te gebruiken voor dit onderzoek is om praktijkervaring vast te leggen en de theorie te toetsen in de praktijk.

Afbakening

Dit onderzoek maakt deel uit van een reeks van onderzoeken naar de verschillende vormen en namen die proposities rijk zijn. Binnen dit onderzoek wordt echter de focus gelegd op architectuurprincipes.

Interne validiteit

Door te werken met meerdere gevallen (architectuurprincipes) zal het raamwerk uitgebreid getoetst worden. Daarnaast wordt de classificatie gevalideerd door de eigenaar van de casus.

Geldigheid en betrouwbaarheid

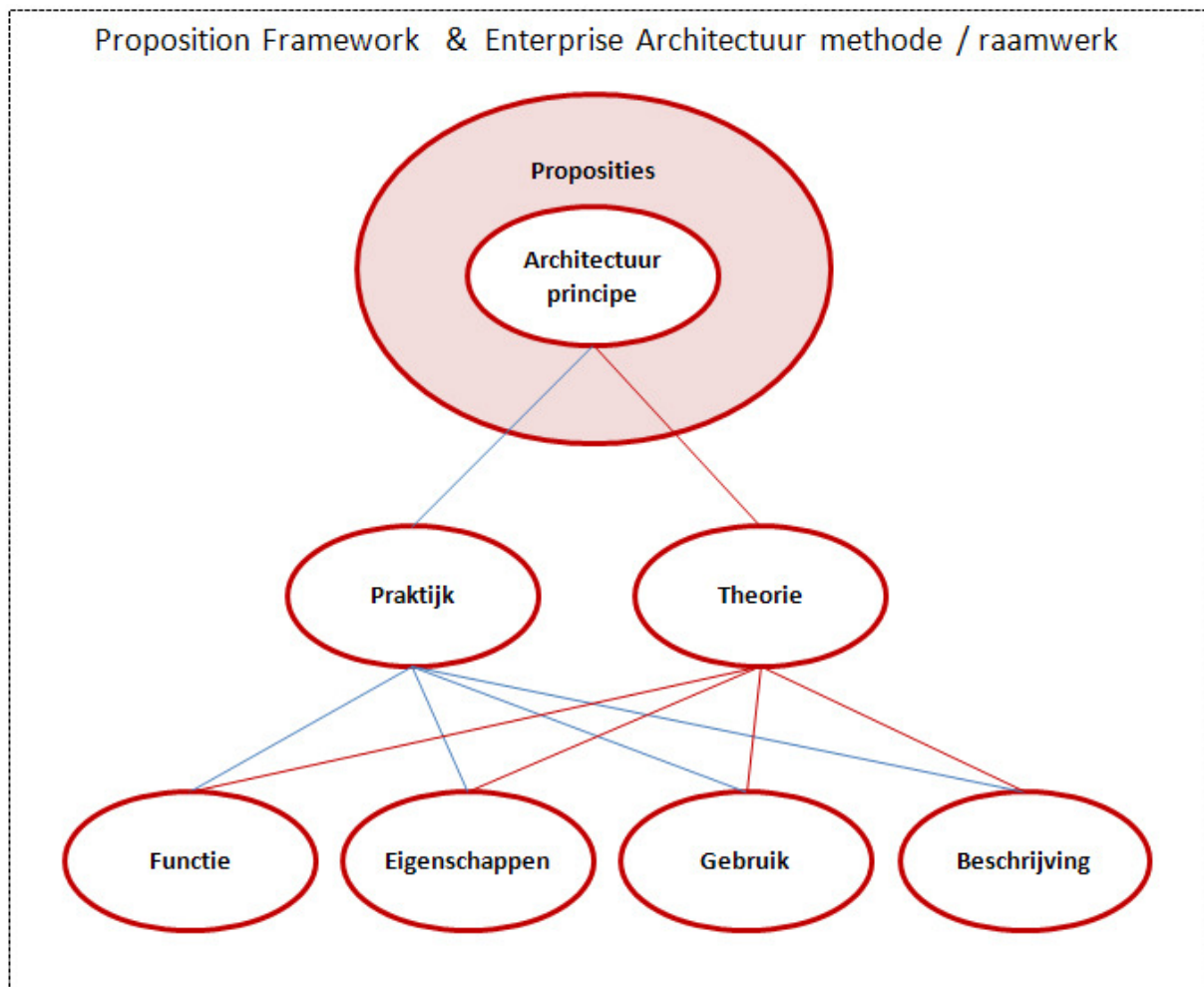
De beargumentatie van de classificatie zal worden vastgelegd. De resultaten en conclusies zullen o.a. berust zijn op meningen, visies en oordelen van de betreffende eigenaar van de casus. Deze verzameling van meningen, visies en oordelen zullen worden vastgelegd en geverifieerd worden door de betrokkenen.

Generaliseerbaarheid

Door het combineren van de resultaten van de andere casussen kunnen de resultaten bevestigd worden.

2.3.4 Conceptueel model

Hieronder is het conceptueel model weergegeven waarop dit onderzoek betrekking heeft. In dit model zal het domein, de variabelen en de relaties tussen deze variabelen beschreven worden.



Figuur 1: Conceptueel model

Domein

Het domein waarop dit onderzoek betrekking heeft zijn architectuurprincipes die gebruikt worden door bedrijven, organisaties en instellingen. Deze architectuurprincipes zijn of worden in de praktijk toegepast en kunnen daardoor inzicht geven in de problematiek die geschetst is in de probleemstelling.

Variabelen

Proposition Framework:

Met het Proposition Framework wordt gemeten welke functies, eigenschappen en beschrijvingen architectuurprincipes in de praktijk en theorie hebben en hoe deze gebruikt worden.

Enterprise Architectuur methode / raamwerk:

Vanuit de theorie van een Enterprise Architectuur methode / raamwerk wordt gekeken welke functies, eigenschappen en beschrijvingen architectuurprincipes in de praktijk en theorie hebben en hoe deze gebruikt worden.

Propositie:

Deze variabele betreft een bepaalde uitspraak die een bepaald label heeft zoals principe, regel, richtlijn, uitgangspunt etc.

Architectuurprincipe:

Een architectuurprincipe is een propositie soort.

Functie architectuurprincipe:

Deze variabele geeft inzicht in de functie(s) die het architectuurprincipe in de praktijk en volgens de theorie heeft.

Eigenschappen architectuurprincipe:

Deze variabele geeft inzicht in de eigenschappen van het architectuurprincipe in de praktijk en volgens de theorie.

Gebruik architectuurprincipe:

Deze variabele geeft inzicht in het gebruik van het architectuurprincipe in de praktijk en volgens de theorie.

Beschrijving architectuurprincipe:

Deze variabele geeft inzicht in de beschrijving(en) van een architectuurprincipe in de praktijk en volgens de theorie.

Relaties

Het Proposition Framework geeft inzicht in de functie, eigenschappen, beschrijving en het gebruik van het architectuurprincipe in de praktijk. Een Enterprise Architectuurmethode geeft inzicht in de functie, eigenschappen, beschrijving en het gebruik van het architectuurprincipe in de theorie.

Het architectuurprincipe vervult een bepaalde rol binnen de verzameling van proposities. De functie, eigenschappen, beschrijving en het gebruik van het architectuurprincipe in de praktijk en theorie bepalen deze rol.

2.4 Methode / Aanpak

Hieronder volgt een beschrijving van de aanpak die gehanteerd is om een antwoord te kunnen geven op de onderzoeksvragen.

Fase 1

Fase 1 beschrijft het deel van het onderzoek waarin het Proposition Framework betrokken is.

Voor het classificeren van de architectuurprincipes van Stater NV door middel van het Proposition Framework v0.9 is de volgende aanpak gekozen:

1. Oriëntatiefase en een korte literatuurstudie naar de achtergronden van het raamwerk en de door Stater NV geleverde documenten.

Het doel hiervan is om te bepalen of het onderstaande experiment plaats kan vinden of eventueel aangepast dient te worden:

Voor iedere propositie uit de casus:

- Noteer de titel en beschrijving van de propositie.
- Noteer de referentie van de bron waar de propositie uit afkomstig is.
- Bepaal de karakteristieken van de propositie in termen van de vijf vectoren die in het raamwerk onderkend worden.

De oriëntatiefase en literatuurstudie is voorafgaande aan het schrijven van het onderzoeksplan uitgevoerd. In een gesprek met de contactpersonen van Stater NV is gebleken dat de beschrijvingen van de architectuurprincipes op sommige vlakken niet voldoende zullen zijn om directe classificatie mogelijk te maken zonder domeinkennis. De classificatie zal daarom in samenwerking met de contactpersonen van Stater NV plaats vinden.

2. Principes classificeren aan de hand van het raamwerk in samenwerking met contactpersonen van Stater NV

De principes zullen voor zover mogelijk geclassificeerd worden waar beperkte domeinkennis voor nodig is. Domeinkennis is opgedaan door middel van een gesprek met de contactpersonen van Stater NV en de aangeleverde documenten van Stater NV. Met deze bagage zal in eerste instantie een classificatie plaats vinden. Deze zal voorgelegd worden aan de contactpersonen van Stater NV en vervolgens in overleg gewijzigd en/of aangevuld. In eerste instantie zullen drie willekeurig gekozen principes worden geclassificeerd en voorgelegd aan Stater NV om de eerste problemen en onduidelijkheden weg te nemen. Vervolgens zullen de overige principes uitgewerkt worden.

Uiteindelijk zullen de contactpersonen van Stater NV hun goedkeuring moeten geven over de classificatie.

3. Verhouding EA methode en Proposition Framework v0.9

Door middel van een experiment zal inzicht verkregen worden over hoe een EA methode zich verhoudt tot het Proposition Framework v0.9 op het gebied van architectuurprincipes. In dit experiment zal informatie verzameld worden over hetgeen wat de methode zegt over wat een architectuurprincipe is door literatuur van de methode te bestuderen. Vervolgens zal een classificatie op basis van deze theorie plaats vinden.

4. Evaluatie Proposition Framework v0.9

Aan de hand van de resultaten en bevindingen die zijn gedaan tijdens het experiment zal een evaluatie plaats vinden van het raamwerk. Hierin zullen bevindingen en aanbevelingen worden gedaan.

5. Evaluatie architectuurprincipes Stater NV

Aan de hand van de resultaten en bevindingen die zijn gedaan tijdens het experiment zal een indicatie gegeven worden in hoeverre de architectuurprincipes goed of slecht zijn. Op basis hiervan zullen ook aanbevelingen worden gedaan.

Fase 2

Fase 2 beschrijft het vervolgonderzoek dat is uitgevoerd naar aanleiding van de vragen die zijn ontstaan in fase 1.

Zoals al geschetst wordt in de probleemstelling, is dit onderzoek zeer exploratief (verkenkend) van aard. Op basis van experimenten, interviews, literatuurstudie en zelfreflectie is getracht een antwoord te geven op de hoofdvraag. Hieronder worden de belangrijkste stappen beschreven die zijn genomen om tot het eindresultaat te komen:

6. Analyseren van kennis en kunde opgedaan in fase 1 van dit onderzoek.

Op basis van de opgedane kennis en verworven resultaten is het vervolgonderzoek bepaald.

7. Visie en mening vastleggen over EA en architectuurprincipes om de startpositie / startpunt duidelijk te maken.

Door de verschillende opvattingen / visies die er zijn over de definitie en functie van architectuur en architectuurprincipes is het van belang om inzicht te geven in de opvattingen en visies die er zijn in het vakgebied en vanuit welke opvatting / visie dit onderzoek wordt voortgezet. Dit is gedaan door het geven van verschillende definities van architectuur en architectuurprincipes van de diverse methoden te combineren met een betoog.

8. Problemen en knelpunten in architectuurprocessen (m.b.t. het formuleren van architectuurprincipes) inzichtelijk maken.

Door middel van een open interview zijn de problemen en knelpunten binnen een architectuurproces inzichtelijk gemaakt. De nadruk heeft hierbij gelegen op de rol van architectuurprincipes binnen dit proces. Daarnaast zijn een aantal architectuurprincipes nader onderzocht om specifieke problemen en knelpunten te ontdekken. Er is gekozen om één praktijkcasus hiervoor als onderzoekselement te selecteren. Het betreft dezelfde casus als in fase 1 (Stater NV). Het resultaat is een lijst van problemen en knelpunten.

9. Praktijkervaring op doen door het herformuleren van architectuurprincipes door middel van een gestructureerd stappenplan (Dragon1).

Om inzicht te verkrijgen in mogelijke oplossingen voor de problemen die onderkend zijn bij stap 8 en een manier om deze problemen te voorkomen zijn een aantal architectuurprincipes uit de betreffende casus geherformuleerd. Op basis van deze resultaten is gekeken in hoeverre de geherformuleerde architectuurprincipes de problemen hadden kunnen voorkomen. Dit is onder andere gedaan op basis van een waardeoordeel van een domeinexpert. Tevens heeft deze stap als doel om inzicht te verkrijgen in de toepasbaarheid van het stappenplan van Dragon1 voor het formuleren van architectuurprincipes.

10. Diverse experimenten uitvoeren om inzicht te verkrijgen en geven in de complexiteit van architectuurprincipes.

Om meer inzicht te verkrijgen in waarom het in de praktijk zo moeilijk blijkt om goede architectuurprincipes te formuleren zijn diverse experimenten uitgevoerd om de functie, definitie en toepassing van architectuurprincipe duidelijk te krijgen. Dit is gedaan op basis van een literatuurstudie en het analyseren van praktijkvoorbeelden. Tevens heeft deze stap als doel om inzicht te verkrijgen in de correctheid van de definitie en functie die aan een architectuurprincipe wordt toegekend door Dragon1.

11. Op basis van de resultaten het huidige stappenplan (Dragon1) analyseren en een verbetervoorstel opleveren.

Op basis van de resultaten uit de voorgaande stappen is gekeken naar de volledigheid en relatie van de attributen van een architectuurprincipe die Dragon1 onderkend. Op basis hiervan zijn nieuwe attributen ter kandidaat gesteld. Hierna is gekeken in welke volgorde het beste invulling kan worden gegeven aan deze attributen. Aan de hand van de resultaten hiervan is een functiemodel opgesteld waarmee de functie van een attribuut kan worden geclassificeerd.

2.5 Structuur onderzoeksrapport

Hieronder wordt in het kort vermeld wat er behandeld wordt in de volgende hoofdstukken:

- In hoofdstuk 3 wordt een overzicht gegeven van het Proposition Framework v0.9. Het is aan te bevelen om deze door te lezen omdat de begrippen en definities gebruikt worden in de overige hoofdstukken van het onderzoeksrapport.
- In hoofdstuk 4 wordt behandeld wat enterprise architectuur is, wat architectuurprincipes zijn en welke visie een sterke invloed heeft op de inhoud van dit onderzoeksrapport.
- In hoofdstuk 5 wordt een introductie van de praktijkcasus gegeven. Het architectuurproces en de architectuurprincipes van de betreffende casus zullen worden besproken.
- In hoofdstuk 6 worden de resultaten van de classificatie van de architectuurprincipes samengevat.
- In hoofdstuk 7 wordt besproken wat volgens een Enterprise Architectuur methode een architectuurprincipe is.
- In hoofdstuk 8 wordt de evaluatie van het Proposition Framework behandeld.
- In hoofdstuk 9 worden aanbevelingen gedaan aan Stater NV hoe ze betere architectuurprincipes kunnen formuleren.
- In hoofdstuk 10 worden de belangrijkste knelpunten en problemen omtrent de architectuurprincipes van Stater NV weergegeven en wordt het resultaat van het herformuleren van deze architectuurprincipes besproken.
- In hoofdstuk 11 worden de resultaten van diverse experimenten die meer inzicht geven in de functie, eigenschappen en definitie van het architectuurprincipe weergegeven.
- In hoofdstuk 12 wordt het stappenplan van Dragon1 nader onderzocht en aangescherpt door de bevindingen uit de voorgaande hoofdstukken te relateren en te verwerken in een vernieuwd stappenplan.
- In hoofdstuk 13 worden de belangrijkste conclusies weergegeven van dit onderzoek en zal een antwoord gegeven worden op de onderzoeksvraag.

3. Proposition Framework

In dit hoofdstuk wordt een overzicht gegeven van het classificatieraamwerk dat is gebruikt tijdens dit onderzoek. Het Proposition Framework v0.9 is een initiatief van het Joint NAF / ArchiMate “Business Principles” Workgroup. Het raamwerk dient als classificatiemiddel voor de verschillende ‘dingen’ die vaak aangeduid worden als ‘principles’, ‘business rules’, ‘rules’, ‘policy’, ‘guidelines’, etc. Deze ‘dingen’ worden in het raamwerk in het algemeen geschaard onder de naam ‘Proposition’. De Nederlandse vertaling hiervan is propositie. Binnen dit deel van dit onderzoeksrapport zal deze term gehanteerd worden. Binnen het raamwerk wordt ‘Proposition’ gedefinieerd als:

“A meaning that is asserted when a sentence is uttered or inscribed and which is true or false”

Uiteindelijk wil men bereiken dat op basis van een classificatie door middel van het raamwerk aangegeven kan worden wat voor een soort uitspraak een bepaalde propositie is en welke relatie er is tussen deze verschillende soorten uitspraken. Het raamwerk omvat vijf attribuuttype klassen [PF08]:

Name	Description
Form	attribute-types pertaining to the form in which the proposition itself is stated.
Object	attribute-types dealing with the identification of the object which the proposition pertains to.
Validity	attribute-types pertaining to the proposition’s claimed/desired validity.
Actors	attribute-types dealing with the identification of those actors who are expected to (be observed to) respect the propositions.
Context	attribute-types pertaining to the contextual embedding of the proposition and its validity claim/desire in terms of proofs or contextual argumentation, examples, etc.

Hieronder wordt een overzicht gegeven van de verschillende attribuuttype klassen met hun attributtypes en de daarbij mogelijke waarden.

Form attribute-types

Attribute type	Classifiers
Level of precision	Informal / Semi-formal / Formal
Level of actionability	Definite / Specific / Actionable

Object attribute-types

Attribute type	Classifiers
Time	A specification of the time-span which the property holds.
Control abstraction	Operations / Structuring / Strategy
Construction abstraction	Valuation / Function/ Construction
Physical abstraction	Physical / Logical / Conceptual
Enablement abstraction	Business / Information / Application / Infrastructure
Organisational range	Application / Information system / Business unit / Enterprise / Ecology
Aspects of dynamic systems	Behavior / Passive structure / Active structure
Systemic order	Operational system / Transformation system

Validity attribute-types

Attribute type	Classifiers
Intricity	Intrinsic / Desired
Probability	Always / Usually / Sometimes
Obedience level	Strict / Overridable / Guiding

Actors attribute-types

Attribute type	Classifiers
Actors	Actors which are expected to (be observed to) respect the propositions

Context attribute-types

Attribute type	Classifiers
Motivation	Intrinsic propositions / Regulating propositions / Guiding propositions
Impact	Discussion of the impact of a proposition
Deployment	Communicate / Construct / Enforce

Het raamwerk zal getoetst worden door middel van het toepassen van het raamwerk op verschillende casussen uit de praktijk. Dit onderzoeksrapport bevat de uitwerking van één van deze casussen.

In bijlage A is de complete versie van het Proposition Framework v0.9 te vinden.

4. Visie op Enterprise Architectuur

Tijdens het verkennen en onderzoeken van het vakgebied Enterprise Architectuur is een bepaalde visie opgebouwd en mening gevormd over wat dit vakgebied inhoudt en welke visies er zijn en tegen welke problemen en onduidelijkheden men binnen dit vakgebied nog vecht. Deze visie en mening wordt gedeeld in dit hoofdstuk. Er zal geen compleet beeld worden geschetst van dit vakgebied, daarvoor moeten te veel aspecten behandeld worden. De belangrijkste aspecten die voor dit onderzoek naar architectuurprincipes van belang zijn zullen worden besproken in de vorm van een betoog. Deze uiteenzetting van visie en mening geeft een beeld van de positie die is ingenomen tijdens dit onderzoek.

4.1 Wat is Enterprise Architectuur?

Zoals al eerder gesteld is hier geen eenduidig antwoord op te geven vanuit de theorie en methoden op het gebied van EA. Er wordt vanuit verschillende perspectieven gekeken naar EA. In dit hoofdstuk zal geen complete opsomming van alle visies en definities worden gegeven. Er zullen een aantal definities worden gegeven die de grote lijnen weergeven. Deze zijn willekeurig gekozen uit het arsenaal van definities dat is bekeken voor dit onderzoek. De gekozen definities moeten dus niet gezien worden als dé definitie, maar als voorbeeld om het betoog te ondersteunen zonder telkens weer nieuwe definities te moeten verzinnen die misschien beter zijn.

Binnen EA zijn er twee soorten architectuur te onderscheiden, namelijk prescriptieve architectuur en descriptieve architectuur. De onderstaande uitleg van Serge Bouwens van Sogeti Nederland BV geeft hier een goed beeld van:

“De eerste stroming benadert architectuur als een richtinggevend instrument dat richtlijnen formuleert ten behoeve van het ontwerpproces. De tweede stroming ziet architectuur meer als een instrument dat de huidige en gewenste situatie vastlegt in modellen.” – [DYA06]

Serge Bouwens geeft al aan dat het vreemd is dat hier een ‘of of’ discussie over wordt gevoerd. Ik ben het eens met deze mening. Het is ‘en en’ in het geval van EA.

De volgende definities geven een beeld van de verschillende visies binnen het vakgebied over wat EA is:

“Architectuur is een consistent geheel van principes en modellen dat richting geeft aan ontwerp en realisatie van de processen, informatievoorziening, technische infrastructuur en organisatorische inrichting.” – [DYA08]

“Enterprise Architecture is about understanding all of the different elements that go to make up the enterprise and how those elements inter-relate.” – [TOGAF08]

“Enterprise Architectuur is de kunde én kunst, ook in de zin van techniek, van het integraal en functiegericht ontwikkelen(ontwerpen en bouwen), veranderen (evolueren, verbeteren en innoveren) en in stand houden, op basis van concepten en rekening houdend met (universele) principes, van organisaties zoals megastructuren, ondernemingen, instellingen, bedrijven en dienstverleningsketen, inclusief systemen en ruimtes die voldoen aan de kwantitatieve en kwalitatieve eisen van de mens. De organisaties, systemen en ruimtes voorzien in diverse behoeftes, voor zowel het individu als de groep” – [Dragon1/07]

Enterprise architectuur is een coherente, consistente verzameling principes, verbijzonderd naar uitgangspunten, regels, standaarden en richtlijnen, die beschrijft hoe de organisatie, de informatievoorziening, de applicaties en de infrastructuur hun vorm hebben gekregen en hoe zij zich voordoen in het gebruik. – [Rijnsbrij04]

Ook aan de definitie van Enterprise Architectuur worden nog veel discussies geweid. Wat valt er onder dit begrip / vakgebied? Hoever reikt het? Wat is het?

Op deze vragen is het lastig een antwoord te geven, vooral als je niet van een definitie wil afwijken. Welke precieze formulering de uiteindelijke definitie heeft is niet zo belangrijk. Het gaat er om dat in veel definities een aantal aspecten die onder architectuur zouden kunnen worden verstaan bij voorbaat worden uitgesloten of weinig tot geen aandacht aan wordt besteed. Er zijn gewoon nog grote gebieden te verkennen binnen het vakgebied.

4.2 Wat zijn architectuurprincipes?

Er zijn binnen het vakgebied EA verschillende opvattingen over wat een architectuurprincipe nu precies is en er zijn veel verschillende definities van architectuurprincipes te vinden. Hieronder volgen een aantal definities. Deze worden hier weergegeven omdat o.a. uit deze definities getracht wordt een beter beeld te krijgen van de diversiteit. Alle eigenschappen die in de literatuur toegedicht worden aan architectuurprincipes zijn van belang binnen EA. Dit wil echter niet zeggen dat deze allemaal toebehoren aan het begrip architectuurprincipe. In deze afstudeerscriptie wordt dan ook gepoogd te laten zien dat het een architectuurprincipe (of principe of welke naam het ook toegekend dient te worden) uit diverse lagen en onderdelen bestaat die allen noodzakelijk zijn. Architectuurprincipes vormen deels het fundament van een architectuur en dienen goed beschreven te zijn.

Veel bedrijven en instellingen hebben moeite met het formuleren van architectuurprincipes. Men schrijft vaak wat men wil bereiken op als architectuurprincipe. Dit verschijnsel leidt tot ontevredenheid en te weinig resultaat in een architectuurproces.

De volgende definities geven een goed beeld van de huidige definities die gehanteerd worden:

“Principles are general rules and guidelines, intended to be enduring and seldom amended, that inform and support the way in which an organization sets about fulfilling its mission.” – [TOGAF08]

“Architectuur is vorm geven aan visie. Architectuurprincipes zijn beleidsuitspraken die in de lijn van die visie richting geven aan de inrichting van de informatievoorziening.” – [DYA08]

“Architectuurprincipe kent in de praktijk de volgende drie betekenissen met allemaal het zelfde doel:

- 1. Een strategisch afgewogen holistische bedrijfsregel die men veel waarde toekent en daarom als strategisch uitgangspunt hanteert*
- 2. Een strategische enterprise-wide en IT-gerichte richtinggevende uitspraak die een toekomstvaste waarheid is en zelden wordt gewijzigd*
- 3. De gehandhaafde werking van iets (zoals een systeem, concept of fenomeen) dat zorgt voor een gereguleerde situatie en beleving in een contextloze ruimte waarmee bepaalde toestanden, gebruiken of toepassingen van iets anders mogelijk worden gemaakt of juist worden beperkt” - [Dragon1/08b]*

“Principes zijn richtinggevende uitspraken ten behoeve van essentiële beslissingen, een fundamenteel idee bedoeld om een algemene eis te vervullen. Principes dienen te worden geconcretiseerd naar zaken die moeten, dat zijn de regels en standaarden, en zaken die verstandig zijn: de richtlijnen, ook wel ‘best practices’ genoemd.” – [Rijsenbrij04]

“Volgens Dragon1 is de definitie van principe: De gehandhaafde wijze waarop iets, zoals een situatie, handeling, ruimte, systeem, domein concept of fenomeen, werkt, in elkaar steekt of wordt gedaan, met een bepaald (beoogd) resultaat als gevolg.” – [Dragon1/08a]

Bij deze diverse definities horen ook weer een lijst van eigenschappen / attributen die beschreven moeten worden en hoe de structuur van een architectuurprincipe er uit hoort te zien. Een korte samenvatting van de huidige stand van zaken:

Er is geen consensus binnen het vakgebied EA over wat een architectuurprincipe is en wat voor uitspraak (statement) nu een architectuurprincipe is en wanneer een uitspraak nu een architectuurprincipe genoemd mag worden. Mijn mening hierover is dat men het eigenlijk over totaal verschillende dingen heeft en de diverse methoden dus de ‘naam’ architectuurprincipes op verschillende dingen plakt. Het gaat er niet om dat één van deze definities de juiste is maar dat er inzicht komt in de relaties tussen alle definities en op basis daarvan iets een naam gegeven kan worden.

Voor deze afstudeerscriptie is gekozen om de definitie die Dragon1 hanteert als basis te nemen om een aanzet te geven tot een model dat wel inzicht geeft en duidelijkheid biedt. Deze keuze is gemaakt omdat Dragon1 een structuur biedt waarin ieder zijn definitie wel een plaats kan vinden.

Mijn visie op de discussie is als volgt:

Je hebt het containerbegrip principes. Dit wordt vaak als synoniem gebruikt voor architectuurprincipe. Dit feit alleen al zorgt voor veel verwarring. Is dit nu hetzelfde? Mag het door elkaar gebruikt worden? Dit zijn vragen die gesteld kunnen worden. Naast het begrip wordt ook nog door veel methoden onderscheid gemaakt in andere soorten principes. Ook de definities van deze soorten principes verschillen weer per methode. Hier volgt een opsomming van diverse soorten waar o.a. onderscheid tussen wordt gemaakt:

“Constructie principes, Structuur principes, Ontwerp principes, Bouw/Realisatie principes, Systeem principes, Fenomeen principes, Concept principes, Domein principes, Solution principes, (stijl)Element principes, Artifact principes, Keten principes, Enterprise principes, Besturings principes, Bedrijfs(functie) principes, Informatie(voorziening) principes, IT-Infrastructuur principes, Security principes” – [Dragon1/08b]

Deze lijst is afkomstig van de Dragon1 methode. Binnen het vakgebied EA worden nog andere soorten principes genoemd. Het doel van het weergeven van deze lijst is om beeld te geven van de diversiteit.

Het begrip concept wordt vaak gerelateerd aan het begrip architectuurprincipe. Deze twee begrippen zijn inherent aan elkaar verbonden. Een concept wordt beschreven door een architectuurprincipe, maar een concept kan ook weer een architectuurprincipe beschrijven. In de praktijk zie je vaak dat men eigenlijke concepten opschrijft in plaats van architectuurprincipes. Men schrijft bijvoorbeeld het concept ‘Single sign-on’ op als architectuurprincipe. Dit is mijn inziens echter een concept of oplossing die toegepast kan worden en geen architectuurprincipe.

De discussie over wat een architectuurprincipe is moet eigenlijk nog uitgesteld worden. Er moet eerst duidelijkheid worden geschept over wat het vakgebied EA nu is en tot waar het reikt en wat de rol van een architect is. Deze discussie loopt namelijk parallel met de discussie over architectuurprincipes. Mijn visie is hier dat alle taken en doelen die door de diverse methoden in het vakgebied EA worden genoemd behoren tot het vakgebied EA. Sommige taken en onderdelen worden nog zwaar onderbelicht, zoals het visualiseren van architectuur en het fundamenteel onderbouwen van architectuur. Dit behoort wat mij betreft tot EA. Het fundamenteel onderbouwen van architectuur is belangrijk. De vraag naar EA is o.a. ontstaan uit de behoefte om ondernemingen en systemen te creëren die stabiel zijn, beter aansluiten bij de wensen van mensen, het beter communiceerbaar maken van de veranderingen die worden doorgevoerd etc. Een ander standpunt wat ik in neem is dat een architect (en zijn architectuur medewerkers) alle facetten belichten en taken hebben die door de diverse methoden worden toegekend. Dragon1 stelt dat een architect moet inspireren, creëren, communiceren en controleren. Wat mij betreft vormen deze vier begrippen een goed beeld van wat een architect moet doen.

Het is tijd om met een frisse blik te kijken naar de geschetste problematiek. Los van de diverse definities en visies dienen uiteindelijk complete methoden voor werken onder architectuur opgeleverd te worden die effectief zijn. Deze methoden kunnen verschillend zijn en bepaalde aspecten van EA minder belichten of overlaten aan anderen, maar dienen wel aan te sluiten op de situatie waarin architectuur toegepast gaat worden. Ook al valt een taak of verantwoordelijkheid wel of niet onder architectuur, de taak moet uiteindelijk wel uitgevoerd worden en een verantwoordelijkheid moet uiteindelijk wel genomen worden. Er kunnen binnen het vakgebied best verschillende definities zijn, als ze maar binnen een methode op elkaar zijn afgestemd zodat deze compleet is.

De volgende bevindingen zijn tijdens dit onderzoek o.a. gedaan over de wijze waarop architectuur-principes geformuleerd worden:

- Men beschrijft een wens of iets wat men wil bereiken.
- Men beschrijft niet hoe of met wat men dit wil bereiken.
- Men beschrijft niet waarom men dit wil bereiken.

Hierdoor worden stellingen opgeschreven die onwaar kunnen zijn en waarvan niet te bewijzen is dat ze in de praktijk voor resultaten gaan zorgen of dat deze resultaten gewenst zijn.

5. Casus

In dit hoofdstuk wordt inzicht gegeven in de praktijkcasus die een grote rol heeft gespeeld in dit onderzoek. De casus voor dit onderzoek is afkomstig van Stater NV. Dit hoofdstuk beschrijft de context van de casus, het architectuurproces van Stater NV en de architectuurprincipes die als onderzoeksobject hebben gediend in dit onderzoek.

5.1 Inleiding casus

Stater is een onafhankelijke leverancier van diensten aan hypotheekverstrekkers in verscheidene Europese landen, met een sterke positie in Nederland. Stater is in 1997 gestart met het beheer van hypotheeken als onafhankelijk opererende dochter van Bouwfonds. Door de uitbreiding van het dienstenpakket en de snelle internationale ontwikkeling, is Stater in 1 januari 2001 zelfstandig geworden. Inmiddels is zij uitgegroeid tot een internationale speler met meer dan 800 medewerkers. Het hoofdkantoor is gevestigd in Amersfoort, daarnaast zijn er vestigingen in Leusden, Bonn (Duitsland) en in Brussel (België). Stater beheert ruim 1 miljoen hypotheeken voor ruim 20 geldgevers in Nederland, Duitsland en België. In Nederland zijn ze verantwoordelijk voor ruim 30% van alle hypotheeken. Om het beheer hier van in goede banen te kunnen lijnen is er een nieuwe omvangrijke applicatie nodig nu het huidige systeem zijn end-of-life status bereikt heeft. Deze nieuwe applicatie genaamd Estate is ontwikkeld onder architectuur waarbij de architectuurprincipes leidend zijn geweest.

Estate is een strategisch programma van Stater dat gestart is in 2004, met het doel een compleet nieuwe ICT-oplossing binnen Stater te implementeren. De aanleidingen voor het starten van het Estate-programma waren divers en de belangrijkste worden genoemd in de ambitie en businessdrivers. Concreet richt het Estate-programma zich op de realisatie van een ICT-oplossing waarmee een aantal doelen bereikt kunnen worden. Sleutelbegrippen zijn flexibiliteit, internationalisatie, schaalbaarheid, integratie met grote klanten en kostenefficiëntie. Het vervangen van de huidige ICT-oplossing, iSHS, is een ingrijpende operatie die een grote inspanning vergt van de medewerkers, een hoge impact heeft op de organisatie, veel geld kost en enkele jaren gaat duren. Naast de ontwikkeling van de nieuwe ICT-oplossing moet het iSHS gewoon blijven bestaan en worden onderhouden, totdat deze kan worden vervangen door de nieuwe oplossing.

De architectuurprincipes die opgesteld zijn voor het Estate-programma zijn gebruikt als casus voor dit onderzoek.

5.2 Architectuurproces Stater NV

In dit hoofdstuk zal in grote lijnen het architectuurproces van Stater NV worden beschreven. Hiermee wordt een beeld geschetst van de wijze waarop zij architectuur zien, wat men verstaat onder architectuurprincipes, hoe men om gaat met architectuur en hoe dit alles zich tot nu toe heeft ontwikkeld binnen Stater NV. Deze informatie is verzameld door middel van een interview met Wim van Stokkum. Hieronder volgen de belangrijkste passages uit dit interview:

Architectuur is iets nieuws binnen stater NV. Een kleine groep mensen zijn dit aan het opzetten. De achtergrond van deze mensen is systeemontwikkeling. De motivatie achter het willen werken onder architectuur is dat men een ICT-oplossing wil ontwikkelen die meer flexibiliteit en transparantie bewerkstelligt en minder beheer en onderhoudskosten heeft.

De architectuurprincipes zijn geformuleerd door de voorbeeld principes uit cursussen OO en CBD te combineren met actuele problemen en de context van Stater NV. Er is gekeken naar OO en Archimate. Ze zijn hierin begeleid door het kenniscentrum Cibit. De definitie die gehanteerd is voor architectuur is afkomstig van DYA:

Onder architectuur verstaan we een consistent geheel van principes en modellen dat richting geeft aan ontwerp en realisatie van de processen, organisatorische inrichtingen, systemen en technische infrastructuur van de organisatie. De architectuur kan in zekere zin gezien worden als het geheel van afspraken dat ervoor zorgt dat individuele ontwikkelingen aansluiten op elkaar en op het overkoepelend bedrijfsbelang. – [DYA] & [STA06]

De bijbehorende definitie van architectuurprincipe volgens DYA:

Architectuur is vorm geven aan visie. Architectuurprincipes zijn beleidsuitspraken die in de lijn van die visie richting geven aan de inrichting van de informatievoorziening. – [DYA08]

Binnen Stater NV zijn ze zoekende geweest. De ontwikkeling van het gedachtegoed architectuur heeft parallel plaatsgevonden met vernieuwing van het systeem. Het is dus een leerproces geweest met afstemmingsproblemen gedurende het traject.

Werken onder architectuur is ook voor de Business van Stater nog onwennig, alsmede ook voor de ontwikkelaars van het huidige systeem iSHS. Het nut is niet altijd even duidelijk. Architectuur is iets abstracts. Het architectuurteam heeft door middel van presentaties proberen duidelijk te maken wat ze aan het doen zijn en wat de meerwaarde is.

De algemene principes zijn aan het begin van het project mondeling met elkaar afgestemd in workshops. Er is mondeling verzekerd van het feit dat iedereen de principes hetzelfde geïnterpreteerd heeft. Het project dat verantwoordelijk is voor de nieuwe ICT-oplossing heeft de principes zelf vertaald naar bouwrichtlijnen en -principes, die ontwikkelaars zelf altijd in praktijk brengen bij diverse projecten.

Tijdens het traject heeft regelmatig afstemming plaats gevonden en is de architectuur meegenomen in de ontwikkeling. Er is getoond hoe de architectuurrichtlijnen in de praktijk gebracht dienen te worden. Daarnaast hebben regelmatig audits plaatsgevonden van externe architectuur organisaties als Gartner. Voor deze audits zijn echter niet de Stater principes gebruikt, maar hun eigen frameworks. Ze hebben hierbij gekeken naar de architectuur, de opzet van componenten, en ook naar de programmatuur zelf. Ook de projectorganisatie is belicht. Daarnaast hebben de architecten van Stater zelf laat in het traject een audit uitgevoerd o.b.v de architectuurprincipes die ze tot dan toe hadden opgesteld.

Toelichting:

De reden voor het willen werken onder architectuur dat Stater NV voor ogen heeft is het bouwen van een betrouwbare, flexibele, transparante, toekomstvaste oplossing. Deze reden is legitiem en ook het doel van werken onder architectuur zoals dit door veel architectuurmethoden wordt beschreven.

Er zijn bestaande principes gehanteerd, die toegespitst zijn op de context van Stater NV. Dit is een goede zaak, maar er is geen 'check' uitgevoerd of deze principes wel hun waarde hebben bewezen in het verleden en of deze aansluiten bij de business van Stater NV.

Het bepalen van hoe Stater NV wil werken onder architectuur en welke functie het architectuurprincipe hier binnen vervuld, heeft parallel plaats gevonden met de ontwikkeling van de ICT-oplossing die onder architectuur gebouwd dient te worden. Het is dus lastig te bepalen in hoeverre de ICT-oplossing onder architectuur is gebouwd en welke invloed dit heeft gehad.

Het mondeling afstemmen van het feit dat iedereen de principes op dezelfde manier heeft geïnterpreteerd kan een gevaar vormen voor de vertaling naar bouwrichtlijnen. De architect dient deze klus te begeleiden.

Architectuurprincipes dienen vooraf men begint met de oplossing te bouwen vast te staan en doorvertaald te zijn naar bouwrichtlijnen. Deze bouwrichtlijnen zijn toetsbaar en kunnen als input dienen voor een audit. Op basis hiervan kan gemeten worden in hoeverre de vooraf opgestelde architectuurprincipes zichtbaar zijn in de uiteindelijke oplossing.

5.3 Overzicht architectuurprincipes

De architectuurprincipes zijn afkomstig uit de Stater Architectuurbeschrijving. Hieronder volgt een overzicht van deze architectuurprincipes [STA06]. De architectuurprincipes zijn verdeeld over vier hoofdgroepen:

Flexibiliteit op meerdere gebieden

- **Procesmodulariteit:** Processen zijn onderverdeeld in activiteiten die onderling onafhankelijk en ontkoppeld zijn; dit geldt zowel voor het primaire proces als ondersteunende activiteiten (controles, e.d.). Iedere activiteit kan zowel intern (bij Stater) als extern (bij de klant of elders) zijn belegd.
- **Productflexibiliteit:** Er mogen geen technische redenen aanwezig zijn die een bepaalde combinatie van productkenmerken verhinderen. Legale combinaties van productkenmerken worden via instellingen/inregelingen gedefinieerd. De ICT-oplossing moet voorbereid zijn op het toevoegen van rekenregels zonder aanpassing van bestaande software. De flexibiliteit van de ICT-oplossing moet zodanig zijn dat niet-woninggebonden leningen ondersteund kunnen worden.
- **Flexibele tijdslijnen:** De periodieke (administratieve) afsluiting kan per individuele financiële afspraak ingeregeld worden. Dagelijks vindt afsluiting van relevante afspraken plaats.
- **Billing:** De ICT-oplossing maakt het mogelijk om op basis van een verzameling gebruiksgegevens verschillende vormen van tarifiering aan te bieden.
- **Systeemmodulariteit:** De ICT-oplossing is onderverdeeld in diensten/modules die onderling zijn ontkoppeld en in isolatie kunnen worden afgenomen.
- **Inregelbaarheid:** De ICT-oplossing heeft meerdere niveaus van inregelbaarheid, waarbij de definities op generiekere niveaus overdraagbaar zijn naar specifiekere niveaus. Zo'n niveau wordt een aspectlevel genoemd. De basis voor dit model is onderscheid tussen generieke inregelingen, landspecifieke inregelingen, klantspecifieke inregelingen.

Integratie met klanten en de omgeving

- **Ketenintegratie:** De choreografie van de activiteiten in het totale hypotheekproces is per klant inregelbaar en kan zowel intern als extern zijn belegd.
- **Straight Through Processing:** De verwerking van aangeboden verzoeken (zoals aanvragen) dient door de ICT-oplossing automatisch uitgevoerd te kunnen worden.
- **Beschikbaarheid:** De ICT-oplossing moet zodanig ingericht zijn dat er geen technische beperkingen zijn aan de beschikbaarheid ervan voor geautoriseerde gebruikers en processen.
- **Traceerbaarheid:** Handelingen van gebruikers en services dienen controleerbaar te zijn en gelogd te worden. Fouten dienen geregistreerd te worden.
- **White-/private-labeling:** Alle user-interfaces en alle brieven en rapportages van alle modules kunnen worden ingesteld naar het uiterlijk en de wensen van de klant.
- **Rapportage:** De ICT-oplossing bevat geen ingebouwde kennis over rapportages. De rapportage functionaliteit is ontkoppeld van de geautomatiseerde ondersteuning van het primaire proces. Alle geautoriseerde partijen krijgen toegang tot hun rapportagedata, in de door hen gewenste vorm van doorsneden en aggregaties.
- **Koppelbaarheid:** De ICT-oplossing moet eenvoudig kunnen worden gekoppeld aan andere systemen, softwarepakketten, databases en netwerken.

- **Multi-channeling:** De modules van de ICT-oplossing kunnen via diverse (technische) kanalen worden gebruikt, en worden niet kanaalspecifiek geïmplementeerd. Deze modules kunnen zowel interactief (via een user-interface) als programmatisch worden gebruikt.
- **Autorisatie en Beveiliging:** Toegang tot functies en gegevens van de ICT-oplossing dient op verschillende niveaus (gebruiker, geldgever, tussenpersoon, etc.) te worden geautoriseerd. Hierbij spelen zaken als single logon, beveiligde interacties tussen de partijen en 'Identity and access management'.

Internationalisatie

- **Meertaligheid:** De ICT-oplossing biedt ondersteuning voor meerdere talen – zowel interactief, in rapportages als in ondersteuning (Help functies). De gekozen taal kan afhankelijk zijn van verschillende aspecten (gebruiker, klant, land, etc.).
- **Multi currency:** De ICT-oplossing biedt ondersteuning voor meerdere valuta – zowel interactief, in rapportages als in ondersteuning (Help functies). Binnen één omgeving van een geldgever wordt met één valuta gewerkt.
- **Compliance:** De ICT-oplossing conformeert zich aan nationale en internationale wet- en regelgeving (IAS/IFRS, BASEL II, WFD, SOXA, SAS70).

Kostenefficiëntie

- **Schaalbaarheid:** De ICT-oplossing is in staat meerdere miljoenen leningen te beheren zonder substantieel verlies van online- en batch-performance. Schaalbaarheid heeft ook betrekking op andere aspecten, waaronder aantal landen, aantal producten, aantal rapportages, aantal gebruikers, etc.
- **Onderhoudbaarheid:** De structuur van de ICT-oplossing is zodanig opgezet dat wijzigingen beperkt blijven tot een minimaal aantal componenten en eenvoudig zijn door te voeren.
- **Beheerbaarheid:** De kosten en benodigde inspanning voor het beheren van de ICT-oplossing dienen minimaal te zijn.

De architectuurprincipes worden beschreven door middel van twee attributen, namelijk de titel van het architectuurprincipe en de omschrijving van het architectuurprincipe.

Principes formuleren volgens DYA en TOGAF:

Stater NV heeft de DYA definitie voor architectuur gehanteerd. Net als TOGAF dient volgens DYA aan de attributen Naam, Statement, Rationale en Consequenties (Implications) invulling gegeven te worden bij het formuleren van een principe. Dit is door Stater NV niet consistent uitgevoerd. De rationale (motivatie) en de consequenties ontbreken in de beschrijving van de architectuurprincipes. Hierdoor is het vooraf niet duidelijk welke impact het principe heeft op de te bouwen ICT-oplossing en is het voor de business van Stater NV lastig te beoordelen of het hanteren van deze principes wel zo noodzakelijk is.

6. Classificatie architectuurprincipes Stater NV

In dit hoofdstuk zijn de resultaten van de classificatie van de architectuurprincipes samengevat. Er wordt beschreven hoe de architectuurprincipes van Stater NV zich verhouden tot het raamwerk en welke bevindingen zijn gedaan tijdens en op basis van deze classificatie. Vervolgens is geconcludeerd welke eigenschappen een architectuurprincipe heeft op basis van de classificatie en de bevindingen. Op basis van deze resultaten zijn aanbevelingen gegeven voor Stater NV omtrent het formuleren en hanteren van architectuurprincipes.

Voor een uitgebreid overzicht van de classificatie wordt verwezen naar bijlage B. Voor het complete document waarin de classificatie is beschreven wordt verwezen naar het document Classificatie Architectuurprincipes Stater NV v1.

Dit hoofdstuk geeft antwoord op de volgende deelvraag:

Welke eigenschappen hebben architectuurprincipes die in de praktijk gebruikt worden?

- Welke eigenschappen zijn toe te kennen aan een architectuurprincipe op basis van de classificatie (door middel van het Proposition Framework) van de architectuurprincipes van Stater NV?

6.1 Werkwijze

De werkwijze van de classificatie is al eerder besproken in de onderzoeksopzet. Echter is tijdens het experiment de volgende stap toegevoegd:

- Noteer de motivatie / reden voor de toegekende classificatie van de propositie

Deze stap is toegevoegd omdat zoals eerder geschreven de architectuurprincipes in onvoldoende mate beschreven zijn en domeinkennis vereist is. Daarnaast is het raamwerk nog in de concept fase en is er nog discussie mogelijk over de interpretatie van de verschillende domeinen, attribuuttypes en bijbehorende waarden. Het noteren van de motivatie, keuzes, discussies en afwegingen geeft inzicht in de keuzes die gemaakt zijn bij de classificatie.

6.2 Resultaat classificatie

Alle architectuurprincipes hebben betrekking op het Estate-programma van Stater NV. Per domein wordt de classificatie die van toepassing is beschreven. Het onderstaande resultaat is gebaseerd op de inhoud van het document Classificatie Architectuurprincipes Stater NV v1. De dikgedrukte woorden zijn attribuutklassen, attribuuttypes of classifiers uit het Proposition Framework v0.9.

6.2.1 Form attribute-types

Alle principes zijn als **Informal** geclassificeerd bij het attribuuttype **Level of precision**. Er wordt geen gebruik gemaakt van een gecontroleerde taal of formele notatiewijze waardoor de classificatiemogelijkheden **Semi-formal** en **Formal** al afvallen. De omschrijving van de architectuurprincipes valt onder de noemer verhalende beschrijving.

Alle principes op één na zijn als **Definite** geclassificeerd bij het attribuuttype **Level of actionability**. Bij deze principes mist een duidelijke omschrijving van de meetinstrumenten. Alleen het principe 'Compliance' heeft de classificatie **Actionable** toegekend gekregen. In de omschrijving van het principe worden meetinstrumenten genoemd. Echter wordt niet precies beschreven hoe de meting uitgevoerd moet worden. Hier is dus nog discussie over mogelijk.

6.2.2 Validity attribute-types

Alle principes hebben bij **Intricity** als classificatie **Desired** toegewezen gekregen. Alle principes beschrijven wat gewenst is in het 'system/enterprise'.

Meer dan de helft van de principes heeft de classificatie **Always** toegekend gekregen (12 van de 21) bij het attribuuttype **Probability**. Aan 7 principes is de classificatie **Usually** toegekend en de classificatie **Sometimes** is 2 keer van toepassing.

De helft van de principes heeft de classificatie **Strict** toegekend gekregen (12 van de 21) bij het attribuuttype **Obedience level**. Aan 8 principes is de classificatie **Overridable** toegekend en de classificatie **Guiding** is 1 keer van toepassing geweest.

6.2.3 Object attribute-types

Er is geen specifiek tijdslimiet verbonden aan de principes. Ze moeten altijd van toepassing zijn zolang het Estate programma in ontwikkeling en gebruik is.

Op de principes 'Rapportage' en 'Multi-channeling' (geclassificeerd als **Logical**) na hebben alle principes bij het attribuut-type **Physical abstraction** de classificatie **Conceptual** toegewezen gekregen aangezien deze principes beschrijven wat voor system/enterprise nodig is, zonder daarbij in te gaan op wat voor IT, mensen en machines hiervoor nodig zijn.

De classificatie bij de attribuut-types **Control abstraction**, **Construction abstraction**, **Enablement abstraction** en **Aspects of dynamic systems** is zeer uiteenlopend. Binnen de groep principes is hier weinig consistentie in te vinden. Voor een overzicht en beschrijving van de keuzes die gemaakt zijn bij de classificatie wordt verwezen naar het document Classificatie Architectuurprincipes Stater NV v1.

Meer dan 75% (16 van de 21) van de principes zijn bij **Organisational range** geclassificeerd als **Information system**. Dit is naar verwachting aangezien de principes betrekking hebben op het Estate-programma.

Bij **Systemic order** is bij alle principes de classificatie **Operational system** van toepassing.

6.2.4 Context attribute-types

Alle principes zijn bij **Motivation** geclassificeerd als **Regulating propositions**. Hiervoor is gekozen omdat alle principes verwezenlijkt moeten worden in het Estate programma. De principes zijn niet richtinggevend maar leidend. Waar een duidelijk risico ten grondslag ligt is gekozen voor **Risk-based**. De overige principes zijn verfijningen van andere principes of van hogere doelen. Dit laatste is vaak van toepassing.

Bij **Impact** zijn punten verzameld die afkomstig zijn van de contactpersonen van Stater en van de documentatie die door Stater beschikbaar is gesteld. Hier is in de textuele beschrijving van de architectuurprincipes geen aandacht aan besteed.

Bij **Deployment** is voor alle principes de classificatie **Communicate** van toepassing.

6.2.5 Actors attribute-types

Bij de **Actors attribute-types** zijn de volgende actors te onderscheiden.

- Architectuurteam & ontwerpteam (ontwikkelteam).
- Stater (Enterprise)
- Klant
- Estate programma (Het informatiesysteem / ICT-oplossing)

Alle principes hebben betrekking op het ontwikkelteam, Stater en het Estate-programma. Bij een aantal principes komt daar de Klant nog bij. Hierbij is gekeken tot hoever de scope van een principe reikt. Deze hangt nauw samen met het attribuuttype Organisational range.

6.3 Bevindingen

6.3.1 Inconsistentie in wijze van formuleren

Tijdens de classificatie is opgevallen dat er weinig consistentie is te vinden in de wijze waarop de architectuurprincipes zijn geformuleerd. De mate van detail waarmee de architectuurprincipes zijn geformuleerd vertoont grote verschillen. Hierdoor verschilt de bruikbaarheid en begrijpbaarheid van de verschillende architectuurprincipes.

6.3.2 Relatie Probability en Obedience level

Er is een duidelijke relatie te vinden tussen de attribuut-types **Probability** en **Obedience level**.

Wanneer de classificatie **Always** van toepassing is bij **Probability**, wordt deze in de casus altijd gevolgd door de classificatie **Strict** bij **Obedience level**.

Wanneer de classificatie **Usually** of **Sometimes** van toepassing is bij **Probability** wordt deze in deze casus altijd gevolgd door de classificatie **Overridable** of **Guiding** bij **Obedience level**. Een combinatie met **Strict** komt niet voor.

6.3.3 Gewenst of observeerbaar?

Bij de classificatie van de attribuut-types **Probability** en **Obedience level** valt ook het volgende op:

De omschrijving van een aantal principes geven in eerste instantie de indruk dat de classificatie **Always** en **Strict** van toepassing zijn. Na overleg blijkt echter dat dit niet het geval is en dat er wel degelijk uitzonderingen mogelijk zijn. Hieronder volgt een voorbeeld (White-/private-labeling):

Alle user-interfaces en alle brieven en rapportages van alle modules kunnen worden ingesteld naar het uiterlijk en de wensen van de klant.

Hier wordt steeds gesproken over 'alle' terwijl dit in de praktijk niet het geval is. De volgende opmerking van de domeinexpert laat dit zien:

"Overtuigbaar voor de componenten die geen multi-label gebruikers groep kent."

Dit maakt het toekennen van de juiste classificatie lastig. Moet de classificatie van toepassing zijn op hetgeen wat is vastgelegd, wat observeerbaar is of wat gewenst is.

6.3.4 Propositie soort

Sommige principes hebben meer weg van een functionele eis, regel of richtlijn dan van een architectuurprincipe. Voor deze principes is de classificatie voor de attribuuttypes **Control abstraction** en **Aspects of dynamic systems** lastig te bepalen, omdat ze niet in te delen zijn in één van de keuzemogelijkheden. Ook bij het attribuut-type **Enablement abstraction** is bij deze principes veel ruimte voor interpretatie. Deze attribuut-types zijn voor de principes van deze casus over het algemeen lastig te classificeren. Bij sommige principes lijken meerdere classificaties mogelijk en zinvol te beargumenteren. Ook doordat het raamwerk nog in ontwikkeling is en er nog ruimte is voor interpretatie kan er over deze classificaties nog gediscussieerd worden.

Bij het attribuut-type **Construction abstraction** zijn bij verschillende principes vaak meerdere classificaties mogelijk. Hier is in overleg gekozen voor de meest passende classificatie. Bij dit attribuuttype is dezelfde opmerking van toepassing als bij het bovenstaande punt namelijk dat er discussie mogelijk is over deze classificatie.

6.3.5 Motivatie

Bij een aantal principes is de keuze bij **Motivation** tussen **refinement-based** en **risk-based** niet te beargumenteren. Beide zijn niet direct van toepassing maar zouden eventueel wel herleid kunnen worden. Dit is echter niet uit de omschrijving van het principe te halen. De motivatie is in veel gevallen een hoger doel bereiken of een gewenste functionaliteit van de ICT-oplossing realiseren. Je kunt hier spreken van een verfijning van één van de hoger gestelde doelen. De classificatie **refinement-based** kan echter niet toegewezen worden omdat het geen directe verfijning is van een ander principe uit de groep van principes.

6.3.6 Deployment strategie

Er is onzekerheid bij de classificatie omtrent het attribuuttype **Deployment**. In eerste instantie is gekozen voor de classificatie **Construct**. De gedachten hierachter was dat het systeem en de procedures rondom het systeem zodanig worden ontworpen zodat de principes gehandhaafd worden in de praktijk. Deze classificatie was toegekend onder de veronderstelling dat het systeem (de ICT-oplossing) zich moet conformeren aan de principes. Uiteindelijk is gekozen voor de volgende invalshoek; het ontwikkelteam van de ICT-oplossing moet zich conformeren aan deze principes. In dit geval is **Communicate** de juiste classificatie.

6.4 Conclusie

Het doel van de classificatie van de architectuurprincipes is om te achterhalen wat architectuurprincipes nu zijn en welke vorm ze terug te vinden zijn in de praktijk. Op basis van de classificatie zijn de volgende eigenschappen toe te kennen aan een architectuurprincipe zoals ze onderkend worden binnen Stater:

- Een principe beschrijft wat gewenst is in een 'system/enterprise'. De classificatie **Desired** is hier van toepassing.
- De **Probability** van een principe kan verschillen. Bij voorkeur is hier **Always** of **Usually** van toepassing.
- Het **Obedience Level** van een principe is **Strict** of **Overridable**. Het afwijken van een principe is bij voorkeur niet gewenst.
- Een principe is een tijdsgebonden wetmatigheid, in zoverre dat als een principe gehanteerd wordt door een actor het ook gehanteerd zal blijven worden voor onbepaalde tijd.
- Een principe kan zowel refereren naar de **Operations** en / of **Structuring** en / of **Strategy** van een onderneming.
- Zowel de toegevoegde waarde (**Valuation**), mogelijkheden (**Functions**) en de wijze waarop iets kan (**Construction**) kan worden beschreven in een principe. De aandacht wordt vooral gevestigd op de constructie en er wordt weinig aandacht besteed aan de toegevoegde waarde.
- Een principe is op conceptueel of logisch niveau beschreven.
- Principes hebben voornamelijk betrekking op een informatie systeem.
- Principes kunnen op verschillende domeinen van toepassing zijn, waaronder alle benoemde niveaus bij **Organisational range**.
- Een principe beschrijft en / of **Behaviour** (wat gebeurt er), en / of **Passive Structure** (waar leidt dit toe/wat is het gevolg/resultaat) en / of **Active Structure** (wat of wie doet het).
- Een principe heeft betrekking op een **Operational system**.
- Een principe is in natuurlijke taal beschreven (**Informal**).

- Een principe is **Definite**. Een principe geeft een richting aan en beperkt de handelingsvrijheid.
- De actors / objecten die zich dienen te conformeren aan een principe of waarbinnen het principe observeerbaar dient te zijn: o.a. mens, klant, informatiesysteem, systeemonderdeel, onderneming, deelonderneming.
- Principes zijn **Regulating propositions**. Ze kunnen zowel **risk-based** als **refinement-based** zijn. Principes zijn afgeleid van de visie, strategie, omgeving en businessrequirements van een onderneming.
- De impact beschrijft de bedrijfs- voordelen en nadelen, implementatie impact en issues en conflicten van het hanteren van een principe.
- Voor het hanteren van een principe is de deployment strategie **Communicate** van toepassing. Men communiceert de principes richting degene die zich er aan dienen te conformeren of er voor moeten zorgen dat een object zich zal conformeren aan het principe.

De eigenschappen van de architectuurprincipes van Stater NV:

De architectuurprincipes van Stater NV beschrijven vooral hoe men wenst dat de ICT-oplossing gebouwd wordt en welke functies er in moeten zitten. Hier mag bij voorkeur niet van worden afgeweken. De architectuurprincipes beschrijven dus vooral de functie en constructie van de te bouwen ICT-oplossing. De architectuurprincipes zijn op logisch of conceptueel niveau beschreven in natuurlijke taal en zijn richtinggevend. De architectuurprincipes hebben vooral betrekking op de ICT-oplossing zelf en weinig op de context en omgeving waarin de ICT-oplossing zal gaan worden ingezet. De impact, motivatie, issues en conflicten zijn niet opgenomen in de beschrijvingen van de architectuurprincipes.

7. Verhouding EA methoden en Proposition Framework v0.9

In het classificatie experiment is gekeken hoe voorbeelden uit de praktijk zich verhouden tot het raamwerk. Dit experiment alleen is onvoldoende om een antwoord te kunnen geven op vragen als: wat voor uitspraken (proposities) zijn nu echt architectuurprincipes? Wat is het verschil tussen de verschillende soorten proposities? Bovendien kan de vraag gesteld worden of de proposities uit de verschillende casussen überhaupt wel het juiste label hebben opgespeld gekregen.

Vanuit deze gedachtegang is het volgende experiment bedacht en gedeeltelijk uitgevoerd:

De diverse EA methoden hebben allen een eigen visie op hetgeen wat nu een architectuurprincipe is. Deze theoretische raamwerken en methoden beschrijven wat een architectuurprincipe is en welke eigenschappen deze heeft. Door te kijken hoe deze beschrijvingen van de diverse methoden zich verhouden tot het raamwerk kan men ook op theoretische gronden een antwoord geven op de vragen die de aanleiding hebben gevormd voor het ontwikkelen van het raamwerk.

Het experiment is als volgt te beschrijven:

1. Verzamel per methode de bijbehorende theorie over (architectuur)principes.
2. Noteer vervolgens per attribuuttype wat de methode feitelijk beschrijft m.b.t. het betreffende attribuuttype.
3. Bepaal op basis van deze beschrijving welke classificatie van toepassing is.
4. Vergelijk de resultaten van de methoden en beschrijf welke overeenkomsten en verschillen er zijn.

Door de omvang van deze afstudeerscriptie is dit experiment niet in zijn geheel uitgevoerd. Er is één methode behandeld om te achterhalen of dit experiment mogelijk is en wat voor resultaat dit oplevert. In vervolg onderzoek kunnen de overige methoden (denk aan DYA, TOGAF etc.) behandeld worden.

De methode die voor dit experiment is uitgekozen betreft Dragon1. Deze methode wordt in het vervolg van deze afstudeerscriptie gebruikt en is daarom het meest relevant voor dit experiment.

In dit hoofdstuk wordt antwoord gegeven op de volgende deelvraag:

Welke eigenschappen hebben architectuurprincipes volgens de theorie?

- Welke eigenschappen zijn toe te kennen aan een architectuurprincipe op basis van de theorie van architectuurmethoden- en raamwerken?

7.1 Experiment

De onderstaande uiteenzetting is van toepassing op een principe volgens de definitie van Dragon1. Per attribuuttype zal een korte uiteenzetting worden gegeven van wat de Dragon1 zegt m.b.t. dit attribuuttype en welke classifieer daarom van toepassing is.

Validity attribute-types

Intricity

Dragon1 zegt het volgende m.b.t. dit attribuuttype:

- Een principe is een constructieve waarheid of een aan zekerheid grenzende waarschijnlijkheid (probability > 0.8) die altijd waar is binnen een bepaalde context, periode, situatie en abstractieniveau van beschouwing.
- Een principe duidt de fundamentele werking aan of een werkend beginsel van een concept, fenomeen of systeem.

- Een principe is zoals het is.

Op basis van deze uitspraken kan geconcludeerd worden dat volgens Dragon1 een principe **Intrinsic** is.

Probability

Dragon1 zegt het volgende m.b.t. dit attribuuttype:

- Een principe is een constructieve waarheid of een aan zekerheid grenzende waarschijnlijkheid (probability > 0.8) die altijd waar is binnen een bepaalde context, periode, situatie en abstractieniveau van beschouwing.

Volgens Dragon1 heeft een principe een **probability** van groter dan 80%. Er wordt echter gezegd dat binnen een bepaalde context, periode, situatie en abstractieniveau van beschouwing een principe altijd waar is. Op basis van deze uitspraak kan geconcludeerd worden dat volgens Dragon1 een principe altijd observeerbaar / van toepassing is. De classifieer **always** past daarom het beste bij een principe volgens Dragon1.

Obedience level

In lijn met wat er beschreven is bij het attribuuttype **Probability** is volgens Dragon1 het **Obedience level** van een principe **Strict**.

Object attribute-types

Time

Dragon1 zegt het volgende m.b.t. dit attribuuttype:

- Een principe is een tijdsgebonden wetmatigheid.

Binnen Dragon1 komt het attribuuttype **Time** onder meer terug in het attribuut Beschouwingssituatie. Hierin onderscheid Dragon1 de volgende situatietypen:

- **AS-IS principles**
Een AS-IS principe is een principe dat zich in de huidige situatie (heden - 1 jaar) daadwerkelijk voordoet, waarmee men er verstandig aan doet er rekening mee te houden.
- **Plateau n - principles**
Een plateau n principe is een principe dat zich tijdelijk voordoet (jaar X).
- **TO-BE principles**
Een TO-BE principe is een principe dat zich in de toekomstige situatie (3-5 jaar) zal gaan voordoen, realistisch en haalbaar is, waarmee men er verstandig aan doet er in de toekomst rekening mee te houden.
- **Envision principles**
Een Envision principe is een principe dat zich in de verre toekomst zal gaan voordoen, maar voor de AS-IS, Plateau1 en TO-BE situatie onrealistisch is.

Een principe kan dus op verschillende tijdspanne van toepassing zijn.

Control abstraction

Volgens Dragon1 dient een principe op elke niveau van controle abstractie van toepassing te zijn. Een principe heeft dus betrekking op **Operations**, **Structuring** en **Strategy**.

Construction abstraction

Volgens Dragon1 dient (of kan) een principe op elke niveau van constructie abstractie van toepassing te zijn. Dit is ondermeer af te leiden van de structuur en inhoud van een principe die wordt gehanteerd door Dragon1:

Als een uitspraak een principe moet worden formuleer dan wat er (steeds) gebeurt, wat dat als gevolg of resultaat heeft en wat dat dan weer mogelijk of onmogelijk maakt.

'door ..[een bepaalde technische oplossing of actie].. zal, of wordt ..[een bepaalde gewenste reactie of opgelost probleem].. met als gevolg dat ..[een bepaald opgeleverd resultaat]..'

Zowel de toegevoegde waarde (**Valuation**), mogelijkheden (**Functions**) als de wijze waarop (**Construction**) dienen te worden beschreven in een principe volgens Dragon1.

Physical abstraction

Dragon1 onderkent de drie niveaus (fysiek, logisch, conceptueel) van fysiek abstractie. Een principe dient volgens Dragon1 zoveel mogelijk op conceptueel niveau beschreven te worden.

Enablement abstraction

Dragon 1 onderscheid binnen dit attribuuttype de volgende classifiers onder de noemer bedrijfssysteem - functie:

- Inkoop
- Verkoop
- Marketingcommunicatie
- Productie
- Financiën
- ICT (IV, ICT-Infra)

Een principe heeft dus een bepaalde bedrijfssysteem – functie volgens Dragon1. Een principe kan op één of meerdere niveaus van toepassing zijn.

Organisational range

Dragon1 zegt het volgende m.b.t. dit attribuuttype:

“Principes, of beter gezegd concepten, worden op verschillende beschouwingsniveaus gehanteerd. Op ondernemingsniveau of bedrijfsniveau kan een concept als wenselijk en haalbaar worden gezien om te gaan hanteren, terwijl op bedrijfsfunctie- of bedrijfsprocesniveau men al snel ziet dat een bepaald concept niet te hanteren of te implementeren zal zijn.”

Dragon 1 onderscheid binnen dit attribuuttype de volgende classifiers onder de noemer organisatiesysteem - beschouwingsniveau:

- Onderneming
- Bedrijf
- Bedrijfsfunctie
- Bedrijfsproces
- Bedrijfstaak
- Handeling

Hieruit blijkt dat volgens Dragon1 een principe op alle gespecificeerde niveaus van toepassing kan zijn.

Aspects of dynamic systems

In het Dragon1 stappenplan voor het formuleren van principes staat het volgende geschreven:

“Als het statement een principe moet worden formuleer dan wat er (steeds) gebeurt, wat dat als gevolg of resultaat heeft en wat dat dan weer mogelijk of onmogelijk maakt.”

Volgens Dragon1 beschrijft een principe dus zowel het **Behaviour** (wat gebeurt er), **Passive Structure** (waar leidt dit toe/wat is het gevolg/resultaat) als **Active Structure** (wat of wie doet het).

Systemic order

Dragon1 maakt binnen dit attribuuttype onderscheid tussen de volgende soorten type principes:

- Planningsprincipe
- Ontwikkel(bouw/ontwerp)principe
- Transformatie/veranderprincipe
- Beheer/instandhoudingprincipe

Hieruit blijkt dat volgens Dragon1 een principe betrekking kan hebben op een **Operational system** of **Transformation system**.

Form attribute-types

Level of precision

Dragon1 zegt o.a. het volgende m.b.t. dit attribuuttype:

- Maak het RAW-statement SMART als het een regel of principe is.
- Als het statement een principe moet worden formuleer dan wat er (steeds) gebeurt, wat dat als gevolg of resultaat heeft en wat dat dan weer mogelijk of onmogelijk maakt.

Binnen Dragon1 wordt niets gezegd over het formeel beschrijven van principes. Principes kunnen worden beschreven door natuurlijke taal of door middel van visualisaties. Dragon1 hanteert hiervoor een structuur die de syntactische variatie beperkt en zorgt voor een optimaal niveau van precisie. Principes zouden volgens Dragon1 dus als **Semi-formal** geclassificeerd worden.

Level of actionability

Volgens Dragon1 moet een principe **Actionable** zijn. Een principe heeft volgens Dragon1 een breed scala aan attributen die beschreven moeten worden. De combinatie van de invulling aan deze attribuuttypen zorgt er voor dat direct bepaald kan worden of het principe gehanteerd wordt en een situatie zich conformeert aan een bepaald principe.

Context attribute-types

Motivation

Van de drie soorten proposities die in het raamwerk genoemd worden bij dit attribuuttype worden **Intrinsic propositions** en **Regulating propositions** erkend door Dragon1. Bij **Intrinsic propositions** vormt, zoals ook aangegeven wordt in het raamwerk, een bewijs dat het principe werkt en zorgt voor het gewenste resultaat de motivatie voor het hanteren van een principe. **Regulating propositions** kunnen zowel **risk-based** als **refinement-based** zijn. Dragon1 ziet strategische uitgangspunten als hogere orde principes. Hiervan zijn principes dus afgeleid.

Impact

De volgende hypothese uit het raamwerk wordt onderkend door Dragon1:

“The impact of a proposition primarily makes sense for regulating and guiding propositions. In the case of inherent propositions, however, one may chose to discuss/illustrate the workings of the underlying mechanism as its impact.”

De impact van een principe komt bij Dragon1 vooral aanbod in de volgende attributen van een principe:

Bedrijfs- voordelen /nadelen, Implementatie impact en Issues en Conflicten

Deployment

De drie deployment strategieën die genoemd worden in het raamwerk kunnen allen van toepassing zijn op een principe volgens Dragon1.

Actor attribute-types

Een actor kan een mens, systeem, systeemonderdeel, organisatie, onderneming, instelling, organisatiedomein, project, proces, product etc. zijn.

7.2 Bevindingen

Het onderstaande overzicht geeft een samenvatting van de analyse die is uitgevoerd in hoofdstuk 7.1. Het beschrijft in een kort overzicht welke eigenschappen een principe heeft volgens Dragon1.

Validity	
Intricity	Intrinsic
Probability	Always
Obedience level	Strict
Object	
Time	Een principe is een tijdsgebonden wetmatigheid. Een principe kan op verschillende tijdspanne van toepassing zijn.
Control abstraction	Operations, Structuring en Strategy
Construction abstraction	Valuation, Function en Construction
Physical abstraction	Conceptual
Enablement abstraction	Een principe kan op elk niveau van toepassing zijn.
Organisational range	Een principe kan op elk niveau van toepassing zijn.
Aspects of dynamic systems	Een principe heeft betrekking op Behaviour, Passive structure en Active structure.
Systemic order	Een principe kan zowel betrekking hebben op een Operational system als een Transformational system.
Form	
Level of precision	Semi-formal
Level of actionability	Actionable
Actors	
Actors	-
Context	
Motivation	Bij Intrinsic propositions vormt een bewijs dat het principe werkt en zorgt voor het gewenste resultaat de motivatie voor het hanteren van een principe. Regulating propostions kunnen zowel risk-based als refinement-based zijn. Dragon1 ziet strategische uitgangspunten als hogere orde principes. Hiervan zijn principes afgeleid.
Impact	De impact beschrijft de Bedrijfs- voordelen /nadelen, Implementatie impact en Issues en Conflicten van een principe.
Deployment	Zowel de strategieën Communicate, Construct en Enforce kunnen van toepassing zijn.

De resultaten van dit experiment geven meer inzicht in welke attributen nu daadwerkelijk iets zeggen over wat een architectuurprincipe is. Dit zal duidelijk worden gemaakt door middel van een voorbeeld. Neem het attribuut **Organisational Range**. Volgens Dragon1 kan elke classifieer die bij dit attribuuttype hoort toegekend worden aan een architectuurprincipe. Anders gezegd; een architectuurprincipe kan betrekking hebben op één of meerdere organisatorische niveaus. Dit wordt ook door andere EA methoden onderkend. Uit de resultaten

van de experimenten waarin 'architectuurprincipes' uit de praktijk zijn geïdentificeerd zou kunnen worden afgeleid dat architectuurprincipes bijna alleen op enterprise niveau betrekking hebben. Dit wil echter niet zeggen dat architectuurprincipes ook daadwerkelijk alleen op enterprise niveau betrekking horen te hebben. Dit geldt voor meerdere attributen uit het raamwerk waaronder alle object attributetypes.

7.3 Conclusie

De termen principe en architectuurprincipe zijn door verschillende methoden gedefinieerd en het vergelijken van deze definities geeft meer inzicht in wat een architectuurprincipe in essentie is dan te kijken naar praktijkvoorbeelden. Deze methoden hebben de term (architectuur)principe geïntroduceerd en hebben hier een definitie en functie aan gegeven. Zij bepalen als het ware wat een architectuurprincipe is en de voorbeelden uit de praktijk zijn op basis daarvan opgesteld. Het analyseren van praktijkvoorbeelden geeft daardoor echter meer inzicht in het toepassen en het formuleren van architectuurprincipes. Dat de praktijkvoorbeelden zo verschillend zijn komt doordat de definitie en functie en de wijze waarop een principe geformuleerd dient te worden bij veel methoden verschillend is en zeer globaal beschreven zijn. Er wordt veel vrijheid overgelaten aan degene die de principes gaat formuleren. De kwaliteit van het geformuleerde principe is dan ook vaak afhankelijk van de kennis en kunde van degene die ze formuleert.

Door dit experiment ook uit te voeren voor de overige EA methoden kan gekeken worden welke eigenschappen van een architectuurprincipe nu daadwerkelijk door iedereen onderkend worden en waar nog over gediscussieerd wordt. Naar de eigenschappen waarover gediscussieerd wordt kan vervolgens verder onderzoek verricht worden, zodat het architectuurprincipe een duidelijke plek, definitie en functie krijgt waarmee het een toegevoegde waarde is in de verzameling van propositie soorten.

De combinatie van het raamwerk te toetsen aan theorie en praktijk geeft een duidelijk inzicht in de toepasbaarheid en compleetheid van het raamwerk en in wat architectuurprincipes nu zijn en in welke vorm ze terug te vinden zijn in de praktijk. Het raamwerk kan completer gemaakt worden door te achterhalen welke theorie van de diverse methoden over architectuurprincipes nog niet terug te vinden is in de vorm van een attribuuttype in het raamwerk.

Op basis van het uitgevoerde experiment worden door Dragon1 de volgende eigenschappen toegewezen aan een principe:

- Een principe is in essentie en bij voorkeur **Intrinsic**. Wanneer een principe **Desired** is dient er een handhavingmechanisme actief te zijn dat er voor zorgt dat het voordeel van het hanteren van het principe ook daadwerkelijk behaald wordt.
- Een principe is in essentie en bij voorkeur altijd observeerbaar / actief in een bepaalde situatie. De **Probability** dient in ieder geval hoger te liggen dan 80%.
- Het **Obedience Level** van een principe is in essentie **Strict**. Een principe kan ook **Overridable** zijn. Het afwijken van een principe is echter niet gewenst.
- Een principe is een tijdsgebonden wetmatigheid. Een principe kan op verschillende tijdsperiodes van toepassing zijn.
- Een principe refereert zowel naar de **Operations, Structuring** en **Strategy** van een onderneming. Indien één van deze gebieden niet belicht wordt betreft het geen principe.
- Zowel de toegevoegde waarde (**Valuation**), mogelijkheden (**Functions**) als de wijze waarop dit bewerkstelligd wordt (**Construction**) dienen te worden beschreven in een principe.
- Een principe is bij voorkeur op conceptueel niveau beschreven.
- Principes kunnen op elk niveau dat onderkend wordt bij **Enablement abstraction** ingezet worden (van toepassing zijn).

- Principles kunnen op verschillende domeinen van toepassing zijn, waaronder alle benoemde niveaus bij **Organisational range**.
- Een principe beschrijft zowel het **Behaviour** (wat gebeurt er), **Passive Structure** (waar leidt dit toe/wat is het gevolg/resultaat) als **Active Structure** (wat of wie doet het).
- Een principe kan zowel betrekking hebben op een **Operational system** als een **Transformational system**.
- Een principe is in natuurlijke taal beschreven en de syntactische variatie wordt ingeperkt door een vast stramien. Een principe is dus **Semi-formal** beschreven.
- Een principe is **Actionable**. De beschrijving van een principe moet er voor zorgen dat het direct duidelijk is of een situatie zich conformeert aan het principe.
- De actors / objecten die zich dienen te conformeren aan een principe of waarbinnen het principe observeerbaar dient te zijn: mens, systeem, systeemonderdeel, organisatie, onderneming, instelling, organisatiedomein, project, proces, product etc. Dit zijn slechts enkele voorbeelden.
- Bij **Intrinsic propositions** vormt een bewijs dat het principe werkt en zorgt voor het gewenste resultaat de motivatie voor het hanteren van een principe. **Regulating propositions** kunnen zowel **risk-based** als **refinement-based** zijn. Principles zijn afgeleid van strategische uitgangspunten en kwaliteitseisen die gezien worden als hogere orde principes.
- De impact beschrijft de bedrijfs- voordelen en nadelen, implementatie impact en issues en conflicten van het hanteren van een principe.
- Voor het hanteren van een principe kunnen alle genoemde deployment strategieën (**Communicate**, **Construct** en **Enforce**) gebruikt worden.

Wanneer is een propositie een architectuurprincipe volgens Dragon1:

Op basis van de benoemde eigenschappen van een architectuurprincipe in dit hoofdstuk, kan men een architectuurprincipe herkennen (zoals Dragon1 een principe definieert). Voldoet een propositie te weinig aan deze eigenschappen dan is het volgens Dragon1 geen principe maar een ander soort propositie (bijvoorbeeld een regel, richtlijn, voorschrift of uitgangspunt).

8. Evaluatie Proposition Framework

In dit hoofdstuk wordt het Proposition Framework v0.9 geëvalueerd. Als eerste zullen algemene bevindingen worden beschreven die op het gehele raamwerk van toepassing zijn. Vervolgens zullen specifieke bevindingen per attribuuttype worden besproken. Op basis van deze bevindingen zijn aanbevelingen gegeven om het raamwerk te verbeteren.

In dit hoofdstuk wordt antwoord gegeven op de volgende deelvragen:

In hoeverre is het Proposition Framework correct en bruikbaar in de praktijk?

- Welke problemen zijn er bij het toepassen van het Proposition Framework op een praktijkcasus?
- Welke aanbevelingen zijn er op het Proposition Framework naar aanleiding van deze problemen?

De dikgedrukte woorden zijn attribuutklassen, attribuuttypes of classifiers uit het Proposition Framework v0.9.

8.1 Algemene bevindingen

Domein soort

Het is niet duidelijk gespecificeerd of een domein een **Ordered domain**, **Elaboration domain** of **Finite domain** is. Bij de meeste attribuut-types is dit af te leiden maar bij de volgende attribuut-types is hier onduidelijkheid over:

- Control abstraction
- Construction abstraction
- Physical abstraction
- Aspects of dynamic systems

Deze onduidelijkheid wordt mede veroorzaakt doordat de classifiers in de vragende vorm beschreven staan. Hierdoor lijkt het alsof je antwoord moet geven op deze vragen. Het kan echter ook vanuit het perspectief bekeken worden dat het principe antwoord geeft op één van de drie vraagstellingen en aan de hand daarvan de classificatie bepaald dient te worden.

In dit onderzoek is gekozen voor de laatste interpretatiewijze. Bij de classificatie van de architectuurprincipes van deze casus is dat bij **Control abstraction**, **Construction abstraction** en **Aspects of dynamic systems** meerdere classificaties mogelijk (te beargumenteren) zijn. Bij **Physical abstraction** was het bij deze casus geen probleem om de juiste classificatie toe te wijzen.

Referenties

Waar attribuuttypes en hun classifiers gebaseerd zijn op referenties is het lastig om zonder deze referenties (artikelen) gelezen te hebben de verschillende waarden van de attribuuttypes en classifiers te begrijpen en te onderscheiden. Niet alle artikelen zijn als openbare bronnen beschikbaar waardoor het niet voor iedereen mogelijk is om optimaal gebruik te maken van het raamwerk.

Interpretatie

Er is veel ruimte voor interpretatie door de lezer / gebruiker van het raamwerk. Meerdere classificaties lijken daardoor te beargumenteren, waardoor vraagtekens ontstaan omtrent de juiste classificatie. Hierdoor is het moeilijk om de doelstellingen van het raamwerk te bereiken omdat er juist duidelijkheid gewenst wordt en het dit niet oplevert.

8.2 Specifieke bevindingen

Per domein zullen de bevindingen en problemen behandeld worden die zijn opgetreden en/of geconstateerd tijdens de classificatie van de architectuurprincipes.

Form attribute-types

Level of precision

Bij het attribuuttype **Level of precision** wordt onderscheid gemaakt tussen Informal, Semi-formal en Formal. De mate van precisie wordt gemeten aan de hand van het niveau van formaliteit, verwijzend naar mogelijke mathematische / automatische interpretatie en manipulatie. In de praktijk blijkt echter dat architectuurprincipes bijna alleen maar op informele wijze worden beschreven. De mate van precisie en detail waarmee een principe in natuurlijke taal (informeel) beschreven kan worden is echter groot. Met dit feit wordt in het raamwerk onvoldoende rekening gehouden.

Level of actionability

Bij dit attribuuttype is de stap tussen de niveaus **Definite** en **Specific** erg groot. Het **Level of actionability** van een principe is sterk afhankelijk van de mate van precisie naar mijn mening. Ook hier geldt weer dat er binnen het niveau **Definite** grote verschillen zijn in de mate waarin een principe 'actionable' is. Ook zijn de niveaus **Specific** en **Actionable** moeilijk toe te kennen door het ontbreken van een duidelijk meetinstrument bij de principes. Hierdoor is al snel het gros van de principes in de praktijk te classificeren als **Definite**. De oorzaak hiervan ligt naar alle waarschijnlijkheid dat veel EA methoden niet erkennen dat een principe meetbaar hoort te zijn, maar richtinggevend.

Validity attribute-types

Intricity

Er zijn geen duidelijke voorbeelden en er is geen duidelijke definitie van wanneer een architectuurprincipe als **Intrinsic** kan worden geclassificeerd.

Probability Level

De zin 'The Probability with which the property is (desired to be) observable in the enterprise' levert verwarring op. De gewenste waarschijnlijkheid dat het principe observeerbaar is komt vaak niet overeen met de werkelijkheid. De vraag die hier kan ontstaan is of de classificatie gebaseerd moet zijn op de gewenste observeerbaarheid of wat er in de werkelijkheid geobserveerd wordt. Hier moet een duidelijke keuze worden gemaakt.

Obedience Level

In het raamwerk wordt de mogelijkheid geschetst dat het attribuut **Probability** misschien onnodig is door de mogelijke connectie tussen **Probability** en **Obedience Level**. Uit deze casus is gebleken dat deze relatie aanwezig is. Mijn inziens voegt het **Probability** attribuut in de huidige vorm weinig toe en kan dit afgeleid worden van het **Obedience Level**.

Zie onderstaand overzicht als mogelijke suggestie:

Obedience Level		Probability
Strict	→	Always

Wanneer er geen uitzonderingen mogelijk/gewenst zijn dan zal de propositie altijd waarneembaar (horen te) zijn.

Overridable	→	Usually
-------------	---	---------

Wanneer er wel uitzonderingen mogelijk zijn dan zou de propositie vaak waarneembaar (horen te) zijn.

Guiding → Sometimes

Wanneer een propositie richtinggevend is zou deze soms waarneembaar (horen te) zijn. Hier zou echter **sometimes** niet als minder dan 50% gedefinieerd moeten worden omdat een richtinggevende propositie ook wel vaker als 50% zou kunnen worden waargenomen.

Object attribute-types

De meeste vraagtekens treden op bij de **Object attribute-types**. Vooral de attribuut-types **Control abstraction**, **Construction abstraction** en **Aspects of dynamic systems** zijn lastig te bepalen voor de principes. Voor de attribuut-types **Control abstraction** en **Construction abstraction** wordt verwezen naar andere artikelen die hier meer inzicht in kunnen geven. Voor **Aspects of dynamic systems** is dit niet het geval. Hier zou meer uitleg en/of voorbeelden gewenst zijn. Vanuit dit oogpunt zou er misschien meer aandacht moeten worden besteed aan de gebruikswijze van het document waarin het raamwerk wordt beschreven.

Contextual attribute-types

Motivation

Bij een aantal principes is het lastig om een beargumenteerde keuze te maken tussen **Risk-based** en **Refinement-based**. Deze principes laten zich niet in één hokje plaatsen. Vaak zijn de principes uit deze casus gedefinieerd naar aanleiding van een hoger doel dat men wil bereiken. Het hogere doel is hier dan in mijn ogen de motivatie. De vraag hierbij is of het hogere doel (vaak een strategisch uitgangspunt of doelstelling) hier dan ook als propositie gezien moet worden waardoor **Refinement-based** in aanmerking komt als classificatie of moet hier een extra mogelijkheid worden toegevoegd.

Actors attribute-types

Voor de Actor dimension staat in het raamwerk de volgende beschrijving: *attribute-types dealing with the identification of those actors who are expected to (be observed to) respect the propositions*. Indien niet duidelijk gespecificeerd is op welke actoren de propositie van toepassing is / betrekking heeft welke actors moeten hier dan worden beschreven? Alle actors die genoemd worden in de propositie (zowel menselijke actors als machines of IT technologie)?

8.3 Conclusies & Aanbevelingen

Per bevinding zal worden aangegeven welke conclusies getrokken kunnen worden op basis van deze bevindingen en welke aanbevelingen op basis hiervan kunnen worden gedaan.

Domeinsoort

Uit dit onderzoek is gebleken dat het domein van de volgende attribuuttypes een **Elaboration domain** betreft:

- Control abstraction
- Construction abstraction
- Physical abstraction
- Aspects of dynamic systems

Bij **Physical abstraction** moet echter toegevoegd worden dat het ook mogelijk is dat er een keuze uit één van de classifiers van toepassing is. Het kan namelijk zo zijn dat een principe op één of meerdere niveau's van **Physical abstraction** is beschreven. Indien er maar één niveau is beschreven dan kan er een keuze worden gemaakt. Indien het principe op meerdere niveaus beschreven is kunnen alle drie de classifiers van toepassing zijn.

Referenties & Interpretatie

De attribuuttypes en classifiers moeten van een uitgebreidere omschrijving worden voorzien en de referenties

moeten openbaar beschikbaar zijn. Vanuit dit oogpunt zou er meer aandacht moeten worden besteed aan de gebruikswijze van het document waarin het raamwerk wordt beschreven.

Level of precision

Op basis van de bevindingen is het gewenst dat **Level of precision** op een andere manier gemeten moet worden. Een principe kan beschreven worden in natuurlijke taal én / of met een visualisatie én / of op een meer formele wijze in de vorm van een formule of schematechniek. In deze gevallen is het mogelijk dat een combinatie van wijzen van formuleren zorgt voor een bepaalde mate van precisie. Dit zijn factoren waarmee rekening moet worden gehouden om een vorm te vinden waarmee **Level of precision** beter gemeten en geëvalueerd kan worden.

Intricity

Er is naar mijn mening een tweesplitsing noodzakelijk binnen het niveau **Intrinsic**. Er zijn namelijk principes zoals beschreven in het raamwerk bij de omschrijving van **Intrinsic**, dus principes die altijd van toepassing zijn en altijd geldig zijn. Je hebt echter ook principes die op dit moment voor waar worden aangenomen, maar die niet zo 'waar' zijn als beschreven wordt bij **Intrinsic**. Dit zijn principes die op dit moment voor waarheid worden gezien of mee omgegaan wordt alsof het een waarheid betreft, maar in de toekomst kunnen veranderen en / of onderuit gehaald kunnen worden. Deze categorie principes liggen naar mijn mening tussen de niveaus **Intrinsic** en **Desired** in. Een extra niveau is daarom gewenst.

Probability & Obedience level

Op basis van de resultaten en bevindingen van dit onderzoek kan geconcludeerd worden dat deze attributen in de huidige vorm te veel overlap hebben en de classifiers onderling directe relaties hebben. Dit wordt mede veroorzaakt omdat er in het raamwerk niet duidelijk gemaakt wordt waarop de classificatie gebaseerd dient te zijn. De vraag hierbij is of de classificatie gebaseerd moet zijn op wat gewenst is (hoe het beschreven staat in een document) of wat daadwerkelijk observeerbaar is in de praktijk. De volgende relaties tussen de attributen is van toepassing op basis van de resultaten uit dit onderzoek:

- Wanneer er geen uitzonderingen mogelijk/gewenst zijn dan zal de propositie altijd waarneembaar (horen te) zijn.
- Wanneer er wel uitzonderingen mogelijk zijn dan zou de propositie vaak waarneembaar (horen te) zijn.
- Wanneer een propositie richtinggevend is zou deze soms waarneembaar (horen te) zijn. Hier zou echter **sometimes** niet als minder dan 50% gedefinieerd moeten worden omdat een richtinggevende propositie ook wel vaker als 50% zou kunnen worden waargenomen.

Proposition Framework

Het Proposition Framework v0.9 is in de huidige vorm een handig overzicht van mogelijke eigenschappen die architectuurprincipes kunnen hebben. Het kan als checklist dienen bij het opstellen van architectuurprincipes en hulp bieden bij het bepalen van de functie van een architectuurprincipe. Hiermee wordt bedoeld dat een organisatie kan bepalen hoe men het architectuurprincipe als instrument wil gaan inzetten en welke eigenschappen het dan moet hebben.

Als classificatiemiddel is het echter nog lastig te gebruiken. Dit komt vooral doordat de classificiers zeer globaal zijn beschreven en daardoor weinig zeggen over wanneer een eigenschap mag worden toegekend aan een architectuurprincipe. Hierdoor is het lastig toe te passen in de praktijk. Hier is veel kennis op het gebied van enterprise architectuur en architectuurprincipes voor nodig. Het raamwerk laat dus nog te veel ruimte voor interpretatie over, waardoor er onzekerheid ontstaat over de juiste classificatie. Hierdoor zijn de resultaten uit de verschillende casussen lastig te vergelijken en is het moeilijk om antwoorden te vinden op de vragen die ten grondslag liggen aan het ontwikkelen van het raamwerk. Het is dus belangrijk om de attribuu types en de classificiers nog scherper te formuleren om verkeerde interpretatie uit te sluiten. Vooral de object attributetypes zijn lastig te classificeren.

Meer inzicht verkrijgen in de eigenschappen van de verschillende propositie soorten wordt in mijn ogen met de huidige versie van het raamwerk en het uitgevoerde experiment nog niet bereikt. Dit komt omdat het raamwerk nog onvoldoende is uitgekristalliseerd en van te voren niet beoordeeld is of de architectuurprincipes uit de casussen wel van voldoende kwaliteit zijn en dus als goed voorbeeld gezien mogen worden.

Uiteindelijk kan het raamwerk wel laten zien hoe verschillende soorten proposities er uit zien (welke eigenschappen ze hebben), maar blijft het lastig om vervolgens te bepalen wat nu een architectuurprincipe, principe, regel etc. genoemd mag worden en welke eigenschappen nu gewenst zijn.

9. Aanbevelingen voor Stater NV

Naar aanleiding van de classificatie en gesprekken met de contactpersonen van Stater is gebleken dat de architectuurprincipes in de huidige vorm veel onduidelijkheden met zich mee brengen. Op basis van deze constatering is een aanzet gedaan om aan te geven op welke punten deze architectuurprincipes verbeterd kunnen worden. Aan de hand van het raamwerk zal aangegeven worden waar en welke verbeteringen kunnen plaats vinden. Het raamwerk dient in eerste instantie als classificatiemiddel maar geeft ook inzicht aan welke aspecten aandacht besteed kan worden.

In hoeverre is het Proposition Framework correct en bruikbaar in de praktijk?

- Hoe kan Stater NV het Proposition Framework gebruiken om betere architectuurprincipes te formuleren?

Form attribute-types

Level of precision

Uit de classificatie is gebleken dat de architectuurprincipes informeel beschreven zijn. De wijze waarop de architectuurprincipes zijn beschreven is echter voor verbetering vatbaar. Er is op dit moment geen duidelijke notatiewijze gebruikt waarmee men de architectuurprincipes beschreven heeft. Er is dus weinig consistentie te vinden tussen de beschrijvingen van de verschillende architectuurprincipes. Hierdoor ontbreekt bij veel architectuurprincipes informatie of wordt juist te specifieke informatie toegevoegd die in de omschrijving van het architectuurprincipe voor verwarring zorgen.

Het belang van het gebruiken van een eenduidige notatiewijze is dat iedereen ziet om wat voor soort uitspraak het gaat en dat er duidelijke afspraken zijn over wat voor informatie er wel of niet in de omschrijving van dit soort uitspraken moet worden vermeld. Door dit te doen krijg je een verzameling van uitspraken van een zelfde niveau waardoor ze gemakkelijker te plaatsen zijn binnen de overige soorten uitspraken die men onderkend binnen een bedrijf.

Level of actionability

De architectuurprincipes uit de casus beschrijven veelal wat men wenst terug te zien in de uiteindelijke ICT-oplossing. De huidige architectuurprincipes stellen daardoor een kader waaraan de ICT-oplossing moet gaan voldoen. De wijze waarop de architectuurprincipes nu zijn geformuleerd laat echter veel ruimte over voor interpretatie door de lezer / uitvoerende. Het is daardoor moeilijk om op basis van deze architectuurprincipes te beslissen of keuzes die worden gemaakt tijdens het ontwikkeltraject wel conform de architectuurprincipes zijn. Doordat er geen duidelijke meetinstrumenten zijn geformuleerd is het ook lastig om in een later stadium te bepalen in hoeverre de ICT-oplossing zich conformeert aan de architectuur en architectuurprincipes die zijn opgesteld. Men moet goed nadenken wat men wil bereiken met de architectuurprincipes. Willen we ze gaan gebruiken als richtlijn of moeten ze leidend zijn. En moeten we het dan wel architectuurprincipes noemen? Als men met zekerheid wil stellen dat de ICT-oplossing zich zal conformeren aan de architectuurprincipes zal hier meer aandacht aan moeten worden besteed.

Object attribute-types

Time

De huidige architectuurprincipes zijn niet tijdgebonden. Door de huidige formulering en het doel van de architectuurprincipes is het ook niet mogelijk om er een specifieke tijdsspanne aan te koppelen.

Control abstraction

Een architectuurprincipe zou moeten verwijzen naar alle niveaus binnen 'Control abstraction'. Een architectuurprincipe moet een deel van een strategisch doel of uitgangspunt verwezenlijken en zou afgeleid moeten zijn van een strategische doel of uitgangspunt. Daarnaast zou het architectuurprincipe moeten verwijzen naar de structuur die men aan (wil) brengen door het hanteren van het architectuurprincipe en wat

dit betekend voor het operationeel functioneren van het systeem of 'enterprise'. Door hier meer aandacht aan te besteden bij het formuleren van een architectuurprincipe, krijgt men een beter inzicht en overzicht van de veranderingen die plaats zullen gaan vinden en de impact die het hanteren van een bepaald architectuurprincipe heeft binnen de organisatie.

Construction abstraction

Hier geldt hetzelfde als bij 'Control abstraction'. Door in de beschrijving van het architectuurprincipe aandacht te besteden aan alle drie de niveaus, krijgt men een beter inzicht welke behoeften van het bedrijf worden afgevangen door middel van het hanteren van het architectuurprincipe en op welke manier dit gebeurt.

Physical abstraction

De architectuurprincipes zijn op conceptueel niveau beschreven. Het is echter interessant om ook aandacht te besteden aan wat voor invloed of mogelijkheden / onmogelijkheden dit meebrengt op logisch en fysiek niveau.

Aspects of dynamic systems

Ook bij dit attribuuttype is het van belang om invulling te geven aan alle drie de niveaus aangezien het verstandig is om een architectuurprincipe te laten beschrijven wat er gebeurt (een werkwijze), waar dit toe leid en wie of wat het doet.

Enablement abstraction, Organisational range en Systemic Order

Het is belangrijk om in ieder geval na te denken op welk gebied een architectuurprincipe ondersteuning biedt en op welke organisatorisch niveau het architectuurprincipe betrekking heeft. Het nadenken over deze vragen geeft een goed beeld van de impact en reikwijdte van een bepaald architectuurprincipe.

Door na te denken over de invulling van de verschillende attribuuttypes verkrijgt men een beter inzicht in de scope en het nut van een architectuurprincipe. Ook de impact van het architectuurprincipe is zo veel beter te bepalen. Op basis van deze factoren kan men beter beslissen of men het architectuurprincipe wil hanteren en op welke wijze men het architectuurprincipe zal handhaven.

Validity attribute-types

Intricity

De huidige architectuurprincipes gaan vooral over wat men wil. In combinatie met het ontbreken van een duidelijk handhavingmechanisme zorgt dit voor een groot risico. Er kan nooit met zekerheid gesteld worden dat men zich zal gaan conformeren aan de architectuurprincipes. Het is daarom gevaarlijk om alleen architectuurprincipes te hanteren waarvan niet is aangetoond dat ze ook daadwerkelijk zullen werken. Het is aan te raden om zoveel mogelijk architectuurprincipes te gebruiken waaraan bewezen praktijkervaring of wetenschappelijk bewijs ten grondslag ligt.

Probability

Een eigenschap van een architectuurprincipe is dat het een uitspraak is die (altijd) geldig is en daarom is er de wens dat deze ook altijd observeerbaar is binnen een bedrijf. Men moet er naar streven dat een principe altijd waarneembaar is binnen het bedrijf. Dit kan bewerkstelligd worden door alleen bewezen architectuurprincipes te hanteren of een effectieve deployment strategie te formuleren en te hanteren.

Obedience level

Door de bovengenoemde eigenschap van een architectuurprincipe is het gewenst dat er zo weinig mogelijk afgeweken wordt van het architectuurprincipe. Uitzonderingen zijn mogelijk maar deze moeten duidelijk worden gedocumenteerd en verantwoord aangezien je afwijkt van een bewezen aanpak.

Context attribute-types

Motivation

De motivatie van de architectuurprincipes ontbreekt in de omschrijving van het architectuurprincipe zelf en ook in het document waarin de architectuurprincipes zijn beschreven. Het is aan te raden om deze vast te leggen zodat men altijd voor ogen heeft wat de achterliggende reden is voor het hanteren van een architectuurprincipe. Dit kan, zoals het raamwerk beschrijft, het uitsluiten van een risico zijn, het verwezenlijken van een hoger doel of ander principe, of een bewijs wat laat zien dat het verstandig / noodzakelijk is om een principe te hanteren.

Impact

Het beschrijven van de impact is noodzakelijk om beslissingen te kunnen nemen over het willen onderkennen / hanteren van een architectuurprincipe. Men moet een inzicht hebben in de mogelijke complicaties of de richting waarin men zal inslaan bij het onderkennen van het architectuurprincipe. Ook een kosten / baten analyse zou hier op zijn plaats zijn. Men krijgt zo een beeld van de kosten die verbonden zijn aan de impact van een architectuurprincipe, waarop men kan beslissen of de kosten opwegen tegen de baten.

Deployment

Doordat in de beschrijving van de architectuurprincipes veel informatie ontbreekt en de architectuurprincipes vooral beschrijft wat men wil, is het belangrijk dat er een deployment strategie aanwezig is die werkt als handhavingmechanisme. Hier is binnen deze casus weinig aandacht aan besteed.

Actors attribute-types

De huidige architectuurprincipes zijn geformuleerd in wensen en eisen waaraan de ICT-oplossing moet voldoen. Het is aan te raden om deze te vertalen naar regels en richtlijnen waar het ontwikkelteam van de ICT-oplossing zich aan dient te conformeren.

De architectuurprincipes uit de besproken casus bevestigen het beeld dat geschetst wordt in de inleiding en probleemstelling. Over veel aspecten is onvoldoende nagedacht en / of niet gedocumenteerd en men heeft geen goed beeld van hoe en wat men wil bereiken met de opgestelde architectuurprincipes. De 'architectuurprincipes', zoals de proposities uit de casus worden genoemd, zijn mijn inziens 'gelabeld' als architectuurprincipe, maar zijn eerder functionele eisen en bouwrichtlijnen. In het volgende hoofdstuk wordt weergegeven welke problemen en onduidelijkheden er in de praktijk ontstaan door onvolledig en slecht geformuleerde architectuurprincipes.

10. Analyse architectuurproces Stater NV

Naast de literatuurstudie (zowel uitgevoerd in fase 1 als fase 2 van deze afstudeerscriptie) en het bekijken en analyseren van architectuurprincipes (fase 1) is het van belang om te achterhalen welke problemen architectuurprincipes in de praktijk met zich meebrengen en hoe deze problemen ontstaan. Om hier een antwoord op te vinden is het architectuurproces van Stater NV nader onderzocht.

Om meer inzicht te verkrijgen in de problematiek omtrent het formuleren van architectuurprincipes en om te toetsen of een gestructureerde methode deze problematiek kan oplossen zijn een aantal architectuurprincipes uit de Stater casus geherformuleerd. Dit is gedaan door een reeks van experimenten waarbij architectuurprincipes zijn geherformuleerd door middel van het Dragon1 stappenplan voor het formuleren van architectuurprincipes. Een uitgebreid verslag hiervan is te vinden in bijlage C.

De Dragon1 methode gaat uitgebreid in op de structuur en eigenschappen van een architectuurprincipe. De aanpak die deze methode biedt, vormt dan ook het startpunt voor dit onderzoek. Door het herformuleren van architectuurprincipes is ervaring opgedaan met het formuleren van architectuurprincipes en met de aanpak van Dragon1. Dit hoofdstuk vormt een eerste stap in een zoektocht naar het beter formuleren van architectuurprincipes.

10.1 Knelpunten en problematiek architectuurproces Stater NV

Alvorens het herformuleren van de architectuurprincipes heeft een open interview plaats gevonden met een domeinexpert waarin de belangrijkste knelpunten en problemen omtrent de architectuurprincipes van Stater NV zijn vastgelegd. Hieronder worden de belangrijkste bevindingen weergegeven:

- **Tijdigheid:** De uitwerking van de architectuurprincipes zijn niet gereed of beschikbaar bij aanvang van een groot automatiseringstraject. Hierdoor loopt architectuur achter de feiten aan.
- **Mate van detail:** De uitvoerende partij heeft veel interpretatie ruimte. Dit is goed opgepakt door het ervaren ontwikkelteam van Stater NV, maar had bij een ander partij makkelijk tot grote afwijkingen kunnen leiden.
- **Onduidelijkheid semantiek:** Maandelijks veranderende de betekenis van bepaalde begrippen en de daarbij gestelde doelstellingen. Men is tijdens het traject nog zoekende.
- **Architectuur wordt uitgelegd o.b.v. prototypes in .Net.** Erg kostbaar en niet echt effectief.
- **Geen verankering architectuur in de business van Stater NV.** D.w.z dat het nastreven van een bepaald architectuurprincipe zou leiden tot onacceptabele hoge doorlooptijd en kosten voor aspecten die nu commercieel door de business niet als belangrijk of haalbaar worden geacht. M.a.w. men heeft getracht geprobeerd door middel van de opgestelde architectuur zich in te dekken voor alle mogelijke toekomstige wellicht handige situaties.
- **Vanwege het niet op tijd en concreet hebben van richtlijnen, kwamen requirements/architectuur richtlijnen achteraf.** Dit moet door rework weer worden doorgevoerd en dit zorgt voor een langere doorlooptijd langer en brengt hoge kosten met zich mee.
- **Het ontbreken van concrete en op de praktijk afgestemde richtlijnen hebben bij het ontwikkelteam dat verantwoordelijk is voor middleware, tot ingrijpend rework geleid.** Consequentie hiervan is veel vertraging en extra kosten. Dit is rechtstreeks het gevolg van onvolledige formulering van de architectuurprincipes en achteraf aanleveren van aanvullende onverwachte requirements.
- **Weinig pragmatisch:** D.w.z dat veel nadruk wordt gelegd op het realiseren van flexibiliteit door middel van architectuurprincipes, terwijl onzeker is of dit ooit nodig is en waarvan duidelijk is dat eerste X klanten er weinig behoefte aan hebben.

- De architecten waren niet nauw genoeg betrokken in het project waardoor men de feeling verloor met praktijk problemen, business context en project problematiek en daardoor niet kunnen sturen en het proces eigenlijk alleen frustreren ondanks goede bedoelingen.

10.2 Herformuleren architectuurprincipes Stater NV

Het herformuleren van de architectuurprincipes van Stater NV is in samenwerking met Wim van Stokkum uitgevoerd. Er is één architectuurprincipe compleet (d.w.z. aan alle attributen invulling gegeven) geherformuleerd door middel van het Dragon1 stappenplan voor het formuleren van architectuurprincipes. Daarnaast zijn er nog een Op basis hiervan is beoordeelt hoe bruikbaar dit stappenplan is en in hoeverre de problematiek opgelost / voorkomen zou kunnen worden indien deze aanpak gehanteerd zou zijn. De belangrijkste bevindingen op basis van een waardeoordeel van Wim van Stokkum zijn als volgt:

Bruikbaarheid:

- Een aantal attributen die beschreven dienen te worden zijn zonder achtergrond informatie en kennis op het gebied van EA lastig te begrijpen. Hier kan een duidelijke omschrijving van wat er verwacht wordt hulp bieden.
- Een bijkomstigheid van bovenstaand punt is dat een aantal attributen lijken te overlappen. De verschillen zijn niet in één oogopslag duidelijk waardoor het gevoel ontstaat dat sommige informatie twee keer wordt gevraagd. Een duidelijkere volgorde voor het invulling geven aan de verschillende attributen zou hier bij kunnen helpen.
- De diepgang waarmee de attributen beschreven te worden is niet beschreven in het stappenplan. Dit is natuurlijk de keuze van degene die de architectuurprincipes opstelt, maar een suggestie / indicatie zou de onervaren architect helpen.
- Van een aantal attributen die worden toegekend aan architectuurprincipes is de meerwaarde niet duidelijk. Hieraan moet meer aandacht worden besteed bij het uitleggen van wat de verschillende attributen inhouden en waar deze voor dienen.
- Er wordt gesteld dat niet alle informatie even relevant is voor iedere doelgroep. Echter is wel alle informatie nodig om alle doelgroepen te voorzien van voldoende informatie omtrent het architectuurprincipe.
- De relatie en wijze waarop de architectuurprincipes kunnen worden doorvertaald naar regels etc. ontbreekt in het stappenplan.

Voordeel van gebruik stappenplan:

- Het beschrijven van de impact van een architectuurprincipe en de mogelijke kosten (een inschatting) kunnen zeer waardevol zijn bij het beslissen of een architectuurprincipe onderkend moet gaan worden door een organisatie.
- De gedetailleerde beschrijving van het architectuurprincipe levert veel meer informatie en inzicht op in wat het architectuurprincipe nu inhoud dan het oorspronkelijke architectuurprincipe.
- Het beschrijven van de rationale achter het architectuurprincipe en de doelen die verwezenlijkt worden met het hanteren van het principe geven een veel betere indruk van waarom men het architectuurprincipe onderkend en wil toepassen.

Op basis van de bevindingen is een kort nieuwe stappenplan opgesteld met Mark Paauwe dat na overleg en aanpassingen gebruikt is om een drietal architectuurprincipes te herformuleren. Het doel van het herformuleren is om te achterhalen welke misstanden de huidige architectuurprincipes herbergen.

Hieronder worden de belangrijkste bevindingen weergegeven die zijn meegenomen in het verdere onderzoek:

- De uitspraken die architectuurprincipes genoemd worden zijn vaak regels, eisen en wensen of deelbeschrijvingen van architectuurprincipes.
- Architectuurprincipes zijn vaak zeer onvolledig geformuleerd. Maar een deel van het resultaat en de wijze waarop het resultaat bereikt dient te worden is beschreven.
- Dient een architectuurprincipe zelf een handhavingmechanisme te zijn of moet er een handhavingmechanisme afgedwongen worden voor een architectuurprincipe?
- Architectuurprincipes zijn niet vertaald naar de toepassing ervan in de context. Hierdoor wordt eigenlijk (bestaande) theorie opgeschreven, zonder een praktijkvertaling te maken. Dit leidt tot een uitspraak die weinig inhoud heeft. Er is geen invulling gegeven aan de context en daardoor veel te algemeen. De architect laat beslissingen over aan anderen en laat dus anderen voor architect spelen en neemt te weinig verantwoordelijkheid.
- Architectuurprincipes worden opgeschreven alsof het een waarheid betreft terwijl dit niet onderbouwd kan worden.
- Architectuurprincipes zijn onvoldoende afgeleid van de eisen en wensen van de business.

De volgende bevindingen over architectuurprincipes en het formuleren van architectuurprincipes zijn gedaan tijdens het herformuleren:

- Een architectuurprincipe heeft de vorm van een gevolgtrekking.
- Het gebruiken van de redeneerstrategie die bij propositielogica gehanteerd wordt is in te zetten om een architectuurprincipe scherper te definiëren. Door te kijken of het resultaat bereikt kan worden door middel van de beschreven aanpak kan geverifieerd worden of dit logisch is of aangescherpt dient te worden.
- De tijdspanne waarin een architectuurprincipe van kracht is (gaat worden) is belangrijk te onderkennen.
- De begrippen; resultaat, gevolg, mogelijkheden en onmogelijkheden, lijken te overlappen en kunnen een synoniem zijn voor elkaar.

Het resultaat van dit experiment is o.a. dat de manier om architectuurprincipes te formuleren die Dragon1 voorstelt in het stappenplan veel potentie heeft en daarom als basis wordt genomen voor het verdere onderzoek.

Het is lastig te beoordelen in hoeverre problemen hadden kunnen voorkomen indien een gestructureerd stappenplan was gebruikt voor het opstellen en formuleren van architectuurprincipes. De architectuurprincipes zijn medio 2004 opgesteld. Toentertijd had men een ander beeld voor ogen van wat een architectuurprincipe moest zijn dan nu het geval is.

Een gestructureerd stappenplan voor het opstellen en formuleren van architectuurprincipes helpt echter voorkomen dat veel gemaakte fouten worden herhaald. Het stappenplan stimuleert om na te denken over zaken die de volledigheid en bruikbaarheid van een architectuurprincipes verhogen. Op basis van het herformuleren zijn echter misstanden aan het licht gekomen die voorkomen hadden kunnen worden door de architectuurprincipes anders te formuleren.

11 Functie, eigenschappen en beschrijving van het architectuurprincipe

In dit hoofdstuk wordt de zoektocht naar de functie, eigenschappen en beschrijving van het architectuurprincipe beschreven. Op de volgende vragen wordt in dit hoofdstuk een antwoord gezocht:

- Wat is een architectuurprincipe en wat betekent dit voor de beschrijving er van?
- Is het mogelijk om architectuurprincipes te formuleren die altijd een waarheid betreffen en is dit gewenst?
- Is de voorgestelde structuur van een architectuurprincipe door Dragon1 correct en te gebruiken als generieke structuur voor architectuurprincipes?
- Zijn de attributen die door Dragon1 aan architectuurprincipes worden toegekend voldoende en hoe dienen deze beschreven te worden?
- Welke relaties zijn er tussen de attributen die door Dragon1 aan architectuurprincipes worden toegekend?

Er zijn een aantal korte analyses en experimenten uitgevoerd om meer inzicht te verkrijgen in de 'aard' van architectuurprincipes en zijn eigenschappen. Het doel hiervan is om de verkregen inzichten te kunnen gebruiken om het stappenplan verder uit te kristalliseren.

11.1 Functie en definitie volgens Dragon1

Zoals al eerder beschreven vormt de theorie van Dragon1 de basis voor dit onderzoek. In dit hoofdstuk zullen dan ook de belangrijkste aspecten van Dragon1 m.b.t. architectuurprincipes besproken worden.

Dragon1 hanteert de volgende definitie van een architectuurprincipe:

De gehandhaafde werking van iets (zoals een systeem, concept of fenomeen) dat zorgt voor een geregleerde situatie en beleving in een contextloze ruimte waarmee bepaalde toestanden, gebruiken of toepassingen van iets anders mogelijk worden gemaakt of juist worden beperkt – [Dragon1/08b]

Dragon1 hanteert de volgende structuur voor een beschrijving van een architectuurprincipe:

Als een uitspraak een principe moet worden formuleer dan wat er (steeds) gebeurt, wat dat als gevolg of resultaat heeft en wat dat dan weer mogelijk of onmogelijk maakt.

In de volgende versie van de uitwerking van deze structuur wordt nog meer invulling gegeven aan dit driedeling:

'door ..[een bepaalde technische oplossing of actie].. zal, of wordt ..[een bepaalde gewenste reactie of opgelost probleem].. met als gevolg dat ..[een bepaald opgeleverd resultaat]..' – [Paauwe08]

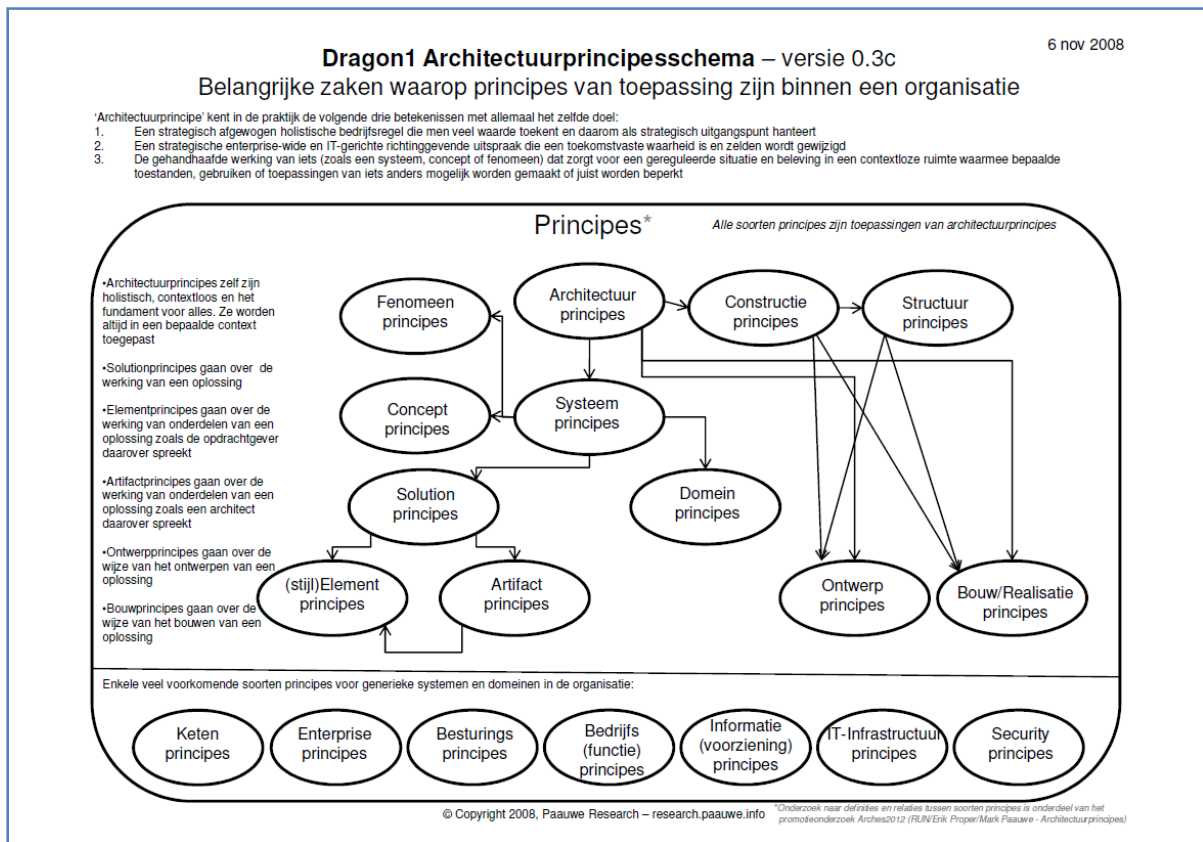
De huidige wijze die door Dragon1 gebruikt wordt voor het formuleren van architectuurprincipes biedt een vaste structuur. Een vaste structuur leent zich voor het meer formeel onderbouwen en misschien wel beschrijven van een architectuurprincipe.

In de structuur is duidelijk een gevolgtrekking aanwezig. Door het doen van verschillende aannames (gebaseerd op een bewezen aanpak) kan geconcludeerd worden dat een bepaald gevolg of resultaat zal optreden. Door dit gevolg of resultaat zal iets mogelijk en /of onmogelijk worden. De structuur heeft dus de volgende delen:

- Deel 1 beschrijft het 'hoe of door'.
- Deel 2 beschrijft 'het gevolg / resultaat of wat men wil bereiken'.
- Deel 3 beschrijft 'wat wordt hierdoor mogelijk / onmogelijk'.

Als in voldoende mate onderbouwd kan worden dat deze structuur logisch en generiek is kunnen de attributen worden gerelateerd aan de verschillende delen en kan een beter stappenplan voor het formuleren van architectuurprincipes worden opgesteld.

De volgende afbeelding geeft een duidelijk overzicht van de denkwijze van Dragon1 m.b.t. architectuurprincipes:



Figuur 2: Dragon1 Architectuurprincipesschema - bron: Dragon1

Deze theorie is vooral gebaseerd op praktijkervaring en daarom is het interessant om deze theoretisch te onderbouwen. In de volgende hoofdstukken zal dan ook dieper ingegaan worden op wat een architectuurprincipe is, welke structuur deze heeft en hoe deze beschreven kan worden.

11.2 Gevolg, resultaten, mogelijkheden: synoniem?

Op basis van het toepassen van de voorgestelde structuur en het analyseren van de structuur is het volgende geconstateerd:

- Deel 1 beschrijft het 'hoe of door' en vormt dus een duidelijk deel van een architectuurprincipe. Deel 2 en 3 lijken echter in eerste instantie op elkaar. Een gevolg, resultaat, mogelijkheid en onmogelijkheid lijken hetzelfde te zijn.

De vraag hierbij is: Is de driedeling nader te verklaren en logisch gekozen en kan deze scherper worden gedefinieerd?

Het doel hierbij is om globaal een antwoord te verkrijgen op de vraag of de driedeling op te delen is in diverse bouwstenen en of de driedeling wel noodzakelijk is.

Op basis van de bovenstaande constatering is eerst gekeken naar deel 2 van de structuur. Is er een verschil aan te wijzen tussen resultaat en gevolg?

Hieronder volgen de twee definities van deze twee begrippen [van Dale]:

re-sul-taat het; o -taten **1** uitslag, uitkomst **2** gevolg: zonder ~

ge-volg het; o **1** de personen die een hooggeplaatst persoon begeleiden **2** -en dat wat uit iets voortvloeit; resultaat: oorzaak en ~; ten ~e van brand door brand || ~ geven aan een uitnodiging die beantwoorden door te komen

De twee begrippen zouden op de volgende manier gebruikt kunnen worden:

Bij een resultaat staat de uitkomst / uitslag vast en is deze dus correct en volledig te formuleren. Bij een gevolg is dit lastiger. Men onderkent dat een gevolg plaats vind, maar hoe deze er precies uit ziet is lastiger te formuleren. Doordat dit niet kan, is het mogelijk om te speculeren / specificeren wat er uit het gevolg kan volgen etc. Deze theorie is interessant om verder te onderzoeken omdat deze ook een relatie heeft met de hardheid (bewijsbaarheid / mate waarin een architectuurprincipe waar is) van een architectuurprincipe. De architectuurprincipes die een gevolg hebben zijn minder hard dan architectuurprincipes die een direct resultaat hebben.

Als een architectuurprincipe uit drie delen bestaat waarvan één het startpunt vormt en het tweede deel een gevolg of resultaat is en het derde deel ook weer een gevolg of resultaat kan op basis van bovenstaande uiteenzetting het logischer zijn om deel 2 als een gevolg te definiëren en deel 3 als het (eind)resultaat.

11.3 Bouwstenen en structuur (deel 1)

Op basis van de resultaten van het onderzoek tot nu toe naar het doel, de structuur en beschrijving van een architectuurprincipe is de volgende definitie opgesteld voor de structuur en beschrijving van een architectuurprincipe:

Een architectuurprincipe beschrijft een aanpak (best-practice), werkwijze, oplossing en het resultaat hiervan. Deze beschrijving is holistisch, contextloos. Een architectuurprincipe is fundamenteel, de basisbouwsteen of opgebouwd uit andere architectuurprincipes.

Het lijkt op basis van deze definitie dat er eigenlijk maar twee delen zijn. Deze mogelijkheid wordt hier verder onderzocht.

De verzamelnaam voor aanpak (best-practice), werkwijze of oplossing wordt in deze theorie 'aanpak' genoemd om zo duidelijkere en kortere zinnen te kunnen formuleren. Overal waar aanpak wordt genoemd kan dus ook werkwijze, oplossing etc. ingevuld worden.

De verzamelnaam voor een gewenst effect, resultaat, doel, mogelijkheid etc. wordt in dit hoofdstuk 'resultaat' genoemd.

De structuur van een architectuurprincipe op basis van de definitie is:

Aanpak => Resultaat

Informeel kan een architectuurprincipe dus als volgt worden beschreven:

[...] resulteert in [...]

Doordat een architectuurprincipe contextloos is kan het resultaat nauwkeurig beschreven worden. Het resultaat is als het ware bekend. Meerdere architectuurprincipes samen kunnen weer een nieuw architectuurprincipe vormen. Het volgende theoretische voorbeeld geeft hier een beeld van:

Architectuurprincipe 1: Aanpak 1 => Resultaat 1

Architectuurprincipe 2: Aanpak 2 => Resultaat 2

Architectuurprincipe 3: (Aanpak 1 => Resultaat 1) ^ (Aanpak 2 => Resultaat 2) => Resultaat 3

of

$$((\text{Aanpak 1} \wedge \text{Aanpak 2}) \Rightarrow (\text{Resultaat 1} \wedge \text{Resultaat 2})) \Rightarrow \text{Resultaat 3}$$

De wijze van formuleren in propositielogica is gebruikt omdat dit een overzichtelijke weergave geeft van de mogelijkheden. De correctheid van de opgeschreven stellingen zijn hier niet van belang. Beide kunnen van toepassing zijn.

Alle tot nu toe opgeschreven architectuurprincipes die in de praktijk gebruikt zijn of worden passen in deze structuur. 'Aanpak' en 'Resultaat' kunnen gezien worden als losse bouwstenen van een architectuurprincipe. Veel uitspraken die in de praktijk architectuurprincipes worden genoemd beschrijven een aanpak of resultaat of beide.

Dit laatste zorgt er echter voor dat bovenstaande beschrijving incompleet is. Een resultaat van iets kan ook weer ingezet worden als aanpak voor een ander resultaat. In bovenstaande uiteenzetting is een te klein deel van dit schakelmechanisme beschreven. De volgende structuur is namelijk van toepassing:

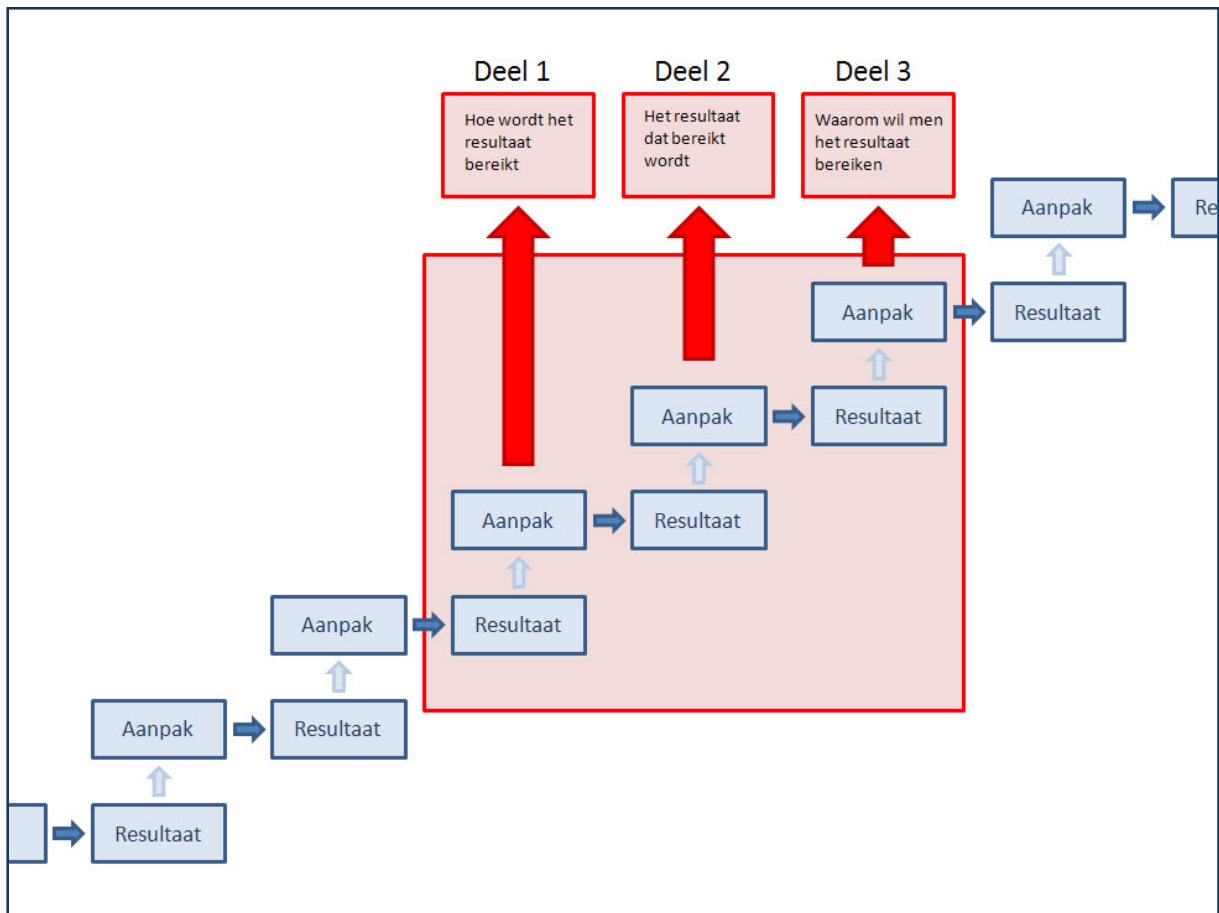
$$(((\text{Aanpak} \Rightarrow \text{Resultaat}) \Rightarrow \text{Resultaat}) \Rightarrow \text{Resultaat}) \Rightarrow \text{etc.}$$

Eigenlijk bestaat een architectuurprincipe uit een reeks van schakelingen van bouwstenen die ergens zijn oorsprong heeft en ergens ooit stopt. Het resultaat van een aanpak vormt weer de aanpak voor een ander resultaat. Deze constatering dient gerelateerd te worden aan de voorgestelde driedeling.

11.4 Bouwstenen en structuur (deel 2)

Zoals eerder geschreven beschrijft een architectuurprincipe vaak iets wat men wel bereiken (wenst) of hoe men iets wil bereiken. De structuur van Dragon1 vangt beide af en is daarom krachtig. In het eerste gedeelte wordt je als het ware gedwongen te beschrijven met welk middel je iets wil bereiken en in het tweede gedeelte staat wat er bereikt wordt. Vervolgens staan in deel 3 de mogelijkheden en of onmogelijkheden. Als je echter gaat kijken naar bijvoorbeeld de belangrijkste attributen die beschreven moeten worden van een architectuurprincipe, kom je tot de conclusie dat een structuur bieden alleen niet voldoende is. Er valt nog altijd over te twisten waar je een deel van een uitspraak moet plaatsen. Neem bijvoorbeeld de attributen die door diverse methoden worden beschreven; motivatie, impact, consequenties. In het 'hoe of door' gedeelte kan zowel een motivatie, impact of consequentie staan. In deel 2 en 3 is dit ook mogelijk. Hieruit zou geconcludeerd worden dat de structuur nog verder opgedeeld zou kunnen worden, maar bovenal dat er altijd nog een stuk voor hetgeen zit wat men een architectuurprincipe noemt en dat er altijd een stuk na komt.

Doordat een beschrijving van een architectuurprincipe eigenlijk maar een deel van een keten van aanpakken en resultaten beschrijft is de driedeling die Dragon1 voorschrijft noodzakelijk. Het middengedeelte beschrijft dan het resultaat (het gewenste resultaat). De stap voor dit resultaat (hoe wordt het resultaat bereikt) wordt beschreven in deel 1 en de stap er na wordt beschreven deel 3 (wat brengt het resultaat met zich mee). Het middengedeelte wordt als het ware in zijn context weergegeven en kan daardoor completer beschreven worden. Men kan als men het noodzakelijk vindt altijd nog verder kijken en de scope uitbreiden. Doordat het in de praktijk handiger is om een driedeling te gebruiken door bovenstaande redenen, moet er echter wel consistent omgegaan worden met het invulling geven aan deze drie delen. De volgende visualisatie is van toepassing op de bovenstaande verwoording:

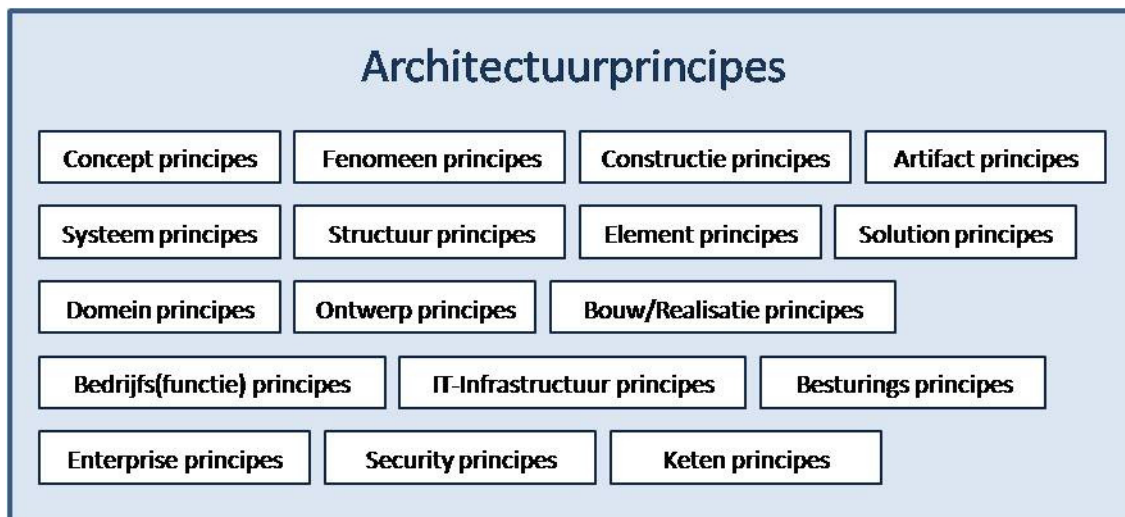


Figuur 3: Bouwstenen architectuurprincipe

De uiteenzetting die beschreven is in dit hoofdstuk is tot stand gekomen door het herformuleren en analyseren van diverse praktijkvoorbeelden (zie bijlage C) en het analyseren van de theorie van Dragon1 en gesprekken met Mark Paauwe. Tijdens het herformuleren en analyseren van deze praktijkvoorbeelden zijn diverse inzichten aan het licht gekomen die in het huidige stappenplan van Dragon1 verwerkt kunnen worden. Daarnaast is er de visie dat de invulling van dit driedeling scherper kan worden gesteld, gerelateerd kan worden aan attributen die beschreven moeten worden van een architectuurprincipe en dat bestaande architectuurprincipes in deze structuur kunnen worden gegoten en ontbrekende stukken kunnen worden aangevuld. Het huidige stappenplan geeft een aanzet. Deze kan echter nog strakker getrokken en van meer detail worden voorzien. De conclusie is dat de structuur generaliseerbaar is. Alle varianten van architectuurprincipes die men tegen zal komen in de praktijk is in te passen in de structuur. Deel 3 is direct af te leiden van een strategisch uitgangspunt of hogere doelstelling.

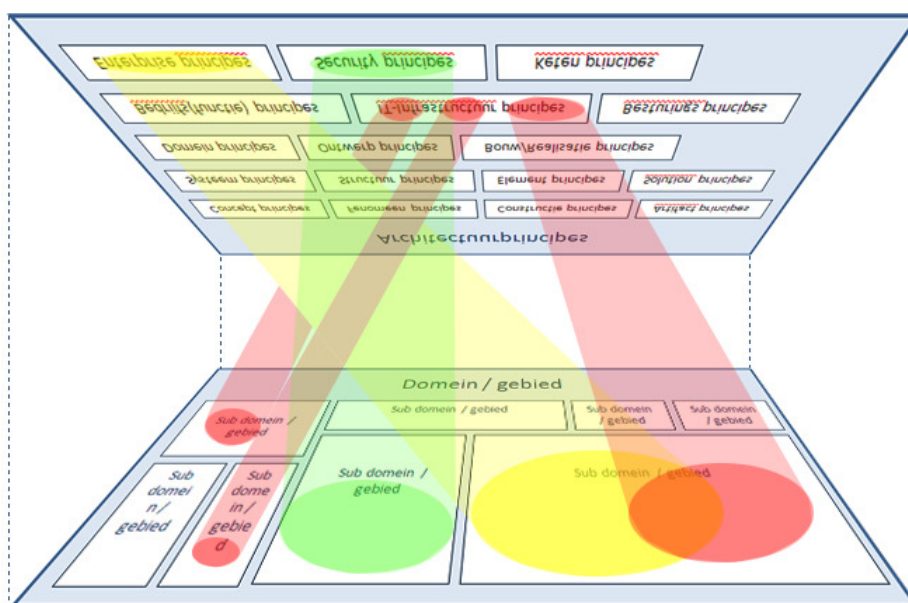
11.5 Soorten architectuurprincipes

In de verschillende EA methoden wordt onderscheid gemaakt tussen diverse soorten principes (bijvoorbeeld ontwerpprincipes, constructieprincipes, besturingsprincipes, security principes etc.). Mijn mening is dat deze soorten gewoon architectuurprincipes zijn. De soorten geven echter aan in wat voor soort context of op welk gebied het architectuurprincipe zou kunnen worden toegepast of van toepassing is. Een architectuurprincipe is een soort van verzamelnaam voor de contextloze soorten principes. De volgende visualisatie is van toepassing:



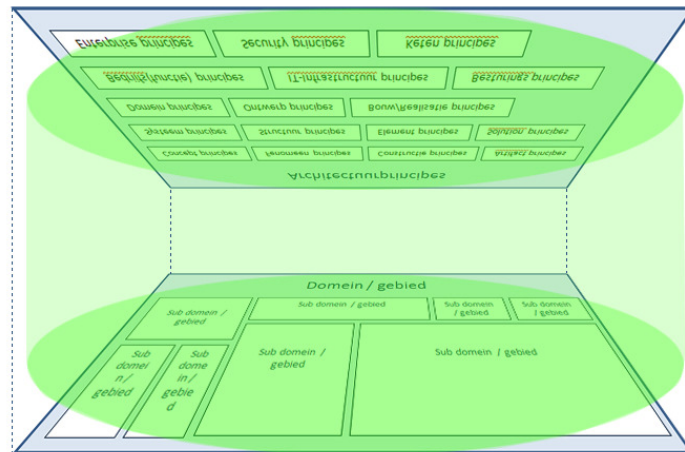
Figuur 4: Soorten architectuurprincipes (1) - *De soorten principes zijn afkomstig van Dragon1.

Deze soorten zijn echter wel belangrijk om te onderkennen. Om een stabiele en gewenste situatie te creëren moeten in theorie alle soorten architectuurprincipes terug te vinden zijn in de architectuur. Het negeren, uitsluiten of niet opnemen van een soort architectuurprincipe leidt tot gaten in de architectuur. Alle gebieden van een architectuur kunnen afgevangen worden door de verschillende soorten. De naam die een soort heeft maakt dus in feite niks uit (de vraag of iets een bouwprincipe of realisatieprincipe heet doet er niet toe), het gaat er om dat alle mogelijke gebieden of domeinen worden afgedekt met een soort architectuurprincipe. Een soort architectuurprincipe kan echter ook in meerdere gebieden of domeinen noodzakelijk zijn. Er is dus geen 1 op 1 relatie. Ondertaannde visualisatie is van toepassing en geeft een beeld bij de bovenstaande uitleg. De eerste afbeelding geeft een beeld van een situatie waar niet alles is afgedekt, maar geeft wel aan hoe een afgedekt gebied er uit kan zien.



Figuur 5: Soorten architectuurprincipes (2)

De onderstaande afbeelding geeft de ideale situatie weer waar naar gestreefd kan worden:



Figuur 6: Soorten architectuurprincipes (3)

De soorten architectuurprincipes kunnen ook onderling een relatie hebben. Zo kan een bepaald architectuurprincipe ondersteund, onderbouwd of gerealiseerd worden door middel van diverse soorten architectuurprincipes.

11.6 Architectuurprincipes: valide?

Naast dit onderscheid in soorten architectuurprincipes kan er ook een onderscheid gemaakt worden in de mate waarin een architectuurprincipe een beschreven waarheid betreft, een bewezen theorie / methode / oplossing / aanpak etc. (valide). Hier is grofweg op simpele wijze een tweedeling in te maken:

1. Een architectuurprincipe die altijd waar is. Het resultaat staat vast en is bewezen.
2. Een architectuurprincipe die niet altijd waar is. Het resultaat staat niet vast en moet nog bewezen worden.

Niet alle architectuurprincipes hoeven namelijk per definitie waar te zijn, of zijn maar in een bepaald geval waar (geldig). Denk hierbij aan architectuurprincipes die niet in elke context van toepassing zijn of zullen gaan werken of een principe waar altijd voor- en nadelen aan zitten. Een andere benaming voor 'Aanpak' kan ook 'Aanname' zijn. Een architectuurprincipe bestaat dan uit een serie aannames die moeten zorgen voor een gewenst resultaat. Die aannames hoeven echter niet allemaal waarheden te betreffen. De stelling luidt dan ook:

Wanneer een architectuurprincipe in een context wordt toegepast dient deze (zo veel mogelijk) waar te zijn.

Deze theorie zal kracht worden bijgezet met een (gesimplificeerd) voorbeeld. Hiermee wordt ook een voorbeeld gegeven van een architectuurprincipe waaraan meerdere architectuurprincipes ten grondslag liggen. Neem het volgende (incomplete) architectuurprincipe. Dit is een architectuurprincipe (concept) dat nog geen bewezen concept is maar wel potentie heeft. Het resultaat staat nog niet vast, maar het is wel zeker dat het invloed heeft op in dit geval de kosten of opbrengsten binnen het proces:

Selfservice principe*

Architectuurprincipe 1: Door de klant zelf zijn artikelen te laten afrekenen wordt bespaard op loonkosten van het kassapersoneel.

Architectuurprincipe 2: Door de klant zelf zijn artikelen te laten afrekenen wordt het voor de klant mogelijk om een bedrag af te rekenen dat niet overeen komt met de juiste prijs en kan er verlies gemaakt worden op de verkoop van artikelen.

Resultaat 1: Verandering in kosten en opbrengsten in afrekenproces.

Resultaat 2 (resultaat van resultaat 1): Kostenreductie.

** Dit architectuurprincipe kan op diverse manieren (vast ook nog beter) worden geformuleerd en een andere naam hebben, het gaat hier puur om het voorbeeld en om de tweedeling kracht bij te zetten.*

Het architectuurprincipe wat hierboven is beschreven is eigenlijk maar een fragment van het complete bouwwerk dat bestaat uit een verzameling van architectuurprincipes. Wat hier echter te zien is dat én het architectuurprincipe is opgebouwd uit andere architectuurprincipes én dat er een trade-off zit in het architectuurprincipe én dat de mate waarin een principe waarheid wordt afhankelijk is van de context waarin deze wordt toegepast. Het hierboven geschetste 'Selfservice principe' ziet er dan ook als volgt uit:

(Architectuurprincipe 1 ^ Architectuurprincipe 2) ^ Context => Resultaat 1 => Resultaat 2

De context heeft invloed in de mate waarin een architectuurprincipe een waarheid betreft en dus ook of het resultaat waarheid wordt. Met dit voorbeeld wordt ook getoond dat het inderdaad bijna onmogelijk is om alleen bewezen architectuurprincipes te gebruiken / hebben. In de toekomst zal blijken of dit 'Selfservice principe' stand houdt en in welke mate het een bewezen aanpak wordt. Naar mate het architectuurprincipe vaker wordt toegepast zal dit blijken. Op basis hiervan ondergaat het architectuurprincipe echter een soort van transformatie. Indien bijvoorbeeld blijkt dat het alleen maar van toepassing is in een context waar de sociale controle hoog is dan kan dit als aanname worden toegevoegd aan het architectuurprincipe en wordt het architectuurprincipe scherper gesteld en komt het dus steeds dichterbij een 'zekerheidje'.

Het bovenstaande voorbeeld geeft de huidige stand van zaken (beschrijving) van het architectuurprincipe weer. Wat men echter vaak doet is bij voorbaat het resultaat al vast leggen (door bijvoorbeeld bij het resultaat al kostenreductie in afrekenproces neer te zetten). Men wil graag dat dit architectuurprincipe als resultaat kostenreductie heeft. Door in de praktijk dit architectuurprincipe toe te passen zal blijken of dit zo is en wanneer dit zo is. Op basis hiervan kan het architectuurprincipe aangescherpt (nieuwe bouwstenen toevoegen en bouwstenen aanscherpen / wijzigen) worden en kan het resultaat omschreven worden als kostenreductie in het afrekenproces. Dit is het streven, er voor zorgen dat je bij wijze van spreken een architectuurprincipe uit de kast kan pakken en wanneer alle randvoorwaarden zijn ingevuld dat dan het resultaat zeker bereikt wordt.

Wat ook vaak gedaan wordt is het principe op een positieve wijze te beschrijven door bijvoorbeeld het volgende architectuurprincipe op te schrijven:

Door de klant zelf zijn artikelen te laten afrekenen, wordt bespaard op loonkosten van het kassapersoneel. Hierdoor worden de kosten in het afrekenproces gereduceerd.

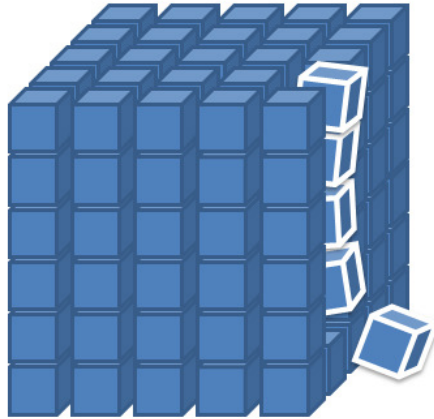
De mate waarin een architectuurprincipe een waarheid betreft is van twee kanten te bekijken:

Betreft het architectuurprincipe zelf een waarheid en betreft de beschrijving van een architectuurprincipe een waarheid.

11.7 Functie van een architectuurprincipe

Een architectuurprincipe wordt toegepast in een context. Een architectuurprincipe zelf is (bewezen) theorie die in een context wordt toegepast omdat men het (positieve) resultaat van een architectuurprincipe wil verwezenlijken in deze context.

Een architectuur is in theorie compleet te beschrijven met architectuurprincipes. Bij wijzen van spreken bestaat een onderneming al compleet uit architectuurprincipes maar zijn deze niet vastgelegd en hangen ze als los zand aan elkaar. Het is bijna onmogelijk in de praktijk alle architectuurprincipes te beschrijven, laat staan dat er alleen bewezen architectuurprincipes kunnen worden toegepast. Een streven kan bijvoorbeeld wel zijn de meest fragiele onderdelen van een onderneming te ondersteunen door bewezen architectuurprincipes zodat hier weinig risico mee wordt gelopen. Dit is slechts een voorbeeld. Onderstaande afbeelding geeft een beeld van de sterkte / kracht van een architectuur en kan op twee manieren geïnterpreteerd worden;



Figuur 7: Architectuur: een bouwwerk van architectuurprincipes

1. Een architectuurprincipe bestaat uit een verzameling van architectuurprincipes. Elk blokje vormt de bouwsteen van een architectuurprincipe. Valt er één om dan vallen de bouwstenen die hier opgebaseerd zijn ook om.
2. Een architectuur bestaat uit een verzameling architectuurprincipes. Elk blokje is hier een architectuurprincipe. Valt er één om dan vallen de architectuurprincipes die hier opgebaseerd zijn ook om en wordt de totale architectuur minder krachtig (sterk).

Een architectuurprincipe is in meerdere situaties in te zetten en dus ook beschikbaar te stellen als gemeengoed en bijvoorbeeld te delen met andere instellingen, ondernemingen, organisaties etc. Dit wordt in de praktijk nog te weinig gedaan waardoor vaak dezelfde architectuurprincipes worden beschreven (met andere woorden).

11.8 Structuur van de beschrijving van een architectuurprincipe

Er is een verschil tussen de beschrijving van een architectuurprincipe en wat een architectuurprincipe is. Een architectuurprincipe is meer dan alleen de uitspraak (het short statement) die de aanpak en het resultaat beschrijft in één of twee zinnen. Deze uitspraak moet dan ook gezien worden als een samenvatting van het architectuurprincipe. Deze samenvatting bevat in ieder geval kernachtig omschreven de aanpak(ken) die gebruikt worden om het resultaat te behalen en het resultaat zelf. Naast deze samenvatting dient ook invulling gegeven te worden aan diverse attributen die het architectuurprincipe compleet maken. Een vereiste aan de beschrijving van een architectuurprincipe is dat deze contextloos is.

De attributen die onderkend zullen worden zijn afkomstig van Dragon1. Een overzicht van de attributen en hun beschrijving is te vinden in bijlage D. Er zal in dit hoofdstuk geen compleet en uitgebreid overzicht gegeven worden van welke attributen van een architectuurprincipe moeten worden vastgelegd en hoe hier invulling aan moet worden gegeven. Dit wordt verder behandeld in hoofdstuk 12.

11.9 Redeneerstrategie en architectuurprincipes

Een strategie die gebruikt kan worden voor het formuleren van architectuurprincipes is de redeneerstrategie van propositielogica. Dit zal duidelijk gemaakt worden met het volgende voorbeeld:

Stel dat men de volgende stelling heeft en dit opgeschreven heeft als architectuurprincipe. Men heeft het gevoel dat het resultaat niet altijd bereikt zal worden waardoor het architectuurprincipe niet voldoende zal werken. Men heeft de volgende stelling geformuleerd:

Aanname 1 ^ Aanname 2 => Gevolg 1 => Resultaat 1

Door logisch te redeneren of 'Gevolg 1' volgt uit 'Aanname 1' en 'Aanname 2' kan men concluderen of deze propositie waar is. Men komt tot de conclusie dat dit niet het geval is. Men voegt 'Aanname 3' toe. Resultaat:

Aanname1 ^ Aanname2 ^ Aanname3 => Gevolg 1 => Resultaat 1

Men komt echter tot de conclusie dat 'Aanname 3' niet sterk genoeg is en dat deze bijvoorbeeld maar in 70% van de gevallen waar is. Men komt de conclusie dat 'Aanname 4' ook een toevoeging kan zijn aan het bewijs en stelt dat als 'Aanname 3' of 'Aanname 4' van toepassing is, het beoogde gevolg bereikt zal worden en dus ook het gewenste resultaat behaalt zal worden.

Aanname1 ^ Aanname2 ^ (Aanname3 v Aanname4) => Gevolg 1 => Resultaat 1

Doordat echter 'Aanname 4' een bepaald nevengevolg 'Gevolg 2' heeft kan niet worden gesteld dat alleen 'Resultaat 1' altijd bereikt zal worden. 'Gevolg 1' en 'Gevolg 2' vormen samen de basis voor 'Resultaat 3'. Het eindresultaat zal dan als volgt zijn:

Aanname1 ^ Aanname2 ^ (Aanname3 v Aanname4) => Gevolg 1 ^ Gevolg 2 => Resultaat 3

Met dit theoretisch voorbeeld wordt getracht inzichtelijk te maken hoe het formuleren van architectuurprincipes in zijn werk gaat en hoe de beschrijving hiervan dus transformaties ondergaat. Door de relatie tussen de attributen is het van belang deze in de juiste volgorde te beschrijven.

De bovenstaande redeneerstrategie wordt al onbewust toegepast. Het bewust toepassen van de redeneerstrategie zorgt er voor dat je kritisch kijkt naar wat er staat en zorgt voor een beschrijving die dichter ligt bij de werkelijkheid en daardoor S.M.A.R.T is beschreven.

Het vervelende bij het formuleren van architectuurprincipes is dat een resultaat bij wijzen van spreken ook weer een aanpak is en het daardoor lastig is om een architectuurprincipe af te bakenen en dito de beschrijving. Wat van belang is dat men een set van architectuurprincipes op een zelfde wijze opstelt en geen lijst van architectuurprincipes onder elkaar zet terwijl er wel degelijk een hiërarchie en relatie is tussen deze architectuurprincipes.

11.10 Het werken met bewezen architectuurprincipes

Er wordt vaak gesteld dat het bijna onmogelijk is om binnen EA te werken met bewezen architectuurprincipes die altijd waar zijn. Men heeft het hier echter niet over architectuurprincipes zoals bovenstaande uitleg de definitie en functie van een architectuurprincipe verwoord. Wanneer een architectuurprincipe wordt toegepast in een bepaalde context wordt het inderdaad een stuk lastiger om het mechanisme dat het architectuurprincipe beschrijft te laten werken en voor het gewenste resultaat te laten zorgen. Op dat moment tredt namelijk het actie / reactie (oorzaak / gevolg) mechanisme in werking en kan het toepassen van een architectuurprincipe oneindige gevolgen hebben. Dit doet echter niets af van het feit dat het wel degelijk mogelijk is om ook te werken op basis van bewezen architectuurprincipes. Er zullen echter ook altijd architectuurprincipes ingezet worden die nog niet compleet bewezen zijn. Eigenlijk is het een vreemd verhaal om te stellen dat een architectuurprincipe niet waar hoeft te zijn. Dat is juist de kracht van een architectuurprincipe en het werken onder architectuur, het toepassen van bewezen aanpakken. Omdat het

lastig is een architectuurprincipe toe te passen in een context om het gewenste resultaat te verkrijgen wil nog niet zeggen dat het in theorie niet mogelijk is en dat men er dus wel naar moet streven.

Op basis van deze theorie is het mogelijk om een soort van 'repository' op te zetten waarbij men architectuurprincipes kan selecteren op basis van bijvoorbeeld resultaat, aanpak, soort principe, etc. Aangezien een doel van architectuurprincipes kan zijn om bewezen kennis en kunde te delen (ook door meerdere organisaties onderling) kan de beschrijving van een architectuurprincipe bij wijze van spreken niet uitgebreid genoeg zijn.

11.11 Het toepassen van een architectuurprincipe

Als een architectuurprincipe een holistische (contextloze) beschrijving is hoe noemen we dan de uitspraken die in de praktijk worden gehanteerd? Het antwoord op deze vraag is eigenlijk heel simpel. De uitspraken zijn te verdelen in uitspraken waarin context is opgenomen en uitspraken waar geen context is opgenomen. Is er context in opgenomen? Dan is het de beschrijving van een toepassing van een architectuurprincipe (deze is echter vaak onvolledig). Is er geen context in opgenomen? Dan is het een architectuurprincipe of een deel hier van. Deze uiteenzetting past binnen de gangbare eigenschappen die worden toegekend aan architectuurprincipes en soortgelijken. De relatie t.o.v. van regels, uitgangspunten etc. wordt in dit onderzoek niet nader onderzocht, maar zou in vervolg onderzoek wel noodzakelijk zijn.

De complexiteit voor het werken onder architectuur met behulp van architectuurprincipes komt eigenlijk pas tot uiting wanneer men de architectuurprincipes gaat toepassen (implementeren) in een context. Opeens komen er veel meer variabelen en invloeden vanuit de context die het lastig maken het architectuurprincipe op de juiste wijze in te zetten en overeind te houden in de situatie. Ook treed op dat moment het actie / reactie (oorzaak / gevolg) mechanisme dat inherent is aan het toepassen van architectuurprincipes. Het resultaat wordt een gevolg en het gevolg krijgt weer gevolgen. Hierdoor ontstaat er een complexe situatie waar het lastig (lees: onmogelijk) is om alle gevolgen (consequenties, impact) te herkennen, laat staan vast te leggen in een document. Vandaar dat het verstandig is om de driedeling te gebruiken. De redenen hiervoor zijn al eerder genoemd.

Hier komt een ander probleem om de hoek kijken. Het onderkennen en toepassen van een architectuurprincipe moet zwaar gewogen worden. In de praktijk wordt vaak de ideale situatie geformuleerd door middel van uitspraken die moeten doorgaan voor architectuurprincipes (of toepassingen hiervan). Bij nader inzien (onderzoek naar consequenties en impact) blijkt de geschetste situatie helemaal niet zo ideaal en komt men tot de conclusie dat bepaalde architectuurprincipes niet onderkend of toegepast hadden mogen worden. Het knelpunt / probleem wat de huidige definities van architectuurprincipes door diverse EA methoden veroorzaken is dat het opschrijven van een architectuurprincipe en daar eventueel wat context in te verwerken al als voldoende wordt gezien en dat de precieze details wel in een later stadium kunnen worden beschreven. Hoe kun je dan echter voorkomen, dat in een later stadium blijkt, dat bij het uitwerken van de toepassing van een architectuurprincipe het gewenste resultaat toch niet bereikt zal worden. Er is dan al een weg ingeslagen die veel tijd en moeite heeft gekost. Architectuur vormt het fundament van de onderneming of het systeem en daar mag / moet bij wijzen van spreken tot in detail toe beredeneert worden voordat er keuzes worden gemaakt.

De theoretische structuur van een toegepast architectuurprincipe op basis van deze theorie is (propositielogica):

Aanpak ^ Context => Gevolg => Resultaat

Hierbij is 'Resultaat' de situatie die ontstaat wanneer het actie / reactie (oorzaak / gevolg) mechanisme is gestopt. Aangezien dit in werkelijkheid nooit ophoudt moet men hier een grens stellen. Het gevolg zal altijd gevolgen hebben en de architect bepaald tot hoever men hier in door moet gaan en bepaald wanneer het gevolg het resultaat wordt. Deze grens is afhankelijk van de scope waarop de toepassing van het architectuurprincipe betrekking heeft. Wanneer deze bereikt is zal er echter nooit één resultaat zijn. Er zijn vaak meerdere resultaten mogelijk (bijvoorbeeld een gewenst of ongewenst resultaat). Wat echter de taak van de architect is om er voor te zorgen dat alles in werking wordt gesteld om de kans dat het gewenste resultaat behaald zal worden het grootste is.

Het toepassen van een architectuurprincipe resulteert dus in een document waarin beschreven staat hoe en waarom een architectuurprincipe wordt toegepast binnen de context en hoe deze gaat leiden tot de gewenste situatie. De beschrijving van het architectuurprincipe vormt de basis en zal worden uitgebreid met informatie uit de context. De precieze uitwerking zal in een later stadium worden uitgewerkt. Het belangrijkste blijft echter de beschrijving van het statement dat de kern (in de vorm van een samenvatting) vormt van de beschrijving van een toegepast architectuurprincipe. Dit is het stuk dat gemakkelijk communiceerbaar is. De uitgebreide achtergrond informatie en documentatie is echter van groot belang. Zoals een architectuurprincipe het fundament van een architectuur dient te zijn, zo dient de documentatie het fundament van de beschrijving van een toegepast architectuurprincipe te zijn.

11.12 Conclusie

Als je nu vanuit de bovenstaande uiteenzetting gaat kijken wat een architect zijn taak is op het gebied van architectuurprincipes, dan is dat het formuleren én selecteren van architectuurprincipes die zorgen voor de gewenste situatie en het toepassen van deze architectuurprincipes in de praktijk. Aangezien architectuurprincipes over het algemeen nog niet op een dergelijke wijze voorhanden zijn wordt de architect vaak wel gedwongen om architectuurprincipes zelf te formuleren. Dit is echter niet zijn kerntaak, maar meer een uit de huidige praktijk gedwongen noodzakelijkheid. Het verwerken van de architectuurprincipes in een ontwerp is hetgeen waar een architect eigenlijk mee bezig zou moeten zijn.

In bovenstaande uiteenzetting is een onderscheid gemaakt tussen de beschrijving van een architectuurprincipe (contextloos) en de beschrijving van hoe een architectuurprincipe in toegepaste vorm in een context er uit ziet. Geconcludeerd kan worden is dat dezelfde attributen beschreven kunnen worden. Wanneer een architectuurprincipe toegepast gaat worden, wordt de beschrijving als het ware uit de kast gepakt en aangevuld met informatie uit de context. Vanuit deze conclusie kan er gesteld worden dat er twee soorten stappenplannen noodzakelijk zijn. Eén om een architectuurprincipe te formuleren en één om een architectuurprincipe toe te passen in een context. Daarnaast kan worden gesteld dat de invulling die gegeven wordt aan de verschillende attributen afhankelijk is van de invulling en relaties tussen de attributen onderling. Er is dus een bepaalde volgorde waarin attributen ingevuld dienen te worden. Verder moet er meer aandacht besteed worden aan op welke wijze er invulling gegeven kan worden aan de diverse attributen. Op basis van het onderzoek kan ook geconcludeerd worden dat het gewenst is om de mate waarin en architectuurprincipe bewezen is vast te leggen.

12 Een stappenplan voor het formuleren van architectuurprincipes

Door de conclusies die getrokken zijn uit het vorige hoofdstuk kan er gesteld worden dat er drie mogelijkheden zijn voor het opstellen van een stappenplan voor het formuleren van architectuurprincipes.

1. Een stappenplan voor het formuleren van (contextloze, holistische) architectuurprincipes
2. Een stappenplan voor het toepassen van architectuurprincipes binnen een organisatie
3. Een stappenplan voor het formuleren van architectuurprincipes waar geen onderscheid wordt gemaakt tussen de bovenstaande twee mogelijkheden.

Eén van deze drie mogelijkheden zal uiteindelijk uitgewerkt dienen te worden in de vorm van een compleet stappenplan. Het voorwerk zal echter zo generiek mogelijk zijn, zodat het mogelijk is om alle mogelijkheden uit te werken.

12.1 Inhoud van de attributen

Eén van de conclusies uit het vorige hoofdstuk is dat de invulling van de diverse attributen nader bekeken dient te worden. Er zal gekeken worden welke attributen, welke invulling dient te krijgen en of attributen kunnen worden weggelaten of dienen worden toegevoegd of opgesplitst. Per attribuut zal dit besproken worden. De originele invulling die Dragon1 geeft aan deze attributen wordt weergegeven indien deze nog steeds van toepassing is (deze is schuin gedrukt). Indien er wijzigingen of toevoegingen aan zijn gedaan op basis van een constatering dan staat dit duidelijk vermeld. Achter de naam van het attribuut staan cijfers weergegeven om te laten zien welke attributen bij de in de inleiding gegeven mogelijkheden van toepassing is.

Een unieke universele ID (1,2,3)

Vanwege de (miljoenen) principes in een enterprise architectuur dient een principe een unieke identificatie te hebben.

Aanvulling

Het ID betreft een uniek nummer of unieke code.

Een unieke universele naam (1,2,3)

Een kort statement van één tot tien woorden die duidelijk maken wat de context en werking van het principe is. Bijvoorbeeld: het zwaartekracht-principe of het gedigitaliseerde werkprocessenprincipe.

Systeem, Concept, of Stijl verwijzing (1,2,3)

Een principe is altijd onderdeel van een systeem, concept, fenomeen of stijl.

Een systeem is een geheel van onderdelen die samenwerken teneinde een gemeenschappelijk doel te realiseren.

Een concept is een gestructureerde, planmatige of systematische werkwijze.

Een fenomeen is een systeem dat in de natuur voor komt en dat ongekende krachten in zich heeft en zeer tot de verbeelding spreekt.

Een stijl is een verzameling van concepten met gemeenschappelijke (stijl)elementen. We maken onderscheid in systemen, concepten en stijlen voor constructie, ontwikkeling en verandering. Voorbeelden van gangbare stijlen zijn service oriëntatie, component oriëntatie, proces oriëntatie, project oriëntatie, data oriëntatie, event oriëntatie, engine oriëntatie en taakgericht werken. Per stijl is het hoofdstijlelement benoemd.

Aanvulling

De invulling hiervan is afhankelijk van de inhoudelijke invulling van het architectuurprincipe. Op basis van deze invulling kan beschreven worden van welke systemen, concepten, fenomenen of stijlen het architectuurprincipe onderdeel is.

Statement (1,2,3)

De formulering van het principle statement (de omschrijving van een principe) komt nogal nauw. De context waarbinnen het principe geldt en de periode en het abstractieniveau waarin het principe geldig is, dient duidelijk te worden afgebakend.

Omdat een principe een waarheid betreft, zijn woorden zoals mogen, willen, kunnen, zouden en moeten niet gewenst in de naam of omschrijving. De formulering van een principe dient niet stellend te zijn, maar een werking te beschrijven. De formulering dient ultimo een oorzaakgevolg relatie te beschrijven maar dient minimaal een volledige entiteitenrelatie te beschrijven.

De omschrijving van een principe gaat in op het mechanisme of de werking van een concept, systeem, fenomeen of stijl.

De omschrijving van een principe moet zoveel mogelijk S.M.A.R.T. zijn.

Aanvulling

Het statement bestaat volgens Dragon1 uit een short statement en een long statement. Eén van de eerste stappen die men moet nemen bij het formuleren van een architectuurprincipe is invulling geven aan het short statement. Hier worden de belangrijkste zaken direct opgeschreven en de basis van het architectuurprincipe wordt gevormd. Dit short statement dient in een later stadium nog te worden aangepast / bijgewerkt om overeen te komen met de inhoudelijke informatie die is toegekend aan de overige attributen. Het long statement is eigenlijk de complete invulling van het architectuurprincipe waarin alle details zijn verwerkt. Het long statement zal niet worden gebruikt binnen het stappenplan aangezien dit gewoon een uitgebreide samenvatting is van de inhoud van de diverse attributen.

Klasse (1,2,3)

Is het een architectuur, constructie of structuurgericht principe?

Is het een kringgedrag principe, taak/volgorde principe of actie-reactie principe?

Wijziging

Het is niet duidelijk in welke klasse er nu onderscheid gemaakt dient te worden. Er staan nu twee klassensoorten die benoemd kunnen worden. De soorten principes die genoemd worden bij de eerste vraag zijn van dezelfde orde als de soorten genoemd worden bij het attribuut Type / activiteit en wordt dus al afgevangen. Het attribuut Klasse zegt dus alleen of het principe een structuurprincipe, kringgedrag principe, taak/volgorde principe of actie-reactie principe is.

Type / activiteit (1,2,3)

Is het een planningsprincipe, ontwikkel(bouw/ontwerp)principe, transformatie/veranderprincipe of beheer/instandhoudingprincipe?

Wijziging

Het betreft hier het classificeren van het soort principe. De volgende type/soorten worden onderkend:

Architectuur principes, Constructie principes, Structuur principes, Ontwerp principes, Bouw/Realisatie principes, Systeem principes, Fenomeen principes, Concept principes, Domein principes, Solution principes, (stijl)Element principes, Artifact principes, Keten principes, Enterprise principes, Besturings principes,

Bedrijfs(functie) principes, Informatie(voorziening) principes, IT-Infrastructuur principes, Security principes - [Dragon1]

Opmerking

Het verschil tussen het attribuut Klasse en Type is niet duidelijk. Het zijn beide classificatieattributen. Hier dient een duidelijker onderscheid in gemaakt te worden. Bovendien wordt hier gesproken over Type terwijl normaal over Soort gesproken wordt.

Abstractieniveau (2,3)

Is het een contextueel, conceptueel, logisch of fysiek principe?

Opmerking

Het abstractieniveau is afhankelijk van de wijze waarop het architectuurprincipe geformuleerd is en dus invulling gegeven is aan de diverse attributen. Er is een relatie tussen het abstractieniveau en de invulling van de diverse attributen. Als een architectuurprincipe op fysiek niveau is beschreven dan is bijvoorbeeld bekend welke technologieën ingezet gaan worden en kan men dus een gedetailleerdere invulling geven aan attributen zoals Implementatie impact, Bedrijfs voor- en nadelen, Rationale etc.

Het gewenste abstractieniveau van een architectuurprincipe is conceptueel. Bij het beschrijven van een architectuurprincipe staat dit dus eigenlijk vast en zou het attribuut dus niet beschreven hoeven te worden. Vandaar dat dit attribuut niet van toepassing is bij mogelijkheid 1. In de praktijk worden de toepassingen van architectuurprincipe ook gewoon architectuurprincipes genoemd. Indien hier geen onderscheid in wordt gemaakt dan is het abstractieniveau een classificatiemiddel dat achteraf toegewezen kan worden op basis van de inhoudelijke beschreven informatie van het architectuurprincipe.

Soort bedrijfssysteem (1,2,3)

Is het een inkoop, verkoop, marketingcommunicatie, productie, financiën, ICT (IV, ICT-Infra)

Opmerking

Hier kunnen meer algemene mogelijkheden aan toegevoegd worden.

Soort organisatiedomein (1,2,3)

Is het bijvoorbeeld een consumentenmarkt principe, een back office principe, een winkelformuleprincipe of een assortiments-principe

Opmerking

Hier kunnen meer algemene mogelijkheden aan toegevoegd worden. Hier worden los wat voorbeelden gegeven. Men kan nu alles opschrijven hier.

Organisatiesysteem beschouwingniveau (1,2,3)

Is het een ondernemingsprincipe, bedrijfsprincipe, bedrijfsfunctieprincipe, bedrijfsproces, bedrijfstak, handeling.

Aanvulling

Het organisatiesysteem beschouwingniveau beschrijft scope / context waar het architectuurprincipe op van toepassing is. Dezelfde uitleg als bij soort bedrijfssysteem is hier van toepassing. Een toevoeging hierbij is echter dat men ook zelf kan bepalen op welke beschouwingniveau een principe van toepassing is. Er kan bijvoorbeeld duidelijk gekozen worden om een principe niet bedrijfsbreed te hanteren maar alleen binnen een

bepaald deel van de organisatie. Indien dit het geval is dient dit al in het beginstadium bepaald te worden zodat er niet onnodige details worden verwerkt in de overige attributen die buiten het beschouwinggebied vallen.

Eigenaar en Beheerder (1,2,3)

Wie de eigenaar en wie is de beheerder van het principe?

Opmerking

Bij mogelijkheid 1 dient hier de naam van de auteur van het originele architectuurprincipe te staan. Bij mogelijkheid 2 en 3 kan hier de naam staan van degene die verantwoordelijk is voor het toepassen van het architectuurprincipe in de organisatie of het architectuurprincipe heeft beschreven.

Issues en Conflicten (1,2,3)

Wat zijn haken en ogen aan het principe om het in de organisatie te hanteren?

Waar conflicteert dit principe met andere principes, of waar werken regels niet vanwege de verkeerde huidige principes?

Wijziging

De invulling die gegeven moet worden aan dit attribuut is onduidelijk en bovendien worden er twee verschillende vragen gesteld. Dit attribuut is op te delen in twee attributen.

- Issues en conflicten (1,2,3)

Dit attribuut beschrijft de interne issues en conflicten die een architectuurprincipe met zich mee (zal gaan) brengen binnen de context, bijvoorbeeld een organisatie.

Wat zijn haken en ogen aan het principe om het in de organisatie te hanteren?

Waar conflicteert dit principe met andere principes, uitgangspunten, regels binnen de organisatie?

Dit attribuut dient o.a. als input voor het attribuut Implementatie impact en Handhavingmechanisme.

- Alternatieve en conflicterende principes (1,2,3)

Hier dient vermeld te worden met welke andere architectuurprincipes het architectuurprincipe in conflict is en welke alternatieve soortgelijke architectuurprincipes er zijn.

Beschouwingssituatie (1,2,3)

We maken onderscheid in situatietypen principes: AS-IS principes, Plateau n - principes, TO-BE principes en Envision principes.

Een AS-IS principe is een principe dat zich in de huidige situatie (heden - 1 jaar) daadwerkelijk voordoet, waarmee men er verstandig aan doet er rekening mee te houden.

Een plateau n principe is een principe dat zich tijdelijk voordoet (jaar X).

Een TO-BE principe is een principe dat zich in de toekomstige situatie (3-5 jaar) zal gaan voordoen, realistisch en haalbaar is, waarmee men er verstandig aan doet er in de toekomst rekening mee te houden.

Een Envision principe is een principe dat zich in de verre toekomst zal gaan voordoen, maar voor de AS-IS, Plateau1 en TO-BE situatie onrealistisch is.

Aanvulling

Er is nog geen niveau onderkend voor een architectuurprincipe dat men altijd van toepassing wil laten zijn

(tijdsonafhankelijk). Deze wordt toegevoegd onder de naam Timeless principle. De invulling die gegeven dient te worden bij de verschillende mogelijkheden verschilt. Bij mogelijkheid 2 en 3 kan hier beschreven worden wanneer bijvoorbeeld de organisatie het architectuurprincipe van toepassing wil laten zijn en bij mogelijkheid 1 kan aangegeven worden wanneer het architectuurprincipe van toepassing is.

Effect-type (1,2,3)

We maken onderscheid in twee effect-type principes: true-principles en false-principles.

Een true-principle is een principe waarvan het handhavingmechanisme effectief is, en zorgt voor de handhaving van het principe.

Een false-principle is een principe waarvan het handhavingmechanisme faalt, en daarmee het principe niet wordt gehandhaafd.

Wijziging

De begrippen true-principle en false-principle dienen anders te worden omschreven. De huidige omschrijving is onduidelijk over of het hier gaat om het interne en/of externe handhavingmechanisme. De volgende wijziging wordt voorgesteld:

- Een true-principle is een principe waarvan het interne handhavingmechanisme effectief is, en zorgt voor de handhaving van het principe.
- Een false-principle is een principe waarvan het interne handhavingmechanisme niet sterk genoeg is (faalt), en daarmee het principe niet uit zichzelf gehandhaafd wordt en dus een extern handhavingmechanisme noodzakelijk is.

Wat in dit attribuut terug komt is de mate waarin een architectuurprincipe waar (bewezen) is. Een false-principle dient te worden aangevuld met een extern handhavingmechanisme om er voor te zorgen dat het in de context zal gaan werken.

First-principle (1,2,3)

Ieder systeem, stijl, fenomeen of concept heeft een kern (van waarheid, of waar het om draait) die door middel van een principe kan worden geformuleerd. Dit principe is het first principle. Van alle systemen, concepten en stijlen die in een architectuur worden gebruikt dient het first principle te worden geformuleerd.

Opmerking

Bij het toepassen van een architectuurprincipe wordt de beschrijving veranderd en het architectuurprincipe dus aangepast waardoor het onwaarschijnlijk is dat hier ooit ja beantwoord zal worden. Dit is eigenlijk alleen van toepassing bij mogelijkheid 1.

Wijziging

De [ja/nee] invulling is alleen van toepassing bij mogelijkheid 1. Bij mogelijkheid 2 en 3 dient hier het First principle zelf neergezet te worden zodat men de 'echte' bron heeft vastgelegd.

Status (2,3)

We maken onderscheid in de volgende statusovergangen met bijbehorende status van een principe: voorstellen, kandideren, in overweging nemen, onderkennen.

Opmerking

Dit attribuut is alleen van toepassing bij mogelijkheid 2 en 3.

Bron(-vermelding) (1,2,3)

Voorbeelden van bronnen van principes zijn: de natuur, ervaring, een door de industrie ontwikkelde technologie, een best practice, een bewezen concept.

De bronvermelding voor een principe is nooit een wens, eis, requirement of doelstelling. Het eisen van een bepaalde kwaliteit of een veranderdoel zijn wel aanleidingen om een bepaald concept (of principes) te willen of te moeten implementeren.

Opmerking

Hier dient dus ook naast de naam van de bron ook de bronvermelding te staan.

Rationale (1,2,3)

De achterliggende reden of onderbouwing (bijvoorbeeld een hogere orde principe) voor het principe

Opmerking

De achterliggende reden (motivatie) voor het onderkennen van het architectuurprincipe dient hier beschreven te worden. Waarom draagt het onderkennen van het architectuurprincipe bij aan bijvoorbeeld het verwezenlijken van bepaalde strategische uitgangspunten.

Exploitatie en veranderdoelen (1,2,3)

Welke doelen liggen ten grondslag aan de keuze voor het (concept van het) principe?

Opmerking

Hier dienen algemene kerndoelen beschreven te worden zoals kostenreductie, verlagen complexiteit, snelle service etc. Aangezien uit de strategische uitgangspunten de architectuurprincipes afgeleid dienen te worden kunnen deze hier worden vermeld.

De attributen Exploitatie en veranderdoelen heeft dus een relatie met het attribuut Rationale. Bij het eerst genoemde attribuut wordt kort aangegeven waar het architectuurprincipe aan bijdraagt. Bij Rationale legt men vervolgens uit waarom dit bijdraagt. Deze twee attributen zouden ook samengevoegd kunnen worden. Lees technisch is het misschien gemakkelijker om deze scheiding te behouden omdat je zo snel kunt scannen waar het aan bijdraagt en indien nodig het waarom nog later gelezen kan worden.

Kwaliteitseisen (1,2,3)

De (kwaliteit)eisen (zie ISO 9126) die worden gesteld door belanghebbenden aan een systeem of oplossing vormen de reden dat met een principe rekening gehouden moet worden of dat een principe geïmplementeerd dient te worden.

Elk kwaliteitsaspect kan worden gezien als een hogere orde principe van een concept.

Opmerking

Een strategisch uitgangspunt kan afgeleid zijn van bijvoorbeeld een kwaliteitseis. Een kwaliteitseis kan ook verwerkt zijn in een strategisch uitgangspunt. Een kwaliteitseis is generiek en dient hier vastgelegd te worden zodat snel inzichtelijk gemaakt kan worden aan welke kwaliteitseisen het architectuurprincipe bijdraagt.

Bedrijfsvoordelen / nadelen (1,2,3)

Het voordeel of nadeel van een principe maakt dat men kan onderbouwen waarom dit principe nodig is of bijdraagt aan het realiseren van de gestelde kwaliteitseisen.

Wijziging

De huidige invulling die gegeven dient te worden volgens bovenstaande omschrijving vertoont te veel overlap met het attribuut Rationale. Op basis hiervan is besloten dat de volgende omschrijving tot minder overlap leidt:

De voor- en nadelen van het toepassen / onderkennen van het architectuurprincipe dienen beschreven te worden. De voordelen die hier opgeschreven dienen te worden zijn de extra voordelen die niet direct af te leiden zijn van de invulling die is gegeven bij de attributen Exploitatie en veranderdoelen en Rationale. De nadelen die hier genoemd worden zijn inherent aan het onderkennen / toepassen van het architectuurprincipe en waar geen handhavingmechanisme tegen op kan werken. Een architectuurprincipe waaruit bijvoorbeeld voortvloeit dat men leverancierafhankelijk wordt, kan als nadeel gezien worden.

Handhavingmechanisme (1,2,3)

Dit is nodig om het principe binnen een bepaalde context altijd te kunnen af te dwingen. Voorbeelden van handhavingmechanismen voor principes zijn: toezicht, controle, sociale druk, beloning, bestraffing en dwingende structuren.

Opmerking

Het betreft hier de externe handhaving.

Implementatie impact (1,2,3)

Wat dient er te worden aangepast om een principe waar te laten zijn in een context. Hoeveel tijd, geld en middelen kost het principe om (qua handhaving en structuurwijziging) te implementeren?

Opmerking

Op basis van de invulling die o.a. gegeven is bij de voorgaande attributen kan de implementatie impact beschreven worden. De tijd en kosten die hier aan verbonden zijn dienen tevens vermeld te worden. Dit laatste is noodzakelijk om de keuze te maken of een architectuurprincipe onderkend dient te worden.

Contextual artifacts (1,2,3)

Voorbeelden van onderdelen in de contexten waar een principe voor kan gelden zijn: systeem, systeemonderdeel, organisatie, onderneming, instelling, organisatiedomein, project, proces, product

Opmerking

Hier dienen niet alleen de bovenstaande begrippen vermeld te worden, maar dient men specifiek te zijn (als dit mogelijk is) en dus ook het soort product, de naam van het product etc. indien het bijvoorbeeld om een product gaat.

12.2 De relaties en volgorde van de attributen

De volgorde waarin invulling gegeven dient te worden aan de diverse attributen dient een logisch verhaal te zijn. Daarnaast is het zo dat de mate van detail en op welk niveau invulling gegeven dient te worden aan een attribuut afhankelijk kan zijn van de invulling van andere attributen. Het op een correcte wijze invulling geven aan attributen is afhankelijk van de invulling die is gegeven aan de andere attributen. Hieronder volgt een uiteenzetting hoe de volgorde bepaald is en het eindresultaat. In deze volgorde bepaling wordt er vanuit gegaan dat de attributen onderling relaties hebben en de invulling hier dus van afhankelijk is. Vanuit deze hypothese is de volgorde bepaald. Achter de naam staat een cijfer dat de voorlopige stap aangeeft waarin invulling gegeven dient te worden aan een attribuut. Binnen een stap maakt de volgorde waarin de attributen beschreven worden geen verschil. Hier kan later nog een volgorde binnen gekozen worden. Indien de volgorde waarin invulling gegeven wordt aan een attribuut niet van belang is wordt dit weergegeven met een horizontale streep achter de naam van het attribuut.

Een unieke universele ID (-)

De ID die meegegeven wordt, is onafhankelijk van de informatie die een architectuurprincipe bevat. Het is niet meer dan een nummer of code. Daarom kan deze zowel als eerste of als laatste worden toegewezen. De volgorde is niet van belang.

Een unieke universele naam (7)

De naam van een architectuurprincipe dient kort de context en werking van het architectuurprincipe te beschrijven. Deze kan het beste als laatste worden ingevuld aangezien dan aan alle inhoudelijke attributen invulling is gegeven en het beste een passende naam bedacht kan worden.

Systeem, Concept, of Stijl verwijzing (7)

Een principe is altijd onderdeel van een systeem, concept, fenomeen of stijl. Deze dient (dienen) hier benoemt te worden. De invulling hiervan is afhankelijk van de inhoudelijke invulling van het architectuurprincipe en kan daarom pas later in het proces worden ingevuld. Dit attribuut kan als laatste worden ingevuld.

Statement (2)

Het short statement dient als eerste geformuleerd te worden en bij wijze van spreken na elke invulling van een attribuut weer bijgewerkt te worden.

Klasse (3)

De klasse waartoe het architectuurprincipe behoort kan van invloed zijn op de invulling van overige attributen. Is het bijvoorbeeld een constructiegericht architectuurprincipe dan dient de invulling van andere attributen hierop aan te sluiten. De scope wordt als het ware afgebakend. Van de andere kant kan het toewijzen van een klasse ook puur als classificatie gezien worden die achteraf bepaald wordt. Wanneer men dan een database heeft met architectuurprincipes kan men deze bijvoorbeeld onderverdelen in de diverse klassen.

Type / activiteit (3)

Dezelfde uitleg en volgordebepaling als bij Klasse is van toepassing.

Abstractieniveau (7)

Het abstractieniveau is afhankelijk van de wijze waarop het architectuurprincipe geformuleerd is en dus invulling gegeven is aan de diverse attributen. Het abstractieniveau kan pas als laatste worden vastgesteld. Er is wel een relatie met het abstractieniveau en de invulling van de diverse attributen. Als bijvoorbeeld een architectuurprincipe op fysiek niveau is beschreven dan zijn het inzetten van bepaalde technologieën benoemt en kan men dus een gedetailleerdere invulling geven aan attributen zoals Implementatie impact, Bedrijfs voor- en nadelen, Rationale etc.

Soort bedrijfssysteem (3)

Het soort bedrijfssysteem beschrijft de scope / context waar het architectuurprincipe op van toepassing is. Dit kan bepaald worden nadat het short statement is vastgelegd. Bovendien zijn andere attributen afhankelijk van wat hier wordt ingevuld dus dient deze als tweede ingevuld te worden. Als laatste kan eventueel de classificatie nog worden bijgesteld indien blijkt dat het toch op een breder of minder breed gebied van toepassing is.

Soort organisatiedomein (3)

Het soort bedrijfssysteem beschrijft de scope / context waar het architectuurprincipe op van toepassing is. Dezelfde uitleg als bij soort bedrijfssysteem is hier van toepassing.

Organisatiesysteem beschouwingniveau (3)

Het organisatiesysteem beschouwingniveau beschrijft scope / context waar het architectuurprincipe op van toepassing is. Dezelfde uitleg als bij soort bedrijfssysteem is hier van toepassing. Een toevoeging hierbij is echter dat men ook zelf kan bepalen op welke beschouwingniveau een principe van toepassing is. Er kan bijvoorbeeld duidelijk gekozen worden om een principe niet bedrijfsbreed te hanteren maar alleen binnen een bepaald deel van de organisatie. Indien dit het geval is dient dit al in het beginstadium bepaald te worden zodat er geen onnodige details worden verwerkt in de overige attributen die buiten het beschouwinggebied vallen.

Eigenaar en Beheerder (-)

De eigenaar en beheerder van het architectuurprincipe kan als laatste worden vastgelegd, maar kan ook al van te voren bekend zijn. De invulling van dit attribuut heeft geen relatie tot de overige attributen.

Issues en conflicten (4)

Dit attribuut vormt o.a. de input voor het attribuut Handhavingmechanisme, Implementatie impact en Bedrijfsvoordelen / nadelen. Deze kan pas bepaald worden als de attributen uit stap 3 zijn ingevuld.

Alternatieve en conflicterende architectuurprincipes (4)

Dit attribuut levert o.a. input voor het attribuut Handhavingmechanisme, Implementatie impact en Bedrijfsvoordelen / nadelen. Deze kan pas bepaald worden als de attributen uit stap 3 zijn ingevuld, maar dient bekend te zijn voordat de eerder genoemde attributen ingevuld kunnen worden.

Beschouwings situatie (3)

De invulling hiervan wordt bepaald door degenen die beslist het architectuurprincipe toe te passen / te onderkennen. De invulling van andere attributen kan hier van afhankelijk zijn. De beschouwings situatie dient na het opstellen van het short statement te worden vastgesteld.

Effect –type (3)

De invulling hiervan is afhankelijk van de invulling die gegeven wordt aan het short statement. Op basis hiervan moet men beoordelen om wat voor type het gaat. De invulling van andere attributen kan hier van afhankelijk zijn. Dit attribuut heeft een bijzondere relatie met het attribuut Handhavingmechanisme. Indien er geen extern handhavingmechanisme nodig is dan is het een true-principle. Indien er een extern handhavingmechanisme nodig is dan is het een false-principle.

First principle (7)

Dit is of een ja / nee vraag of hier dient het First principle genoteerd te worden. Indien het een ja / nee vraag betreft dan kan dit pas als laatste worden bepaald aangezien dan pas het architectuurprincipe geformuleerd is.

Status (7)

De volgorde lijkt niet van belang maar het is logischer om hier pas invulling aan te geven als bekend is hoe het architectuurprincipe er uit ziet. Bij het formuleren van een architectuurprincipe staat deze standaard op voorstellen.

Bron(-vermelding) (7)

Hier kan pas compleet invulling aangegeven worden indien de inhoud van de inhoudelijke attributen bekend is. Deze dient dus als laatste ingevuld te worden.

Rationale (2 - 7)

Hier kan pas invulling aangegeven worden indien de inhoud van het attribuut Exploitatie en veranderdoelen is

vastgesteld. De rationale kan meteen worden ingevuld nadat het attribuut Exploitatie en veranderdoelen is ingevuld, dit kan echter ook in een later stadium.

Exploitatie en veranderdoelen (1)

De invulling van dit attribuut vormt de aanleiding van het willen toepassen van een architectuurprincipe. Men gaat een architectuurprincipe formuleren die deze gaat ondersteunen / verwezenlijken. Deze staan van tevoren al vast en kan daarom meteen ingevuld worden.

Kwaliteitseisen (2 - 7)

Hier kan pas invulling aangegeven worden indien de inhoud van het attribuut Exploitatie en veranderdoelen is vastgesteld. De kwaliteitseisen kunnen meteen worden vastgelegd nadat het attribuut Exploitatie en veranderdoelen is ingevuld, dit kan echter ook in een later stadium. Dit laatste is een betere optie aangezien het geformuleerde architectuurprincipe meer raakvlakken kan hebben met bepaalde kwaliteitseisen dan vooraf vastgesteld.

Bedrijfsvoordelen / nadelen (5)

Dit attribuut kan pas ingevuld worden nadat alle inhoudelijke attributen van het architectuurprincipe zijn vastgelegd.

Handhavingmechanisme (5)

Dit attribuut kan pas ingevuld worden nadat alle inhoudelijke attributen van het architectuurprincipe die voor stap 5 zitten zijn vastgelegd. Het handhavingmechanisme is eigenlijk een attribuut wat voortvloeit uit hetgeen wat is opgeschreven bij Issues en Conflicten. Daarnaast is het attribuut optioneel. Indien het een true-principle betreft is een handhavingmechanisme niet noodzakelijk.

Implementatie impact (6)

Dit attribuut kan pas ingevuld worden nadat alle inhoudelijke attributen van het architectuurprincipe zijn vastgelegd en het handhavingmechanisme is beschreven. Nu alle inhoudelijke informatie bekend is kan de impact van het principe worden bepaald.

Contextual artifacts (7)

Dit attribuut kan pas ingevuld worden nadat alle inhoudelijke attributen van het architectuurprincipe die voor stap 7 zitten zijn vastgelegd.

12.3 Resultaat analyse attributen

Op basis van de analyse in hoofdstuk 12.1 en 12.2 is de onderstaande geordende lijst van attributen het resultaat. De benaming van sommige attributen komt niet meer geheel overeen met de nieuwe inhoud die aan hun is toegekend. Daarom worden ook de nieuwe namen weergegeven die gehanteerd zullen worden in het vervolg (indien de naam niet gewijzigd is staat hier de originele naam). Achter de nieuwe naam is de originele naam te vinden. Bij stap staat de volgordebepaling.

Nieuwe naam	Originele naam	Stap
Exploitatie en veranderdoelen	Exploitatie en veranderdoelen	1
Statement	Statement	2
Rationale	Rationale	2- 7
Kwaliteitseisen	Kwaliteitseisen	2 - 7
Klasse	Klasse	3
Soort principe	Type / activiteit	3
Soort bedrijfssysteem	Soort bedrijfssysteem	3
Soort organisatiedomein	Soort organisatiedomein	3
Organisatiesysteem beschouwingniveau	Organisatiesysteem beschouwingniveau	3
Beschouwings situatie	Beschouwings situatie	3
Effect –type	Effect –type	3
Issues en conflicten	Issues en conflicten	4
Alternatieve en conflicterende architectuurprincipes	Alternatieve en conflicterende architectuurprincipes	4
Bedrijfsvoordelen / nadelen	Bedrijfsvoordelen / nadelen	5
Extern handhavingmechanisme	Handhavingmechanisme	5
Implementatie impact	Implementatie impact	6
Contextual artifacts	Contextual artifacts	7
Systeem, Concept, of Stijl verwijzing	Systeem, Concept, of Stijl verwijzing	7
Bron(-vermelding)	Bron(-vermelding)	7
First principle	First principle	7
Abstractieniveau	Abstractieniveau	7
Status	Status	7
Naam	Een unieke universele naam	7
ID	Een unieke universele ID	-
Eigenaar en Beheerder	Eigenaar en Beheerder	-

Het onderstaande overzicht geeft aan wanneer attributen ingevuld kunnen worden en wanneer deze nog een wijziging kunnen ondergaan, die invloed kan hebben op de invulling van de overige attributen. Het onderstaande overzicht dient gebruikt te worden om het stappenplan op de juiste momenten de benodigde informatie te laten vragen zodat er zo efficiënt mogelijk een architectuurprincipe beschreven kan worden die de juiste informatie bevat die aansluit bij de context waarin het architectuurprincipe kan worden toegepast. Het betreft hier natuurlijk de ideale situatie. Het beschrijft de ideale volgorde. Een stappenplan zou deze volgorde zoveel mogelijk moeten afdwingen.

Attribuutnaam	Stap 1	Stap 2	Stap 3	Stap 4	Stap 5	Stap 6	Stap 7
Exploitatie en veranderdoelen							
Statement							
Rationale							
Kwaliteitseisen							
Klasse							
Soort principe							
Soort bedrijfssysteem							
Soort organisatiedomein							
Organisatiesysteem beschouwingniveau							
Beschouwingssituatie							
Effect –type							
Issues en conflicten							
Alternatieve en conflicterende architectuurprincipes							
Bedrijfsvoordelen / nadelen							
Extern handhavingmechanisme							
Implementatie impact							
Contextual artifacts							
Systeem, Concept, of Stijl verwijzing							
Bron(-vermelding)							
First principle							
Abstractieniveau							
Status							
Naam							
ID							
Eigenaar en Beheerder							

Het bovenstaande overzicht dient als volgt gelezen te worden:

- Een donkergroen blokje geeft de gewenste stap aan waarin invulling gegeven dient te worden aan een bepaald attribuut. Indien er twee donkergroene blokjes aanwezig zijn dient er de eerste keer invulling gegeven te worden aan een attribuut en de tweede keer een eventuele aanvulling gedaan te worden.
- Een lichtgroen blokje geeft aan dat het mogelijk is om op dat moment invulling te geven aan een attribuut of de ingevulde informatie te wijzigen.

Speciale aandacht verdienen de ‘classificatie’ attributen Klasse, Soort principe, Soort bedrijfssysteem, Soort organisatiedomein en Organisatie beschouwingniveau. De classificatie die nu is gebruikt, is afkomstig van de Dragon1 methode. Het aantal ‘classificatie’ attributen is uit te breiden indien men dit noodzakelijk vindt. Men kan hiermee de context waarop het architectuurprincipe van toepassing is beperken. De ‘classificatie’ attributen kunnen afgestemd worden op de organisatie of op een andere EA methode. Indien men van mening is dat andere soorten ‘classificatie’ attributen beter aansluiten dan vormt dit geen probleem.

12.4 Kandidaat attributen

Op basis van het onderzoek dat tot nu toe is gedaan, worden een aantal nieuwe attributen ter kandidaat gesteld. Er zal per nieuw attribuut beschreven waarom deze een meerwaarde heeft en hoe de inhoud bepaald kan worden, en welke relaties het heeft en in welke volgorde het ingevuld dient te worden.

12.4.1 Alternatieve representatiewijze

Er wordt op het gebied van EA onderzoek gedaan naar alternatieve representatiewijze van een architectuurprincipe. Voorbeelden hiervan zijn het visualisatieonderzoek van Mark Paauwe en Daan Rijsenbrij [Digitale architectuur in beeld brengen en communiceren op basis van het Paauwe-Rijsenbrij Visualisatieschema] en het onderzoek naar het formaliseren van architectuurprincipes door P. van Bommel, S.J.B.A. Hoppenbrouwers, H.A. Proper, and Th.P. van der Weide [Giving Meaning to Enterprise Architectures - Architecture Principles with ORM and ORC].

Het huidige stappenplan leidt tot een architectuurprincipe beschreven in natuurlijke taal. Deze vorm is altijd gewenst, maar kan worden versterkt door een alternatieve representatiewijze. Dit attribuut dient de alternatieve representatiewijze te bevatten. Het attribuut is optioneel aangezien niet voor elk architectuurprincipe een alternatieve representatiewijze zal worden gemaakt.

Dit attribuut kan dus ook een verzameling van verschillende alternatieve representatiewijzen bevatten. Bij voorbeeld een ORM-model om een bepaald aspect van het architectuurprincipe specifiek te belichten of een visualisatie die een bepaalde doelgroep op een snelle manier het architectuurprincipe inzichtelijk maakt.

Op deze manier kan een architectuurprincipe specifiek zijn, maar toch gemakkelijker communiceerbaar. Men kan de voor de doelgroep belangrijke informatie samenvatten in deze alternatieve representatiewijze zodat ze gemakkelijk hun oordeel kunnen vestigen of ze het architectuurprincipe willen gaan hanteren binnen hun organisatie. Wat ook mogelijk is om alleen de informatie die interessant is voor de doelgroep in tekstuele vorm samen te vatten en deze als alternatieve representatiewijze aan te bieden. Op deze wijze wordt de belangrijkste informatie beter communiceerbaar en is de onderbouwing altijd te vinden in de complete beschrijving van het architectuurprincipe.

Het invulling geven aan dit attribuut is mogelijk nadat alle attributen zijn ingevuld. Dit attribuut is optioneel en inzetbaar bij de drie eerder genoemde mogelijke stappenplannen.

12.4.2 Principeoplossingen / aanpakken

In het shortstatement van het architectuurprincipe wordt al aandacht besteed aan de principeoplossingen die gekozen zijn om het gewenste resultaat te verwezenlijken. Echter niet alle informatie kan worden weergegeven in het shortstatement, dat een samenvatting betreft van het architectuurprincipe, en daarom dienen de gekozen principeoplossingen in een apart attribuut vernoemd en beschreven te worden.

De invulling die gegeven wordt aan dit attribuut vormt een deel van de onderbouwing van het architectuurprincipe.

In dit attribuut dienen de gekozen principeoplossingen (aanpakken) beschreven te worden. Hiervan dient de naam en omschrijving vastgelegd te worden en waarom voor deze oplossing / aanpak is gekozen.

De inhoud van dit attribuut vormt de basis van deel 1 van het short statement van een architectuurprincipe. Het hoe wordt hiermee beschreven. De inhoud van dit attribuut is vooral interessant voor degene die de architectuurprincipes moet gaan doorvertalen naar regels en voor degene die de principeoplossingen moet gaan verwezenlijken.

Dit attribuut heeft een relatie met het short statement waar de principeoplossingen en aanpakken eigenlijk al gekozen worden en kort beschreven worden. Daarnaast dient deze beschreven te zijn voordat het attribuut Validiteit ingevuld kan worden. Ook de attributen die na stap 3 komen kunnen input verkrijgen van dit attribuut. Dit attribuut dient daarom bij stap 3 ingevuld te worden. Dit attribuut is inzetbaar bij de drie eerder genoemde mogelijke stappenplannen.

12.4.3 Validiteit

Het attribuut validiteit gaat in op de mate waarin een architectuurprincipe een bewezen onderbouwing heeft die garant staat voor het bereiken van het gewenste resultaat. Binnen organisaties, en dan vooral bij de IT-gerelateerde gebieden, wordt veel gebruik gemaakt van nieuwe technologieën, methoden, best practices etc. Het gebruiken van deze nieuwe, nog onbekende oplossingen kan leiden tot ongewenste resultaten. Door volledig te vertrouwen op deze oplossingen kan men aardig de mist in gaan en worden belangrijke doelstellingen en uitgangspunten niet gerealiseerd. Het is daarom verstandig om inzichtelijk te maken in welke mate een architectuurprincipe wordt onderbouwd met bewezen oplossingen of juist onbewezen oplossingen. Het is daarom de bedoeling om per gekozen oplossing de mate waarin deze bewezen is te beschrijven. Op basis hiervan kan men inzien welk risico men neemt door het architectuurprincipe te onderkennen. Het streven is natuurlijk om bij dit attribuut een 100% score te behalen, wat in de praktijk bijna onhaalbaar is door de vele factoren die invloed hebben op het succesvol inzetten van een oplossing. Dit attribuut vormt een soort waakhond voor de fout een doelstelling of uitgangspunt puur proberen te verwezenlijken door een architectuurprincipe waarvan niet aannemelijk gemaakt kan worden dat deze zal gaan werken als men deze gaat toepassen. Een andere interpretatie die gegeven kan worden aan dit attribuut is dat het de kans van falen van het architectuurprincipe beschrijft.

Dit attribuut is een verfijning en uitbreiding van het attribuut Effect-type en beschrijft eigenlijk de mate waarin de interne handhaving 'true' of 'false' is.

Aan dit attribuut kan pas invulling gegeven worden indien het attribuut Principeoplossingen / aanpakken is beschreven. Het attribuut kan ook pas ingevuld worden nadat de inhoud van het attribuut Alternatieve en conflicterende architectuurprincipes is bepaald. Een conflicterend architectuurprincipe kan immers de validiteit aan tasten van een architectuurprincipe. De attributen Issues en Conflicten en Extern handhavingmechanisme kunnen pas ingevuld worden indien de validiteit vast staat.

De conclusie is dat Validiteit het attribuut Effect-type dient te vervangen. De volgorde waarin dit attribuut beschreven dient te worden wijkt echter af van het oorspronkelijke attribuut Effect-type. Dit attribuut is inzetbaar bij de drie eerder genoemde mogelijke stappenplannen.

12.5 Overzicht attributen en volgorde inclusief kandidaat attributen

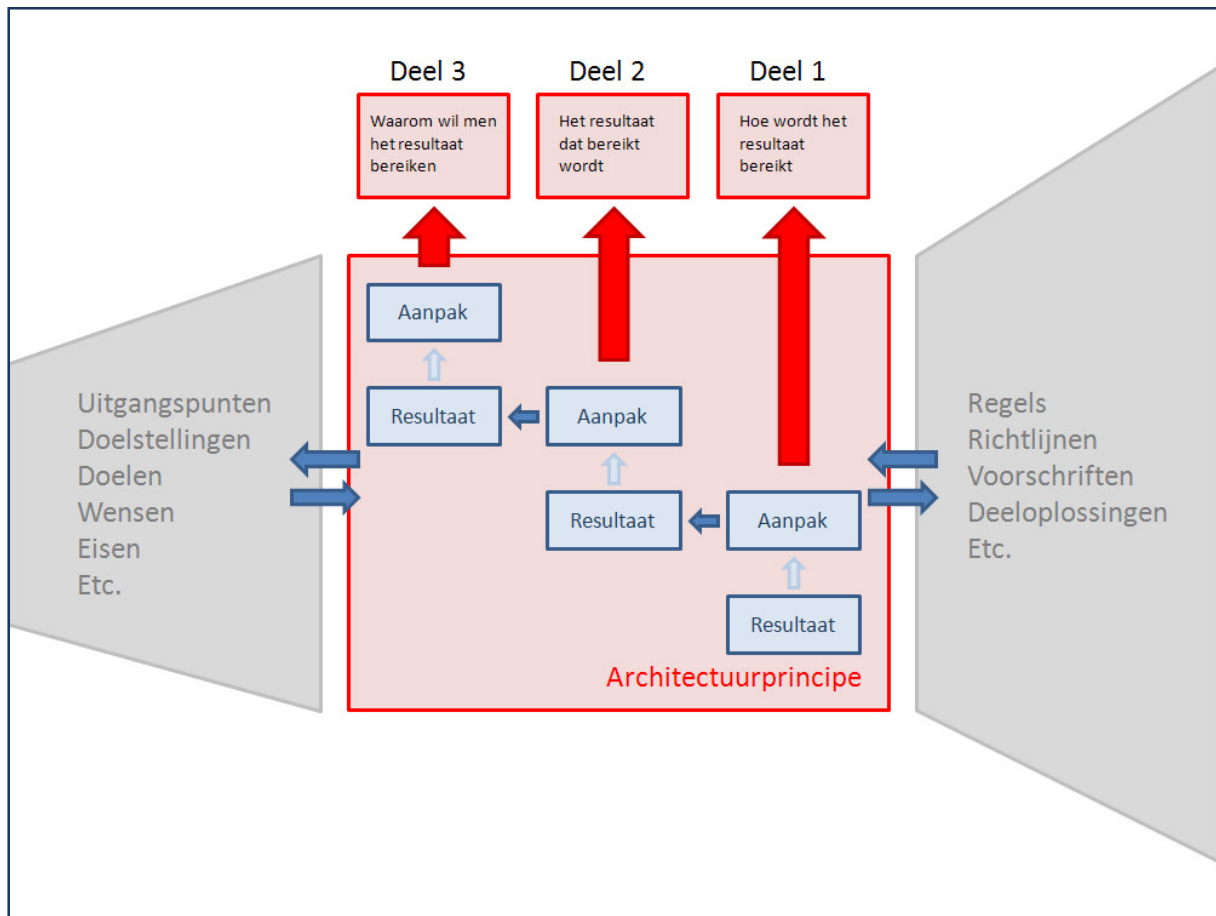
In onderstaand overzicht zijn de kandidaat attributen verwerkt:

Attribuutnaam	1	2	3	4	5	6	7	8	9	10
Exploitatie en veranderdoelen	■									
Statement		■	■	■	■	■	■	■	■	
Rationale		■	■	■	■	■	■	■	■	
Kwaliteitseisen		■	■	■	■	■	■	■	■	
Klasse			■							
Soort principe			■							
Soort bedrijfssysteem			■							
Soort organisatiedomein			■							
Organisatiesysteem beschouwingniveau			■							
Beschouwingssituatie			■							
Principeoplossingen / aanpakken			■							
Alternatieve en conflicterende architectuurprincipes				■						
Validiteit					■					
Issues en conflicten						■				
Bedrijfsvoordelen / nadelen							■			
Extern handhavingmechanisme							■			
Implementatie impact								■		
Contextual artifacts									■	
Systeem, Concept, of Stijl verwijzing									■	
Bron(-vermelding)									■	
First principle									■	
Abstractieniveau									■	
Status									■	
Naam									■	
ID	■	■	■	■	■	■	■	■	■	
Eigenaar en Beheerder	■	■	■	■	■	■	■	■	■	
Alternatieve representatiewijze										■

Het bovenstaande overzicht geeft de ideale volgorde weer waarin invulling gegeven dient te worden aan een architectuurprincipe zodat de juiste informatie in voldoende mate belicht wordt.

12.6 De context van het architectuurprincipe

Op basis van de resultaten tot nu toe geeft de volgende visualisatie een goed beeld van waar het architectuurprincipe zich bevindt in het grotere geheel bestaande uit regels, uitgangspunten, richtlijnen etc.

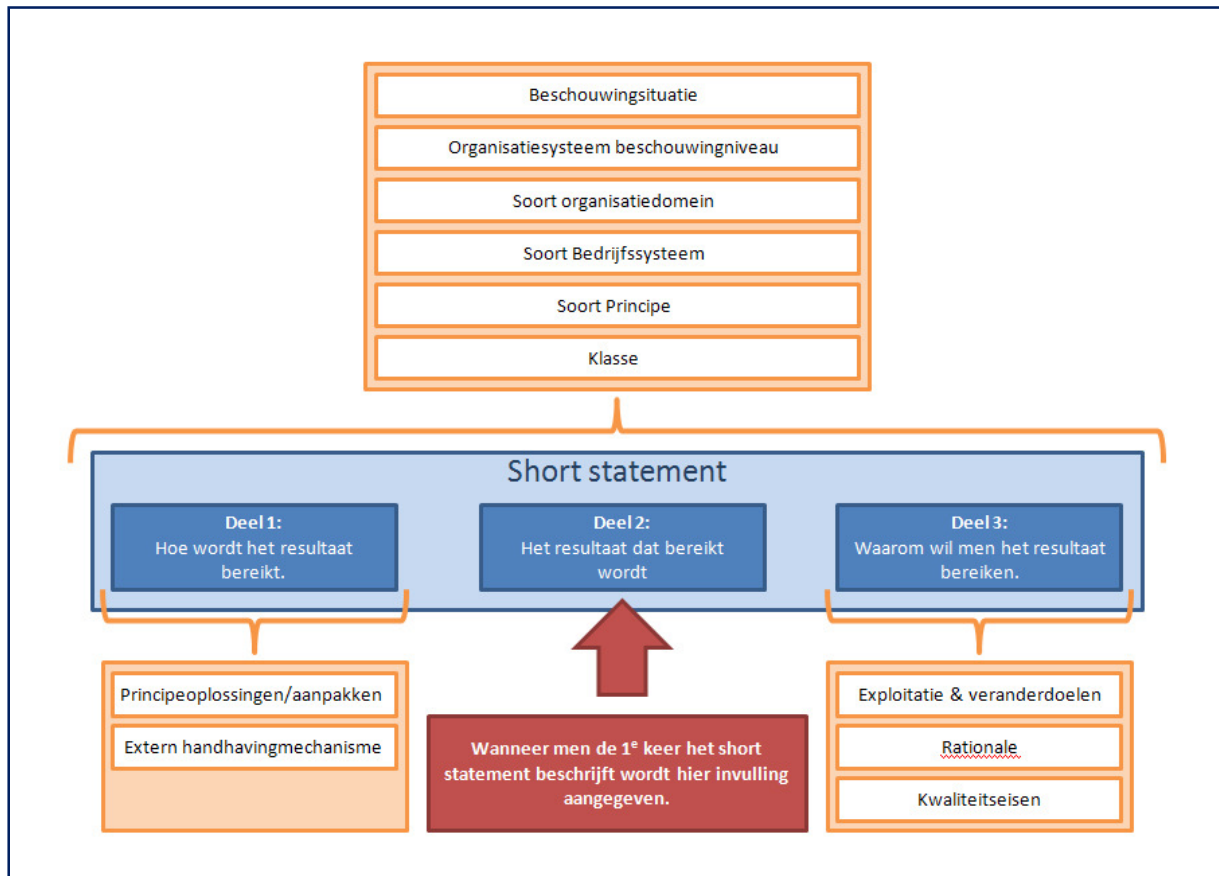


Figuur 8: De context van het architectuurprincipe

Zoals een architectuurprincipe op bovenstaande wijze gedefinieerd en gepositioneerd staat, is duidelijk te zien welke functie een architectuurprincipe kan vervullen; een mechanisme dat de vertaling van strategie naar regels, richtlijnen etc. in goede banen dient te leiden. Hierbij is de informatie die deel 3 bevat afgeleid van de strategie of hogere orde doelstellingen of principes. De informatie in deel 1 bevat de informatie die noodzakelijk is voor de vertaling naar de operationele kant.

12.7 Relatie attributen met het short statement

Het short statement van het architectuurprincipe verdient speciale aandacht aangezien deze tijdens het formuleren van het architectuurprincipe telkens wordt aangescherpt en uiteindelijk de kernsamenvatting vormt van het architectuurprincipe. De overige attributen bepalen welke invulling gegeven dient te worden aan het short statement. Op basis van deze functie dient het short statement dan ook volledig te zijn en dient de belangrijkste informatie van diverse attributen erin terug te vinden zijn. Om een beter inzicht te verkrijgen welk deel van het short statement welke informatie dient te bevatten, zal de relatie tussen de attributen en het short statement beschreven worden. Op basis van dit resultaat kan duidelijk beschreven worden welke informatie het short statement dient te bevatten en dus behoort tot de kern van het architectuurprincipe. De onderstaande afbeelding geeft aan welke attributen invulling geven aan welk deel van het short statement.



Figuur 9: Shortstatement

De attributen beschouwingssituatie, organisatiesysteem beschouwingniveau, soort organisatiedomein, soort bedrijfssysteem, soort principe en klasse hebben allen betrekking op de gehele beschrijving van het short statement. Deze attributen bakenen de scope af en bepalen het detailniveau van de beschrijving. Het eerste deel wordt vooral beschreven door de gekozen oplossingen en het externe handhavingmechanisme. Deze twee attributen samen beschrijven de manier waarop het resultaat uit deel 2 verwezenlijkt zal worden. Deel 2 bevat een (concrete) deeloplossing/resultaat die de schakel vormt tussen de operationele en strategische kant. Dit deel wordt dus niet afgevangen door een attribuut, maar dient als eerste te worden opgeschreven wanneer men het short statement gaat beschrijven. Dit deel wordt in de praktijk vaak beschreven en gezien als een architectuurprincipe. Deel 3 bevat de belangrijkste informatie die afkomstig is van de attributen exploitatie en veranderdoelen, rationale en kwaliteitseisen.

12.8 Attribuut functiesoorten

De resultaten van hoofdstuk 12.1 t/m 12.7 hebben inzichtelijk gemaakt dat de attributen van een architectuurprincipe verschillende soorten functies hebben. Op basis van deze functie behoort een attribuut tot een bepaalde functiesoort. De volgende functiesoorten zijn te definiëren: Communicatie, Motivatie, Bedreiging, Handhaving, Context, Verwijzing, Beheer. Deze zullen in het volgende overzicht nader worden toegelicht. Het betreft hier een voorstel die nog geverifieerd dient te worden.

Funciesoort	Omschrijving
Communicatie	Attributen die behoren tot de functiesoort Communicatie dienen het communicatieve aspect van het architectuurprincipe te bevorderen.
Motivatie	Attributen die behoren tot de functiesoort Motivatie bevatten informatie over waarom men het architectuurprincipe wil gaan toepassen.
Bedreiging	Attributen die behoren tot de functiesoort Bedreiging beschrijven de valkuilen, risico's, bedreigingen etc. die het willen toepassen van een architectuurprincipe of het succesvol toepassen van een architectuurprincipe in de weg kunnen staan.
Handhaving	Attributen die behoren tot de functiesoort Handhaving beschrijven de maatregelen die genomen dienen te worden om de bedreigingen te minimaliseren en het succesvol toepassen van een architectuurprincipe mogelijk maken.
Context	Attributen die behoren tot de functiesoort Context dragen bij aan het afbakenen van de context waartoe het architectuurprincipe betrekking heeft en de mate van detail die de beschrijving van het architectuurprincipe heeft.
Verwijzing	Attributen die behoren tot de functiesoort Verwijzing bevatten achtergrond informatie, verwijzingen, referenties etc. of gaan specifieker in op bepaalde zaken die eerder zijn beschreven in andere attributen van het architectuurprincipe.
Beheer	Attributen die behoren tot de functiesoort Beheer bevatten informatie die noodzakelijk is voor het beheer en onderhoud van het architectuurprincipe.

Op basis van dit voorstel voor de verdeling van de attributen in functiesoorten is de volgende classificatie van toepassing:

Funciesoort	Attribuut
Communicatie	<ul style="list-style-type: none"> • Naam • Statement (short statement) • Alternatieve representatiewijze
Motivatie	<ul style="list-style-type: none"> • Exploitatie en veranderdoelen • Rationale • Kwaliteitseisen
Bedreiging	<ul style="list-style-type: none"> • Alternatieve en conflicterende architectuurprincipes • Issues en conflicten • Bedrijfsvoordelen / nadelen • Validiteit
Handhaving	<ul style="list-style-type: none"> • Principeoplossingen / aanpakken • Extern handhavingmechanisme • Implementatie impact
Context	<ul style="list-style-type: none"> • Klasse • Soort principe • Soort bedrijfssysteem • Soort organisatiedomein • Organisatiesysteem beschouwingniveau • Beschouwingssituatie
Verwijzing	<ul style="list-style-type: none"> • Contextual artifacts • Systeem, Concept, of Stijl verwijzing • Bron(-vermelding) • First principle

	<ul style="list-style-type: none"> • Abstractieniveau
Beheer	<ul style="list-style-type: none"> • ID • Status • Eigenaar en Beheerder

12.9 Overzicht uiteindelijke attributen

Op basis van de resultaten en constatering uit hoofdstuk 11 en hoofdstuk 12 wordt hieronder een overzicht gepresenteerd van de attributen van een architectuurprincipe. Per attribuut wordt de naam weergegeven en de inhoud die het attribuut dient te bevatten.

Attribuut	Beschrijving	Stap	Functie
1. Exploitatie en veranderdoelen	<p>Welke doelen liggen ten grondslag aan de keuze en het opstellen van het (concept van het) principe?</p> <p>Hier dienen strategische uitgangspunten of (bedrijfs)doelstellingen vermeld te worden. Hierdoor wordt inzichtelijk gemaakt welke strategische uitgangspunten of doelstellingen (deels) worden verwezenlijkt door dit architectuurprincipe. Indien deze strategische uitgangspunten of doelstellingen niet voorhanden zijn kunnen hier ook algemene kerndoelen beschreven worden zoals kostenreductie, verlagen complexiteit, snelle service etc.</p> <p>Door hier invulling aan te geven wordt de relatie tussen het architectuurprincipe en de strategie & visie van een onderneming vastgelegd waardoor het belang van het hanteren van het architectuurprincipe onderstreept wordt.</p>	1	Motivatie
2. (short) Statement	<p>De formulering van het statement (de omschrijving van een architectuurprincipe) komt nogal nauw. De context waarbinnen het architectuurprincipe geldt en de periode en het abstractieniveau waarin het architectuurprincipe geldig is, dient duidelijk te worden afgebakend.</p> <p>Omdat een architectuurprincipe een waarheid betreft, zijn woorden zoals mogen, willen, kunnen, zouden en moeten niet gewenst in de naam of omschrijving. De formulering van een architectuurprincipe dient stellend te zijn, maar ook een werking te beschrijven. De formulering dient ultimo een oorzaakgevolg relatie te beschrijven maar dient minimaal een volledige entiteitenrelatie te beschrijven.</p> <p>De omschrijving van een principe gaat in op het mechanisme of de werking van een concept, systeem, fenomeen of stijl.</p> <p>De omschrijving van een architectuurprincipe moet zoveel mogelijk S.M.A.R.T. zijn. Dit wordt bereikt door het inhoudelijke invulling geven aan het architectuurprincipe en het aanscherpen van het short statement na iedere grove wijziging van de inhoud van de overige attributen.</p> <p>Het statement beschrijft de kern van het architectuurprincipe en vormt tevens de basis van het architectuurprincipe. Dit statement dient in een later stadium nog te worden aangepast / bijgewerkt om overeen te komen met de inhoudelijke informatie die is</p>	2-9	Communicatie

	<p>toegekend aan de overige attributen.</p> <p>Het statement van een architectuurprincipe bestaat uit drie delen:</p> <p>Deel 1: Beschrijft hoe het gewenste resultaat wordt bereikt. Deel 2: Beschrijft welk resultaat bereikt dient te worden. Deel 3: Beschrijft waarom men het resultaat wil bereiken (wat wordt hierdoor mogelijk of onmogelijk.)</p> <p>Het statement dient als volgt beschreven te worden:</p> <p><i>'door ..[(een) bepaalde principeoplossing(en) / aanpak(ken) en eventueel een extern handhavingmechanisme].. zal, of wordt ..[een bepaalde gewenste situatie bereikt of probleem opgelost].. met als resultaat dat ..[een bepaald opgeleverd resultaat bereikt zal worden]..'</i></p>		
3. Rationale	<p>De achterliggende reden (motivatie) voor het onderkennen van het architectuurprincipe dient hier beschreven te worden.</p> <p>Waarom draagt het onderkennen van het architectuurprincipe bij aan bijvoorbeeld het verwezenlijken van de eerder beschreven exploitatie en veranderdoelen of strategische uitgangspunten.</p>	2-9	Motivatie
4. Kwaliteitseisen	<p>De (kwaliteit)eisen (zie ISO 9126) die worden gesteld door belanghebbenden aan een systeem of oplossing vormen de reden dat met een architectuurprincipe rekening gehouden moet worden of dat een architectuurprincipe gehanteerd dient te worden.</p> <p>Elk kwaliteitsaspect kan worden gezien als een hogere orde principe van een concept. Ook een strategisch uitgangspunt kan afgeleid zijn van bijvoorbeeld een kwaliteitseis. Een kwaliteitseis kan ook verwerkt zijn in een strategisch uitgangspunt. Een kwaliteitseis is generiek en dient hier vastgelegd te worden zodat snel inzichtelijk gemaakt kan worden aan welke kwaliteitseisen het architectuurprincipe bijdraagt.</p> <p>De volgende ISO 9126 kwaliteitseisen zijn van toepassing:</p> <p>Functionality (Suitability, Accurateness, Interoperability, Compliance, Security) Reliability (Maturity, Fault tolerance, Recoverability) Usability (Understandability, Learnability, Operability) Efficiency (Time behavior, Resource behavior) Maintainability (Analyzability, Changeability, Stability, Testability) Portability (Adaptability, Installability, Conformance, Replaceability)</p>	2-9	Motivatie
5. Klasse	<p>Is het een kringgedrag principe, taak/volgorde principe of actie-reactie principe?</p>	3	Context
6. Soort principe	<p>Is het een Constructie principe, Structuur principe, Ontwerp principe, Bouw/Realisatie principe, Systeem principe, Fenomeen principe, Concept principe, Domein principe, Solution principe,</p>	3	Context

	<p>(stijl)Element principe, Artifact principe, Keten principe, Enterprise principe, Besturings principe, Bedrijfs(functie) principe, Informatie(voorziening) principe, IT-Infrastructuur principe, Security principe.</p> <p>De soorten principes die hier genoemd worden zijn veelvoorkomende soorten die in de praktijk onderkend worden. Indien binnen de onderneming andere soorten principes of andere benamingen worden gebruikt kunnen deze gebruikt worden.</p>		
7. Soort bedrijfssysteem	Is het een inkoop, verkoop, marketingcommunicatie, productie, financiën, ICT (IV, ICT-Infra)?	3	Context
8. Soort organisatie domein	<p>Is het bijvoorbeeld een consumentenmarkt principe, een back office principes, een winkelformuleprincipe of een assortiments-principe?</p> <p>Hanteer hier begrippen die binnen de onderneming onderkend worden zodat men 'feeling' krijgt en begrijpt waar dit architectuurprincipe betrekking op heeft.</p>	3	Context
9. Organisatie systeem beschouwing niveau	Is het een ondernemingsprincipe, bedrijfsprincipe, bedrijfsfunctieprincipe, bedrijfsproces, bedrijfstaak, handeling?	3	Context
10. Beschouwing situatie	<p>We maken onderscheid in situatietypen principes: AS-IS principes, Plateau n - principes, TO-BE principes, Envision principes en Timeless principes.</p> <p>Een AS-IS principe is een principe dat zich in de huidige situatie (heden - 1 jaar) daadwerkelijk voordoet, waarmee men er verstandig aan doet er rekening mee te houden.</p> <p>Een plateau n principe is een principe dat zich tijdelijk voordoet (jaar X).</p> <p>Een TO-BE principe is een principe dat zich in de toekomstige situatie (3-5 jaar) zal gaan voordoet, realistisch en haalbaar is, waarmee men er verstandig aan doet er in de toekomst rekening mee te houden.</p> <p>Een Envision principe is een principe dat zich in de verre toekomst zal gaan voordoet, maar voor de AS-IS, Plateau1 en TO-BE situatie onrealistisch is.</p> <p>Een Timeless principe is een principe dat altijd van toepassing is en dus tijd onafhankelijk.</p>	3	Context
11. Principeoplossing en / aanpakken	Hier dienen de gekozen principeoplossingen (aanpakken) beschreven te worden. Hiervan dient de naam en omschrijving vastgelegd te worden en waarom voor deze oplossing / aanpak is gekozen.	3	Handhaving

12. Alternatieve en conflicterende architectuurprincipes	Hier dient vermeld te worden met welke andere architectuurprincipes het architectuurprincipe in conflict is en welke alternatieve soortgelijke architectuurprincipes er zijn. Geef hier ook aan waarom juist voor dit architectuurprincipe / concept is gekozen en waarom deze 'beter' is dan de alternatieve architectuurprincipes.	4	Bedreiging
13. Validiteit	Hier dient de mate waarin een architectuurprincipe een bewezen onderbouwing heeft die garant staat voor het bereiken van het gewenste resultaat inzichtelijk gemaakt te worden. Geef hier in ieder geval per gekozen oplossing aan in hoeverre deze succesvol is gebleken in het verleden of bewezen is door onderzoek.	5	Bedreiging
14. Issues en conflicten	Dit attribuut beschrijft de interne issues en conflicten die een architectuurprincipe met zich mee (zal gaan) brengen binnen de context, bijvoorbeeld een organisatie. Wat zijn haken en ogen aan het principe om het in de organisatie te hanteren? Waar conflicteert dit principe met andere principes, uitgangspunten, regels binnen de organisatie?	6	Bedreiging
15. Bedrijfsvoordelen / nadelen	De voor- en nadelen van het toepassen / onderkennen van het architectuurprincipe dienen beschreven te worden. De voordelen die hier opgeschreven dienen te worden zijn de extra voordelen die niet direct af te leiden zijn van de invulling die is gegeven bij de attributen Exploitatie en veranderdoelen en Rationale. De nadelen die hier genoemd worden zijn inherent aan het onderkennen / toepassen van het architectuurprincipe en waar geen handhavingmechanisme tegen op kan werken. Een architectuurprincipe waaruit bijvoorbeeld voortvloeit dat men leverancierafhankelijk wordt, kan als nadeel gezien worden.	7	Bedreiging / Motivatie
16. Extern handhavingmechanisme	Hier dienen de externe handhavingmechanismen beschreven te worden die noodzakelijk zijn om het architectuurprincipe te laten werken binnen een context. Dit is nodig om het principe binnen een bepaalde context altijd te kunnen af te dwingen indien het een false-principle betreft. Voorbeelden van handhavingmechanismen voor principes zijn: toezicht, controle, sociale druk, beloning, bestraffing en dwingende structuren.	7	Handhaving
17. Implementatie impact	Wat dient er te worden aangepast om een architectuurprincipe waar te laten zijn in een context. Hoeveel tijd, geld en middelen kost het architectuurprincipe om (qua handhaving en structuurwijziging) te verwezenlijken?	8	Handhaving

<p>18. Contextual artifacts</p>	<p>Beschrijf op welke contextuele artifacten het architectuurprincipe betrekking heeft.</p> <p>Voorbeelden van onderdelen in de contexten waar een principe voor kan gelden zijn: systeem, systeemonderdeel, organisatie, onderneming, instelling, organisatiedomein, project, proces, product.</p> <p>Hier dienen niet alleen de bovenstaande begrippen vermeld te worden, maar dient men specifiek te zijn (als dit mogelijk is) en dus ook het soort product, de naam van het product etc. indien het bijvoorbeeld om een product gaat.</p>	<p>9</p>	<p>Verwijzing</p>
<p>19. Systeem, Concept, of Stijl verwijzing</p>	<p>Beschrijf van welke systemen, concepten, fenomenen of stijlen het architectuurprincipe onderdeel is.</p> <p>Een principe is altijd onderdeel van een systeem, concept, fenomeen of stijl.</p> <p>Een systeem is een geheel van onderdelen die samenwerken teneinde een gemeenschappelijk doel te realiseren.</p> <p>Een concept is een gestructureerde, planmatige of systematische werkwijze.</p> <p>Een fenomeen is een systeem dat in de natuur voor komt en dat ongekende krachten in zich heeft en zeer tot de verbeelding spreekt.</p> <p>Een stijl is een verzameling van concepten met gemeenschappelijke (stijl)elementen. We maken onderscheid in systemen, concepten en stijlen voor constructie, ontwikkeling en verandering. Voorbeelden van gangbare stijlen zijn service oriëntatie, component oriëntatie, proces oriëntatie, project oriëntatie, data oriëntatie, event oriëntatie, engine oriëntatie en taakgericht werken. Per stijl is het hoofdstijlelement benoemd.</p>	<p>9</p>	<p>Verwijzing</p>
<p>20. Bron(-vermelding)</p>	<p>Voorbeelden van bronnen van principes zijn: de natuur, ervaring, een door de industrie ontwikkelde technologie, een best practice, een bewezen concept.</p> <p>De bronvermelding voor een principe is nooit een wens, eis, requirement of doelstelling. Het eisen van een bepaalde kwaliteit of een veranderdoel zijn wel aanleidingen om een bepaald concept (of principes) te willen of te moeten implementeren.</p> <p>Hier dient dus ook naast de naam van de bron ook de bronvermelding te staan.</p>	<p>9</p>	<p>Verwijzing</p>
<p>21. First principle</p>	<p>Ieder systeem, stijl, fenomeen of concept heeft een kern (van waarheid, of waar het om draait) die door middel van een principe kan worden geformuleerd. Dit principe is het first principle. Van alle systemen, concepten en stijlen die in een architectuur worden gebruikt dient het first principle te worden geformuleerd.</p>	<p>9</p>	<p>Verwijzing</p>

	Is het architectuurprincipe een First principle? Ja, Nee. Indien nee: schrijf het First principle op dat de basis vormt van het architectuurprincipe		
22. Abstractieniveau	Is het architectuurprincipe beschreven op conceptueel, logisch of fysiek niveau?	9	Verwijzing
23. Status	We maken onderscheid in de volgende statusovergangen met bijbehorende status van een principe: voorstellen, kandideren, in overweging nemen, onderkennen.	9	Beheer
24. Naam	Een kort statement van één tot tien woorden die duidelijk maken wat de context en werking van het principe is. Bijvoorbeeld: het zwaartekracht-principe of het gedigitaliseerde werkprocessenprincipe.	9	Communicatie
25. ID	Vanwege de (miljoenen) principes in een enterprise architectuur dient een principe een unieke identificatie te hebben. Het ID betreft hier een uniek nummer of unieke code.	1-9	Beheer
26. Eigenaar en Beheerder	Wie de eigenaar en wie is de beheerder van het principe?	1-9	Beheer
27. Alternatieve representatie wijze	Hier dient de alternatieve representatiewijze beschreven of getoond te worden. Dit attriboot kan ook een verzameling van verschillende alternatieve representatiewijzen bevatten. Bijvoorbeeld een ORM-model om een bepaald aspect van het architectuurprincipe specifieker te belichten of een visualisatie die bij een bepaalde doelgroep op een snelle manier het architectuurprincipe inzichtelijk maakt.	10	Communicatie

12.10 Conclusie

In hoofdstuk 12 is getracht het stappenplan van Dragon1 voor het formuleren van architectuurprincipes te optimaliseren. Dit is gedaan door kritisch te kijken naar de toegevoegde waarde van de attributen, de invulling die er aan gegeven dient te worden en de onderlinge relaties tussen de attributen. Ook de volgorde waarin de attributen behandeld dienen te worden en de functie die de attributen hebben ten opzichte van het geheel zijn vastgesteld. Op basis van de resultaten van hoofdstuk 11 en 12 kan een stappenplan opgesteld of aangepast gaan worden. De hoofdlijnen zijn nu uitgezet en levert al veel houvast. Een gedetailleerd stappenplan kan echter nog bijdragen aan het fijn afstellen van de attributen en meer inzicht bieden in de precieze wijze waarop de invulling van een attribuut beschreven dient te worden. In dit hoofdstuk is al extra aandacht besteed aan het short statement van een architectuurprincipe en de invulling die hier aangegeven dient te worden. Ook de overige attributen verdienen een dergelijke beschrijving en analyse.

In dit hoofdstuk is het architectuurprincipe gepositioneerd als een mechanisme dat de verbinding vormt tussen strategie en operatie. Het is de vertaalslag van strategie naar operatie en daardoor een krachtig hulpmiddel. Door de functie van een architectuurprincipe op dergelijke wijze te formuleren ontstaat ook veel meer inzicht in wat een architectuurprincipe moet beschrijven, waarom dit zo is en met welke diepgang dit moet gebeuren. De toegevoegde waarde van een architectuurprincipe komt zo aan het licht. Voor veel mensen is een

architectuurprincipe een uitspraak waarvan men de functie maar lastig kan vaststellen en relateren aan begrippen als regels en doelstellingen en uitgangspunten. Doordat het architectuurprincipe de eerder beschreven functie en inhoud mee te geven wordt de meerwaarde helder. Het beschrijven van zowel een aanpak als het resultaat in een principe is noodzakelijk. Laat je één van deze weg dan betreft je uitspraak al snel een regel, oplossing, doelstelling, uitgangspunt of streven en heeft het geen nut hier het label architectuurprincipe op te spelden. Het zorgt dan alleen voor verwarring en onbegrip. Kortom, deze definitie en functie van het architectuurprincipe maakt het een effectief mechanisme dat van toegevoegde waarde is en is te relateren en positioneren ten opzichte van al ingeburgerde termen binnen ondernemingen als regels, doelstellingen, eisen en wensen, uitgangspunten etc. De diverse EA methoden dienen dan ook veel meer aandacht te besteden aan de relatie tussen architectuurprincipes en de eerder genoemde termen. Men moet het architectuurprincipe een duidelijke functie kunnen toewijzen om zo het gebruik er van te kunnen motiveren.

13 Conclusies

In dit hoofdstuk wordt antwoord gegeven op de onderzoeksvragen en worden de belangrijkste conclusies weergegeven.

13.1 Beantwoording deelvragen

Welke eigenschappen hebben architectuurprincipes die in de praktijk gebruikt worden?

De bijdrage aan het antwoord op deze vraag is geleverd door het beantwoorden van de volgende vraag:

Welke eigenschappen zijn toe te kennen aan een architectuurprincipe op basis van de classificatie (door middel van het Proposition Framework) van de architectuurprincipes van Stater NV?

De architectuurprincipes van Stater NV beschrijven vooral hoe men wenst dat de ICT-oplossing gebouwd wordt en welke functies er in moeten zitten. Hier mag bij voorkeur niet van worden afgeweken. De architectuurprincipes beschrijven dus vooral de functie en constructie van de te bouwen ICT-oplossing. De architectuurprincipes zijn op logisch of conceptueel niveau beschreven in natuurlijke taal en zijn richtinggevend. De architectuurprincipes hebben vooral betrekking op de ICT-oplossing zelf en weinig op de context en omgeving waarin de ICT-oplossing zal gaan worden ingezet. De impact, motivatie, issues en conflicten zijn niet opgenomen in de beschrijvingen van de architectuurprincipes.

Welke eigenschappen hebben architectuurprincipes volgens de theorie?

De bijdrage aan het antwoord op deze vraag is geleverd door het beantwoorden van de volgende vraag:

Welke eigenschappen zijn toe te kennen aan een architectuurprincipe op basis van de theorie van architectuurmethoden- en raamwerken?

In dit onderzoek is de Dragon1 methode nader onderzocht. De theorie over architectuurprincipes van deze methode is gepositioneerd doormiddel van het raamwerk.

- Een principe is in essentie en bij voorkeur **Intrinsic**. Wanneer een principe **Desired** is dient er een handhavingmechanisme actief te zijn dat er voor zorgt dat het voordeel van het hanteren van het principe ook daadwerkelijk behaald wordt.
- Een principe is in essentie en bij voorkeur altijd observeerbaar / actief in een bepaalde situatie. De **Probability** dient in ieder geval hoger te liggen dan 80%.
- Het **Obedience Level** van een principe is in essentie **Strict**. Een principe kan ook **Overridable** zijn. Het afwijken van een principe is echter niet gewenst.
- Een principe is een tijdsgebonden wetmatigheid. Een principe kan op verschillende tijdsperiode van toepassing zijn.
- Een principe refereert zowel naar de **Operations**, **Structuring** en **Strategy** van een onderneming. Indien één van deze gebieden niet belicht wordt betreft het geen principe.
- Zowel de toegevoegde waarde (**Valuation**), mogelijkheden (**Functions**) als de wijze waarop dit bewerkstelligd wordt (**Construction**) dienen te worden beschreven in een principe.
- Een principe is bij voorkeur op conceptueel niveau beschreven.
- Principes kunnen op elk niveau dat onderkend wordt bij **Enablement abstraction** ingezet worden (van toepassing zijn).

- Principles kunnen op verschillende domeinen van toepassing zijn, waaronder alle benoemde niveaus bij **Organisational range**.
- Een principe beschrijft zowel het **Behaviour** (wat gebeurt er), **Passive Structure** (waar leidt dit toe/wat is het gevolg/resultaat) als **Active Structure** (wat of wie doet het).
- Een principe kan zowel betrekking hebben op een **Operational system** als een **Transformational system**.
- Een principe is in natuurlijke taal beschreven en de syntactische variatie wordt ingeperkt door een vast stramien. Een principe is dus **Semi-formal** beschreven.
- Een principe is **Actionable**. De beschrijving van een principe moet er voor zorgen dat het direct duidelijk is of een situatie zich conformeert aan het principe.
- De actors / objecten die zich dienen te conformeren aan een principe of waarbinnen het principe observeerbaar dient te zijn: mens, systeem, systeemonderdeel, organisatie, onderneming, instelling, organisatiedomein, project, proces, product etc. Dit zijn slechts enkele voorbeelden.
- Bij **Intrinsic propositions** vormt een bewijs dat het principe werkt en zorgt voor het gewenste resultaat de motivatie voor het hanteren van een principe. **Regulating propositions** kunnen zowel **risk-based** als **refinement-based** zijn. Principles zijn afgeleid van strategische uitgangspunten en kwaliteitseisen die gezien worden als hogere orde principes.
- De impact beschrijft de bedrijfs- voordelen en nadelen, implementatie impact en issues en conflicten van het hanteren van een principe.
- Voor het hanteren van een principe kunnen alle genoemde deployment strategieën (**Communicate**, **Construct** en **Enforce**) gebruikt worden.

In hoeverre is het Proposition Framework correct en bruikbaar in de praktijk?

De bijdrage aan het antwoord op deze vraag is geleverd door het beantwoorden van de volgende vragen:

Welke problemen zijn er bij het toepassen van het Proposition Framework op een praktijkcasus?

Welke aanbevelingen zijn er op het Proposition Framework naar aanleiding van deze problemen?

Hoe kan Stater NV het Proposition Framework gebruiken om betere architectuurprincipes te formuleren?

Het Proposition Framework v0.9 is in de huidige vorm een handig overzicht van mogelijke eigenschappen die architectuurprincipes kunnen hebben. Het kan als checklist dienen bij het opstellen van architectuurprincipes en hulp bieden bij het bepalen van de functie van een architectuurprincipe. Hiermee wordt bedoeld dat een organisatie kan bepalen hoe men het architectuurprincipe als instrument wil gaan inzetten en welke eigenschappen het dan moet hebben.

Als classificatiemiddel is het echter nog lastig te gebruiken. Dit komt vooral doordat de classifiers zeer globaal zijn beschreven en daardoor weinig zeggen over wanneer een eigenschap mag worden toegekend aan een architectuurprincipe. Hierdoor is het lastig toe te passen in de praktijk. Hier is veel kennis op het gebied van enterprise architectuur en architectuurprincipes voor nodig. Het raamwerk laat dus nog te veel ruimte voor interpretatie over, waardoor er onzekerheid ontstaat over de juiste classificatie. Hierdoor zijn de resultaten uit de verschillende casussen lastig te vergelijken en is het moeilijk om antwoorden te vinden op de vragen die ten grondslag liggen aan het ontwikkelen van het raamwerk. Het is dus belangrijk om de attribuuftypes en de classifiers nog scherper te formuleren om verkeerde interpretatie uit te sluiten. Vooral de object attribuuftypes zijn lastig te classificeren.

Meer inzicht verkrijgen in de eigenschappen van de verschillende propositie soorten wordt in mijn ogen met de huidige versie van het raamwerk en het uitgevoerde experiment nog niet bereikt. Dit komt omdat het

raamwerk nog onvoldoende is uitgekristalliseerd en van te voren niet beoordeeld is of de architectuurprincipes uit de casussen wel van voldoende kwaliteit zijn en dus als goed voorbeeld gezien mogen worden.

Uiteindelijk kan het raamwerk wel laten zien hoe verschillende soorten proposities er uit zien (welke eigenschappen ze hebben), maar blijft het lastig om vervolgens te bepalen wat nu een architectuurprincipe, principe, regel etc. genoemd mag worden en welke eigenschappen nu gewenst zijn.

Welke problemen ontstaan er in de praktijk door de wijze waarop architectuurprincipes gebruikt en geformuleerd worden?

De bijdrage aan het antwoord op deze vraag is geleverd door het beantwoorden van de volgende vraag:

Welke problemen zijn te onderkennen binnen het architectuurproces van Stater NV en welke rol speelt het gebruik en de wijze van formuleren van architectuurprincipes hierin?

- **Tijdigheid:** De uitwerking van de architectuurprincipes zijn niet gereed of beschikbaar bij aanvang van een groot automatiseringstraject. Hierdoor loopt architectuur achter de feiten aan.
- **Mate van detail:** De uitvoerende partij heeft veel interpretatie ruimte. Dit is goed opgepakt door het ervaren ontwikkelteam van Stater NV, maar had bij een ander partij makkelijk tot grote afwijkingen kunnen leiden.
- **Onduidelijkheid semantiek:** Maandelijks veranderende de betekenis van bepaalde begrippen en de daarbij gestelde doelstellingen. Men is tijdens het traject nog zoekende.
- **Architectuur wordt uitgelegd o.b.v. prototypes in .Net.** Erg kostbaar en niet echt effectief.
- **Geen verankering architectuur in de business van Stater NV.** D.w.z dat het nastreven van een bepaald architectuurprincipe zou leiden tot onacceptabele hoge doorlooptijd en kosten voor aspecten die nu commercieel door de business niet als belangrijk of haalbaar worden geacht. M.a.w. men heeft getracht geprobeerd door middel van de opgestelde architectuur zich in te dekken voor alle mogelijke toekomstige wellicht handige situaties.
- **Het ontbreken van concrete en op de praktijk afgestemde richtlijnen** hebben bij het ontwikkelteam dat verantwoordelijk is voor middleware, tot ingrijpend rework geleid. Consequentie hiervan is veel vertraging en extra kosten. Dit is rechtstreeks het gevolg van onvolledige formulering van de architectuurprincipes en achteraf aanleveren van aanvullende onverwachte requirements.
- **Weinig pragmatisch:** D.w.z dat veel nadruk wordt gelegd op het realiseren van flexibiliteit door middel van architectuurprincipes, terwijl onzeker was en is of dit ooit nodig is en waarvan duidelijk is dat de eerste klanten er weinig behoefte aan hebben.
- **De architecten waren niet nauw genoeg betrokken in het project**, waardoor ze de 'feeling' verloren met praktijk problemen, business context en project problematiek en daardoor konden ze niet sturen en het proces eigenlijk alleen frustreren, ondanks hun goede bedoelingen.

Kunnen deze problemen voorkomen worden door architectuurprincipes anders te formuleren?

De bijdrage aan het antwoord op deze vraag is geleverd door het beantwoorden van de volgende vraag:

Zijn deze problemen te voorkomen door gebruik te maken van een gestructureerd stappenplan voor het opstellen van architectuurprincipes?

Het is lastig te beoordelen in hoeverre problemen hadden kunnen voorkomen indien een gestructureerd stappenplan was gebruikt voor het opstellen en formuleren van architectuurprincipes. De architectuurprincipes zijn medio 2004 opgesteld. Toentertijd had men een ander beeld voor ogen van wat een architectuurprincipe moest zijn dan nu het geval is.

Een gestructureerd stappenplan voor het opstellen en formuleren van architectuurprincipes helpt echter voorkomen dat veel gemaakte fouten worden herhaald. Het stappenplan stimuleert om na te denken over zaken die de volledigheid en bruikbaarheid van een architectuurprincipes verhogen, maar geeft ook aan hoe en wanneer architectuurprincipes geformuleerd en gebruikt dienen te worden. Op basis van het herformuleren zijn ook misstanden aan het licht gekomen die voorkomen hadden kunnen worden door de architectuurprincipes anders te formuleren. Een waardeoordeel van de domeinexpert bevestigt deze conclusie.

13.2 Beantwoording onderzoeksvraag

Welke eigenschappen en functie heeft een architectuurprincipe en hoe dient een architectuurprincipe geformuleerd te worden?

Op basis van verkregen antwoorden op de deelvragen en het aanscherpen van een stappenplan voor het formuleren van architectuurprincipes is het volgende antwoord verkregen op de onderzoeksvraag:

De functie van het architectuurprincipe is het vertalen van de strategie naar een ontwerp voor een oplossing die deze strategie ten uitvoer brengt. Een architectuurprincipe zorgt er voor dat strategische uitgangspunten, doelen, eisen en wensen op de juiste wijze worden doorvertaald naar (ontwerp)regels en (bouw)richtlijnen. Een architectuurprincipe geeft dus richting aan het ontwerp van de organisatie.

Een architectuurprincipe beschrijft hoe iets werkt of hoe iets is én wat dit voor resultaat heeft en hoe dit resultaat bijdraagt aan het verwezenlijken van de strategische uitgangspunten, doelen, eisen en wensen. Het architectuurprincipe beschrijft een bewezen, betrouwbare en toekomstvaste aanpak waarvan het resultaat gegarandeerd kan worden binnen een bepaalde context. De beschrijving dient begrijpbaar te zijn voor de opdrachtgever en opdrachtnemer. Een architectuurprincipe beschrijft een (deel van een) concept dat kan worden toegepast in een context.

Een architectuurprincipe heeft een communicatieve functie en de motivatie voor het hanteren van het architectuurprincipe dient te worden onderkend en vastgelegd. De bedreigingen die het succesvol implementeren en / of hanteren van het architectuurprincipe in de weg kunnen staan dienen bekend te zijn en met een effectief handhavingmechanisme te niet worden gedaan. De context waarin het architectuurprincipe zal worden toegepast dient bepaald te worden en heeft invloed op de wijze waarop invulling gegeven wordt aan de rest van het architectuurprincipe.

Hoe dient een architect op de juiste manier architectuurprincipes te formuleren en te gebruiken:

- Stel in samenwerking met de opdrachtgever de strategische uitgangspunten (doelen, eisen en wensen) vast indien dit nog niet gedaan is.
- Bepaal welke (bewezen) concepten kunnen worden gebruikt in een oplossing om deze strategische uitgangspunten te verwezenlijken.
- Beschrijf met de architectuurprincipes hoe deze concepten werken en hoe deze bijdragen aan het verwezenlijken van de strategische uitgangspunten, doelen, eisen en wensen.
- Laat de architectuurprincipes doorvertalen naar (ontwerp)regels en (bouw)richtlijnen.
- Geef invulling aan de attributen van de architectuurprincipes die beschreven dienen te worden zoals handhavingmechanisme, implementatie impact, validiteit, bedrijfsvoordelen- en nadelen, kosten / baten zodat de opdrachtgever een objectief en realistisch beeld heeft van waar men voor kiest.
- Verkrijg goedkeuring van de opdrachtgever over de geformuleerde architectuurprincipes.
- Zoek een geschikte partij die deze oplossing kan realiseren.

Door het architectuurprincipe deze functie en beschrijving mee te geven is het architectuurprincipe van toegevoegde waarde t.o.v. van de al bestaande richtinggevende uitspraken. Sterker nog: het vormt de verbindende schakel tussen deze richtinggevende uitspraken en is daardoor een krachtig hulpmiddel om een toekomstvaste en betrouwbare organisatie of ICT-oplossing te realiseren.

13.3 Belangrijkste bevindingen

Bevinding 1

Door alle architectuurprincipes op conceptueel niveau te beschrijven is er minder verschil in de mate van detail waarmee een architectuurprincipe beschreven is en zijn architectuurprincipes herbruikbaar (ook voor andere organisaties). Daarnaast heeft men zo een houvast waar men een architectuurprincipe aan kan herkennen en kan men het een plaats geven binnen de overige proposities die een organisatie erkend.

Bevinding 2

Het formaliseren van architectuurprincipes draagt bij aan het scherper formuleren van architectuurprincipes en zorgt er voor dat de mate waarin de beschrijving vrij is van ambiguïteit hoger is. Het redeneerproces dat hierbij gevolgd wordt leidt tot inzicht in onduidelijkheden en inconsistentie in de beschrijving in natuurlijke taal. Door het bewust hanteren van de redeneerstrategie die toegepast wordt bij propositie- en predicatenlogica komt men tot een beschrijving van een principe die dichter bij de werkelijkheid ligt. Op deze wijze formuleer je als het ware een te bewijzen stelling. Het bewust toepassen van deze redeneerstrategie zorgt er voor dat je kritisch kijkt naar wat er staat en draagt bij aan het S.M.A.R.T. beschrijven van een architectuurprincipe.

Bevinding 3

Door de verschillende soorten proposities die over het algemeen erkend worden en waar ook nog eens vaak verschillende functies en definities aangekoppeld zijn, is het bijna onmogelijk om deze op een juiste wijze aan elkaar te knopen en aan elkaar te relateren. Wanneer je een definitie of functie van een architectuurprincipe vast stelt dien je ook de overige propositie soorten van een definitie en functie te voorzien zodat de meerwaarde duidelijk wordt en de functie en definitie weinig tot geen overlap heeft met de al erkende proposities. Op basis hiervan is het noodzakelijk dat wanneer een organisatie architectuurprincipes gaat formuleren eerst vast stelt: welke functie en definitie ken ik toe aan een architectuurprincipe zodat het een toegevoegde waarde is t.o.v. van de al gehanteerde propositiesoorten binnen de organisatie. De gevolgen van wanneer dit niet wordt gedaan is eens te meer zichtbaar geworden in dit onderzoek.

Bevinding 4

Een architectuurprincipe dat alleen een wens, eis of streven beschrijft heeft geen toegevoegde waarde ten opzichte van andere proposities die gehanteerd worden binnen een organisatie. Deze functie wordt immers al afgevangen door de propositie soorten 'eisen', 'wensen' en 'uitgangspunten'.

Bevinding 5

Een architectuurprincipe dat alleen beschrijft hoe men iets moet doen, wat men niet moet doen, wat wel mag en wat niet mag, waar men zich bij voorkeur wel of niet aan moet houden heeft geen toegevoegde waarde ten opzichte van andere proposities die gehanteerd worden binnen een organisatie. Deze functie wordt immers al afgevangen door de propositie soorten 'regels', 'voorschriften', 'richtlijnen'.

Bevinding 6

Een architectuurprincipe heeft toegevoegde waarde ten opzichte van een andere proposities die gehanteerd worden binnen een organisatie indien het een mechanisme of wijze beschrijft dat er voor zorgt dat de propositie soorten 'eisen, wensen en uitgangspunten' op de juiste wijze worden doorvertaald naar de propositie soorten 'regels, voorschriften en richtlijnen'.

Bevinding 7

Er is een verschil tussen het formuleren van een architectuurprincipe en het hanteren / toepassen van een architectuurprincipe. Dit onderscheid wordt nauwelijks onderkend en zorgt voor veel problemen omtrent de functie van een architectuurprincipe en de wijze waarop deze geformuleerd dient te worden. Vaak wordt maar één van deze taken uitgevoerd door de architecten binnen een organisatie, waardoor er altijd een stuk

ontbreekt:

- Wanneer men alleen architectuurprincipes formuleert is niet bekend hoe deze gehanteerd of toegepast gaan worden (hoe ze geïmplementeerd of passen binnen een organisatie).
- Wanneer men allen architectuurprincipes toepast of hanteert en de beschrijving hiervan vastlegt is niet bekend wat de oorspronkelijke architectuurprincipes zijn geweest en waar de motivatie vandaan is gekomen om deze te hanteren.

De ideale situatie zou zijn als een architect alleen architectuurprincipes hanteert of toepast en het oorspronkelijk architectuurprincipe als bron vermeld. Men kan zo altijd zien wat de motivatie is geweest van het toepassen / hanteren van het principe. Het telkens maar weer zelf formuleren van architectuurprincipes door architecten leidt tot het feit dat constant dezelfde architectuurprincipes worden opgeschreven, alleen net op een andere wijze geformuleerd. Daarnaast leidt dit tot het opstellen van architectuurprincipes waarvan men niet weet of deze wel bewezen zijn. Een architectuurprincipe dient bewezen te zijn en wanneer dit niet voor 100% het geval is dient gedocumenteerd te zijn waar de zwakke punten en mogelijke risico's liggen. Hier kan men bij het toepassen, hanteren en erkennen van een architectuurprincipe rekening mee houden.

13.4 Vervolgonderzoek

Om dit onderzoek in zijn geheel af te ronden zijn nog een aantal activiteiten noodzakelijk. De belangrijkste activiteiten bestaan uit het verder uitwerken van het stappenplan en deze te toetsen in de praktijk. De resultaten van dit onderzoek kunnen echter vooral gebruikt worden voor het optimaliseren van bestaande stappenplannen voor het formuleren van architectuurprincipes.

Wat ik vooral wil benadrukken is dat de functie en inhoud van een architectuurprincipe nu is eenduidig vastgesteld en erkend dient te worden zodat het geplaatst kan worden binnen de overige lijst van instrumenten zoals regels, doelstellingen en uitgangspunten.

13.5 Reflectie

Een afstudeerscriptie, onderzoeksrapport, betoog, column of welke vorm dan ook waarin je iets schrijft over Enterprise Architectuur leidt tot een stuk tekst waarin de mening en visie op dit vakgebied sterk in terug komt. Een afstudeerscriptie over Enterprise Architectuur schrijven is een lastige klus. Voordat je überhaupt iets kunt zeggen over EA ben je al een jaar verder en is kennis opgedaan en zijn meningen en visies gevormd die hoe dan ook terugkomen in alle onderdelen van het document dat het resultaat beschrijft. Het vergaren van kennis, het uitvoeren van experimenten, het onderzoeken, het beschrijven, het vormen van een mening en visie vindt tegelijkertijd plaats en is een continue proces. Het is dan ook lastig om een goede structuur te hanteren waarin je een duidelijke afscheiding kunt maken tussen proces, kennis, experiment, resultaat, conclusie. Het schrijven van deze afstudeerscriptie is een proces geweest waarin steeds weer de probleemstelling en het doel is bijgeschaafd.

Voor dat ik begon aan deze afstudeerscriptie heb ik via cursussen op de Radboud Universiteit Nijmegen al kennis gemaakt met de begrippen digitale architectuur, enterprise architectuur, principes en architectuurprincipes. Wat mij hier toen van bij is gebleven, is dat voor velen (waaronder ik zelf) dit lastig te plaatsen was. De functie en meerwaarde was niet meteen duidelijk maar toch blijft het hangen en de interesse voor het onderwerp ook. Gedurende de verdere kennismaking met het vakgebied en het schrijven van deze afstudeerscriptie is gebleken dat mijn eerste indruk van dit vakgebied niet geheel onlogisch was. Door de verschillende definities, functies en eigenschappen die worden toegekend aan EA en de ontelbare begrippen die hier binnen zijn te noemen is het ook lastig om een duidelijke beeld te krijgen van wat EA nu is en welke rol architectuurprincipes nu hebben. Door het schrijven van deze afstudeerscriptie heb ik een duidelijk beeld gekregen van de verschillende opvattingen en een positie ingenomen over wat nu EA is en welke functie het heeft. Doordat via een geleidelijke weg hier pas inzicht in is verkregen heeft het meer tijd gekost om deze afstudeerscriptie af te ronden dan gepland. Een stuk wat een maand geleden geschreven is kan door nieuwe inzichten weer omver geworpen worden wat veel extra werk oplevert. Helaas is het men niet meer gelukt om daadwerkelijk een compleet stappenplan op te leveren. Uiteindelijk is het echter een zeer leerzame periode geweest waar ik zeer intensief het vakgebied EA gevolgd en bekeken heb.

Literatuur

- [BHPW06] P. van Bommel, S.J.B.A. Hoppenbrouwers, H.A. Proper, and Th.P. van der Weide. Giving Meaning to Enterprise Architectures - Architecture Principles with ORM and ORC. In R. Meersman, Z. Tari, and P. Herrero, editors, *On the Move to Meaningful Internet Systems 2006: OTM Workshops - OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, GADA, MIOS+INTEROP, ORM, PhDS, SeBGIS, SWWS, and WOSE 2006*, Lecture Notes in Computer Science, Montpellier, France, EU, October/November 2006. Springer, Berlin, Germany, EU.
- [BBHP07] P. van Bommel, P.G. Buitenhuis, S.J.B.A. (Stijn) Hoppenbrouwers, and H.A. Proper. Architecture Principles – A Regulative Perspective on Enterprise Architecture. In M. Reichert, S. Strecker, and K. Turowski, editors, *Enterprise Modelling and Information Systems Architectures (EMISA2007)*, number 119 in Lecture Notes in Informatics, pages 47–60, Bonn, Germany, EU, Oktober 2007. Gesellschaft fur Informatik.'
- [BMM06] BMM Team. Business Motivation Model (BMM) Specification. Technical Report dtc/06–08–03, Object Management Group, Needham, Massachusetts, USA, August 2006.
- [Bui07] P.G. Buitenhuis. Fundamenten van het principe (Foundations of principles). Master's thesis, Institute for Computing and Information Sciences, Radboud University Nijmegen, Nijmegen, The Netherlands, EU, March 2007. In Dutch.
- [CJN+07] G.J.N.M. Chorus, Y.H.C. Janse, C.J.P. Nellen, S.J.B.A. Hoppenbrouwers, and H.A. Proper. Formalizing Architecture Principles using Object-Role Modelling. Via Nova Architectura, February 2007.
- [Dragon1/07] Mark Paauwe, Het methodisch raamwerk voor enterprise architectuur van Paauwe, december 2007
- [Dragon1/08a] Mark Paauwe, Dragon1 Stappenplan - Architectuurprincipes opstellen, reviewen en herformuleren v0.15.2c, september 2008.
- [Dragon1/08b] Mark Paauwe, Dragon1 Architectuurprincipesschema – versie 0.3c, november 2008.
- [DYA06] Bijdrage van Serge Bouwens, senior informatie architect, Sogeti Nederland B.V. aan de rubriek 'Archi-tekst' op www.dya.info
- [DYA08] Serge Bouwens: White Paper 'Architectuurprincipes', deel 1: Basics, april 2008.
- [Gre2007] Danny Greefhorst, Erik Proper, Frank van den Ham, Principles: de hoeksteen voor architectuur.
- [GA03] J. Gordijn and H. Akkermans. Value based requirements engineering: Exploring innovative e-commerce ideas. *Requirements Engineering Journal*, 8(2):114–134, 2003.
- [GKV03] D. Greefhorst, H. Koning, and H. van Vliet. Dimensies in architectuurbeschrijvingen. *Informatie*, 45(11):22–27, 2003. In Dutch
- [IEE00] Recommended Practice for Architectural Description of Software Intensive Systems. Technical Report IEEE P1471–2000, The Architecture Working Group of the Software Engineering Committee, Standards Department, IEEE, Piscataway, New Jersey, USA, September 2000.
- [Mae03] R. Maes. Informatiemanagement in kaart gebracht. PrimaVeraWorking Paper 2003-02, June 2003. In Dutch.
- [Paauwe/Rijsenbrij07] Mark Paauwe – Daan Rijsenbrij, november 2007, Digitale architectuur in beeld brengen

en communiceren op basis van het Paauwe-Rijsenbrij Visualisatieschema, Versie 0.3 – een verkenning

- [SBV06] SBVR Team. Semantics of Business Vocabulary and Rules (SBVR). Technical Report dtc/06–03–02, Object Management Group, Needham, Massachusetts, USA, March 2006.
- [STA06] Stater Architectuurbeschrijving v1.1
- [TOGAF08] Website Open Group - Togaf, <http://www.opengroup.org/togaf>, november 2008
- [Rijsenbrij04] D.B.B. Rijsenbrij (2004), Architectuur in de digitale wereld (versie nulpunt drie). Inaugurale rede (2004).
- [xAF06] xAF working group. Extensible Architecture Framework version 1.1 (formal edition). Technical report, 2006.

Proposition Framework V0.9

Joint NAF/ArchiMate “Business Principles” Workgroup

January 30, 2008

1 Introduction

When studying the many contemporary definitions on architecture [SG96, IEE00, TOG04, L+05], one can discern two key perspectives on enterprise architecture:

A regulation-oriented perspective in which an architecture is regarded as a way to govern the design, evolution or transformation of a system by means of regulations. In other words, in this perspective architecture is regarded as a prescriptive notion limiting the design freedom with regards to the design and evolution of a system.

A design-oriented perspective in which the essential properties of a system are being designed. This perspective treats architectures as actual specifications of high level system designs focussing on ‘architecturally relevant’ design decisions and tradeoffs.

In each of these perspectives, we will see things referred to as “principles”. Their definitions, however, may quite well differ considerably, providing all the more reason to start our investigations. Furthermore, some authors use the term policy rather than principle [Kee91]. In addition, several organisations make use of “business rules”, where at times the difference “business rules” and “principles” has become rather blurry. For us this marks the starting point of our investigations. We aim to gain a better insight into the different occurrences of these “things” and hopefully arrive at a conceptual framework in which to position/classify these “things”. A secondary aim is to arrive at a terminological framework.

The aim of this document is to provide a first version of a framework to classify “things called principles” and “things called business rule”. This document is part of an effort to understand the different concepts referred to as architecture principles, design principles, business principles and business rules, and their mutual relationships. We aim to do so by using this framework to classify principles occurring in practice, and refining the framework in the process. The former will take the form of a series of student projects involving the investigation of “policies”, “principles”, “guidelines” and “business rules”, and their motivations, in a number of real-world cases.

In the next weeks this framework should evolve based on discussions among the members of the working group. By February, when we expect the student projects to commence, we should have reached version 1.0 of this framework. By then we should also be able to produce an ORM domain model to aid the students in their project work.

Whatever a policy, principle, or rule is, we take the assumption that it is a proposition which a system may or may not exhibit. This system might be the actual enterprise, the design of the enterprise or the transformation process of the enterprise. In the framework we will see differences in the systems a to which such a proposition may refer to, the validity of the proposition, its goals, its specificity, etc. In discussing the framework we will actually try and avoid the terms policy/principle/rule as much as possible, and simply refer to the term proposition.

Based on [SBV06], we define a proposition to be:

A meaning that is asserted when a sentence is uttered or inscribed and which is true or false.

As also discussed in [SBV06], the word ‘proposition’ has two common meanings: first, a statement that affirms or denies something, and second, the meaning of such a statement. The concept ‘proposition’ is here defined in the second sense and should not be confused with the statement of a proposition. Once the framework has evolved through the use in experiments, we can endeavour

to define: policies, business principles, design principles, business rules, design rules, guidelines, etcetera as specialisations of the general concept of proposition.

To enable the effective classification of policies, principles and rules, we aim to create a classification framework with an orthogonal set of attribute types in terms of which these propositions can be classified. The framework will comprise the following classes of attribute-types:

Form – attribute-types pertaining to the form in which the proposition itself is stated.

Object – attribute-types dealing with the identification of the object which the proposition pertains to.

Validity – attribute-types pertaining to the proposition's claimed/desired validity.

Actors – attribute-types dealing with the identification of those actors who are expected to (be observed to) respect the propositions.

Context – attribute-types pertaining to the contextual embedding of the proposition and its validity claim/desire in terms of proofs or contextual argumentation, examples, etc.

Each attribute-type has an underlying domain of allowed values. These values are referred to as the classifiers. In the case of most of the attribute-types we actually only have a small number of allowed values with a strict order on these values. In this case, the attributes types could be regarded as dimensions spanning a vector space. However, we will also see some attribute-types which do not have a specific order defined for the values in their associated domain. Even more, in the case of e.g. motivations, the classifiers may be narrative descriptions referring to other propositions.

In our case studies we will allow propositions, for each attribute-type, to have an arbitrary number of classifiers. Based on our empirical studies we will be able to glean more specific constraints, which e.g. express the plausibility of allowing only one classifier for a specific dimension.

This means we have the following fact types and sub-types:

- Proposition is referred to by Name
- Proposition has Definition
- Proposition is classified for Attribute type with Classifier
- Attribute-type has underlying Domain
- Attribute-type is of Attribute-type class
- Finite domain has Value
- Finite domain IS-SUB-TYPE-OF Domain
- Ordered domain has Value at Position
- Ordered domain IS-SUB-TYPE-OF Domain
- Elaboration domain refers to Proposition
- Elaboration domain IS-SUB-TYPE-OF Domain

with as constraints:

- Attribute-type class is one of fvalidity; object; actors; form; contextg
- each Proposition has precisely one Definition

However, there is one complicating factor. A given proposition may have multiple constellations of classifiers associated. This is due to the fact that the same proposition may have different levels of desiredness and contextual embedding depending on e.g. the actors who are to obey the proposition, or the part of the enterprise that should obey by them, etc. Therefore, we need to refine:

Proposition is classified for Attribute-type with Classifier
to:

Proposition has Classification
Classification has for Attribute-type associated Classifier

This also means that were we stated above: “.. in the case of e.g. motivations, the classifiers may be narrative descriptions referring to other propositions”, we actually have to refine this to “the classifiers may be narrative descriptions referring to classifications of other propositions”.

However, for the object and form attribute types, only unique classifications are allowed:

for each combination of a Proposition and
an Attribute-type which is of Attribute-type class ‘form’ or ‘object’,
there exists at most one Classifier such that:
that Proposition has some Classification
which has for that Attribute-type associated that Classifier

Hypothesis:

To each combination of a proposition and one of its validity sub-classifications, a unique actor sub-classification can be associated (actors a are subjected to proposition p with validity v).

Hypothesis:

To each combination of a proposition and a pair of its validity and actor sub-classifications, a unique context sub-classification can be associated (actor a is subjected to proposition p for validity v, in context c

2 Validity attribute-types

According to the Webster dictionary, the primary meaning of the word principle is:

a: a comprehensive and fundamental law, doctrine, or assumption **b (1):** a rule or code of conduct **(2):** habitual devotion to right principles <a man of principle> **c:** the laws or facts of nature underlying the working of an artificial device.

What seems to distinguish the above three sub-definitions are levels of validity. We actually split this into two attribute-types.

2.1 Intricity

The first validity dimension deals with the intricity of the proposition. The second attribute-type (see the next subsection) deals with the level at which it is inherent or desired in the system. This leads to the following classifiers:

Intrinsic – referring to the laws or facts of nature underlying the working of a system. These are properties which are inherent to some system. Some examples are: (1) Heisenberg’s uncertainty principle, (2) Bernoulli’s principle, (3) Pauli’s exclusion principle. These propositions are unavoidable. They hold like the laws of nature.

Desired – referring to propositions which one would like to have in a system/enterprise. These propositions concern directives or doctrines that require some pro-active form of deployment.

2.2 Probability

The probability with which the property is (desired to be) observable in the enterprise. For our experiment we will use three classifiers:

Always – 100%.

Usually – Less than 100%, but at least 50%.

Sometimes – Less than 50%.

Hypothesis:

For intrinsic propositions, only the always and usually options will be used.

2.3 Obedience level

The extend to which the proposition is abided by (in the case of desired propositions, this becomes enforcement). Based on [SBV06] we identify the following classifiers:

Strict – no exceptions occur/are-allowed.

Overridable – exceptions are possible, if they are motivated.

Guiding – the proposition provides guidance to actors.

There is probably a connection between the probability and the obedience levels. We might even come to the conclusion that the probability attribute-type is unnecessary for our purposes.

3 Object attribute-types

In this section we discuss the attribute-types that can be used to classify the object which the proposition pertains to. This typically pertains to the attribute-types present in architecture frameworks. The discussion below is therefore based on pre-existing work on attribute-types in architecture/information-management frameworks [GKV03, xAF06] and frameworks for principles and/or business rules [Bui07, BBHP07]. Nevertheless, not all object attribute-types are based on attribute-types from architecture (principle) frameworks.

Note: the object attribute-types pertain to the combination of a proposition and its validity vector. For example:

_ Proposition p with intricity 'desired' and probability 'usually' pertains to the business architecture during 2008.

_ Proposition p with intricity 'desired' and probability 'always' pertains to the business architecture from 2009 onward.

3.1 Time

A specification of the time-span during which the property holds. This puts the object which the proposition pertains to on a temporal plane. In other words, we could make a distinction between the enterprise (as an object referred to by the proposition) in 2008 and the enterprise during 2009.

For the temporal attribute-type we do not specify specific values. The temporal attribute-type of a proposition can be classified in terms of the interval: in 2008, from 2011 onward, before 1999, etcetera.

3.2 Control abstraction

This attribute-type, based on [Mae03], is concerned with abstraction from the operational enterprise. In [Mae03] a distinction is made between: strategy, structure and operations, also associated with the alliteration (in Dutch): richten (aim), inrichten (organize) and verrichten (do). We adopt this distinction as follows:

Operations – refers to the enterprise as it is operationally functioning (operational level).

Structuring – refers to the structuring of the enterprise (tactical level).

Strategy – refers to the strategy followed by the enterprise in achieving its goals (strategic level).

3.3 Construction abstraction

The level of abstraction from the construction of the enterprise. In our case studies we identify:

Valuation – what value does the system/enterprise provide to its environment/ecology?

Function – which functions does the system/enterprise offer to its environment in creating this value?

Construction – how does the system/enterprise realize these functions?

The function/construction distinction is due to [Die06], while the valuation is based on [GA03].

3.4 Physical abstraction

This attribute-type is concerned with the level of abstraction from underlying technologies (including IT, human technology, machines, etc) used to implement the system. This attribute-type is based on [ISO87]. In our case studies we identify:

Physical – the actual mapping of the system/enterprise onto technological components and/or infrastructural elements.

Logical – how will the system/enterprise be implemented in terms of technologies and types of infrastructural elements?

Conceptual – what system/enterprise is needed, irrespective of IT, human technology, machines, etc?

3.5 Enablement abstraction

This attribute-type refers to the support (enablement) of business products and services in terms of information technology. When considering an enterprise, several system-types can be discerned covering different facets of the enterprise [xAF06]. Some example system-types are: business system, information system, production system, IT infrastructure and management & control system. Most architecture frameworks, in line with their IT roots, which focuses on business realisation through IT. In these latter frameworks we usually find classifiers (system types) such as: business, information systems, applications and infrastructure. In our case studies we shall therefore use the following classifiers:

Business – the business products and services, their markets, etcetera, the business processes needed to produce/deliver the products and services. This perspective identifies why we would need (automated) information processing.

Information – the information domains and information processing needed to realise the business activities. This perspective identifies

Application – the IT applications needed/selected to support information processing.

Infrastructure – the IT infrastructure used for/by the IT applications.

3.6 Organisational range

This refers to the range of the domain under consideration. In our case studies we will distinguish five classifiers:

Application – a specific software application and its direct context (including operational maintenance).

Information system – a specific information system (possibly comprising a number of applications) and its direct context (including operational maintenance).

Business unit – the proposition refers to a specific business unit within the enterprise.

Enterprise – the proposition refers to the entire enterprise.

Ecology – the proposition refers to the value-chain/ecology in which the enterprise operates.

Question:

Will principles mainly refer to classes of systems, as suggested in the work by Dietz and Hoogervorst, or will we also see principles referring to a specific system?

3.7 Aspects of dynamic systems

Enterprises are dynamic systems. In such systems there will be actors/agents which exhibit behaviour, which will impact on objects in the domain. In our case studies we shall use the following classifiers:

Behaviour – what happens in the enterprise/system?

Passive structure – what is this happening to?

Active structure – what/who is doing it?

3.8 Systemic order

An enterprise can be regarded as a system producing results to its environment. Such an operational system has a strategy for doing business, it has a structure and has its operations. In addition to the operational system, there might be a transformation system which is transforming the operational system into a system which is hopefully better able to seize opportunities. This transformation system also has its own strategy, structure and operations (being the transformation of the operational system). Needless to say that both the operational system and the transformation system have their own products and processes. Interestingly enough, one of the products of the transformation system will be a new operational system. The operational system can also be regarded as a first order system, while the transformation system then becomes a second order system which changes/transforms a first order system. This leads to the following classifiers:

Operational system – the enterprise considered as a first order system, dealing with its operational products and processes.

Transformation system – the enterprise considered at a second order level at which we can observe processes and products pertaining to the transformation of the operational system.

4 Form attribute-types

The form attribute-types primarily refer to the way the actual propositions are formulated, except for the last one.

4.1 Level of precision

The precision at which the results are specified. A possible way to express the level of precision would be in terms of its level of formality, referring to the level at which it would allow for mathematical/automated interpretation and/or manipulation. Some example levels would

be [Poh94]:

Informal – Informal would typically be a graphical sketch or a loose narrative description.

Semi-formal – Semi-formal involved the use of a controlled (graphical or textual) language, i.e. limiting the allowed syntactic variation, yet still without a well-defined semantics.

Formal – Formal implies the use of a (restricted) language with a well-defined semantics, enabling a precise and unambiguous interpretation of the results.

4.2 Level of actionability

The level at which a proposition is actionable by actors:

Definite – The proposition has distinct or certain limits (even though these may still be ambiguous).

Specific – The proposition is free from ambiguity. This might for example be done by using a language such as SBVR [SBV06] to express the propositions using an ORM [Hal01] based fact model at its base.

Actionable – In line with the definition provided in [SBV06], we consider a proposition to be actionable when an actor (who knows about this proposition) is able to decide directly whether or not an observed situation (including his or her own behaviour) complies to the proposition.

For a proposition to be actionable it will have to be accompanied by a clear definition of the measurements needed to assess the validity of the proposition in a given situation.

Hypothesis:

Based on the level of actionability, the set of sensible levels of precision is expected to decline. For specific, only semi-formal and formal are expected to be relevant, while in the case of actionable this probably shrinks to formal only.

5 Contextual embedding of propositions

In addition to characterising the propositions themselves, propositions found ‘in the wild’ are likely to be embedded in a context in terms of motivations, discussions of impact, etcetera. This leads to a set of attribute-types which have an infinite domain, in other words, not limited to a specific pre-defined set of values.

5.1 Motivation

The motivation of the proposition.

Hypothesis:

Depending on the purpose of the proposition (intrinsic, regulating or guiding), the motivation of the proposition will be different.

Intrinsic propositions – In the case of intrinsic propositions, the motivation requires a proof of some sort.

Regulating propositions – Two kinds of motivations may be used for regulating propositions.

Risk-based – As discussed in [BMM06], business policies and rules can be motivated in terms of risks resulting from the potential influence of influences. In [BBHP07] this idea is worked out in terms of a cost/benefits analysis to motivate the introduction of regulations. One would expect regulating propositions to be based on risks with a high expected impact.

Refinement-based – One proposition may be the refinement another proposition. In this case the motivation is probably some form of implementation decision. Note: one would expect the characteristics of the purpose/subject/form attribute-types to

be consistent or “refining” between the father and child proposition.

Guiding propositions – Both risk-based and refinement-based motivations can be used for guiding propositions. In the case of risk-based motivations, one would typically expect the proposition to address risks with a low expected impact. In the case of a refinement-based motivation, this is usually a suggested way of abiding by the higher level proposition.

1The expected impact of a risk is the product of the chance of the risk occurring and the impact when it would occur.

5.2 Impact

A discussion of the impact of a proposition. This will probably take place in terms of examples from the design-oriented perspective on architecture. For example, in terms of positive and negative examples using the ArchiMate [L+05] notation.

Hypothesis:

The impact of a proposition primarily makes sense for regulating and guiding propositions. In the case of inherent propositions, however, one may choose to discuss/illustrate the workings of the underlying mechanism as its “impact”.

5.3 Deployment

In the case of regulating and guiding propositions, the propositions need to be deployed somehow to ensure their conformance/application in the system. We identify three main strategies:

Communicate – Communicate the actors in the system that should abide-by/apply the propositions such that they abide by them. This is likely to involve soft-skills and/or mechanisms which recommend the actors which propositions to apply.

Construct – Construct the systems/mechanisms used by the actors in the system in such a way that abiding-by/applying the propositions is encouraged.

Enforce – Create an enforcement/punishment mechanism that enforces the application of the propositions.

Note: the enforcing strategy in a context of design principles included in an enterprise architecture strengthens the architect’s image as a police officer, rather than someone who helps projects (the educate strategy!).

6 The experiment

For each policy/principle/rule in the case:

1. List its short name and copy the actual proposition.
2. List a reference to its source(s) in the original documentation (document name and page).
3. Determine its characteristic(s) in terms of the five vectors.

References

[BBHP07] P. van Bommel, P.G. Buitenhuis, S.J.B.A. (Stijn) Hoppenbrouwers, and H.A. Proper. Architecture Principles – A Regulative Perspective on Enterprise Architecture. In M. Reichert, S. Strecker, and K. Turowski, editors, Enterprise Modelling and Information Systems Architectures (EMISA2007), number 119 in Lecture Notes in Informatics, pages 47–60, Bonn, Germany, EU, Oktober 2007. Gesellschaft für Informatik.

[BMM06] BMM Team. Business Motivation Model (BMM) Specification. Technical Report dtc/06–08–03, Object Management Group, Needham, Massachusetts, USA, August 2006.

[Bui07] P.G. Buitenhuis. Fundamenten van het principe (Foundations of principles). Master’s

- thesis, Institute for Computing and Information Sciences, Radboud University Nijmegen, Nijmegen, The Netherlands, EU, March 2007. In Dutch.
- [Die06] J.L.G. Dietz. Enterprise Ontology – Theory and Methodology. Springer, Berlin, Germany, EU, 2006.
- [GA03] J. Gordijn and H. Akkermans. Value based requirements engineering: Exploring innovative e-commerce ideas. Requirements Engineering Journal, 8(2):114–134, 2003.
- [GKV03] D. Greefhorst, H. Koning, and H. van Vliet. Dimensies in architectuurbeschrijvingen. Informatie, 45(11):22–27, 2003. In Dutch.
- [Hal01] T.A. Halpin. Information Modeling and Relational Databases, From Conceptual Analysis to Logical Design. Morgan Kaufmann, San Mateo, California, USA, 2001.
- [IEE00] Recommended Practice for Architectural Description of Software Intensive Systems. Technical Report IEEE P1471–2000, The Architecture Working Group of the Software Engineering Committee, Standards Department, IEEE, Piscataway, New Jersey, USA, September 2000.
- [ISO87] Information processing systems – Concepts and Terminology for the Conceptual Schema and the Information Base, 1987. ISO/TR 9007:1987.
- [Kee91] P.W.G. Keen. Shaping the Future – Business Design Through Information Technology. Harvard Business School Press, Boston, Massachusetts, USA, 1991.
- [L+05] M.M. Lankhorst et al. Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, Berlin, Germany, EU, 2005.
- [Mae03] R. Maes. Informatiemanagement in kaart gebracht. PrimaVeraWorking Paper 2003-02, June 2003. In Dutch.
- [Poh94] K. Pohl. The three dimensions of requirements engineering: a framework and its applications. Information Systems, 19(3):243–258, 1994.
- [SBV06] SBVR Team. Semantics of Business Vocabulary and Rules (SBVR). Technical Report dtc/06–03–02, Object Management Group, Needham, Massachusetts, USA, March 2006.
- [SG96] M. Shaw and D. Garlan. Software Architecture: Perspectives on an Emerging Discipline. Prentice–Hall, Englewood Cliffs, New Jersey, USA, 1996.
- [TOG04] TOGAF – The Open Group Architectural Framework, 2004.
- [xAF06] xAF working group. Extensible Architecture Framework version 1.1 (formal edition). Technical report, 2006.

Bijlage B: Uitwerking classificatie zonder commentaar en motivatie

Deze bijlage is toegevoegd om een helder overzicht te geven van de classificatie van de architectuurprincipes.

Procesmodulariteit

General	
Short name	Procesmodulariteit
Description	Processen zijn onderverdeeld in activiteiten die onderling onafhankelijk en ontkoppeld zijn; dit geldt zowel voor het primaire proces als ondersteunende activiteiten (controles, e.d.). Iedere activiteit kan zowel intern (bij Stater) als extern (bij de klant of elders) zijn belegd.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Strategy
Construction abstraction	Valuation
Physical abstraction	Conceptual
Enablement abstraction	Application
Organisational range	Ecology
Aspects of dynamic systems	Active structure
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Sometimes
Obedience level	Overridable
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater & Klant • Estate programma (het informatiesysteem)
Context	
Motivation	Regulating proposition (risk-based) [Stater heeft verschillende klanten in meerdere landen. De manier waarop het hypotheekstelsel werkt in deze landen verschilt onderling. In elk land en voor elke klant kan het proces anders verlopen. De ICT-oplossing moet aansluiten bij deze situatie. De klant moet in staat zijn om zelf het proces samen te stellen en kunnen kiezen welke deelprocessen ze willen automatiseren door middel van de ICT-oplossing van Stater. Daarnaast kunnen er binnen een land verschillende klanten zien die bijvoorbeeld later willen instappen in het proces of delen van het proces zelf willen uitvoeren.]
Impact	Aangedragen door Wim van Stokkum: Positief: <ul style="list-style-type: none"> • Grotere dienstverlening. • Verbetering commerciële propositie: geldgevers die anders zouden afhaken kunnen nu wel worden aangesloten. Negatief: <ul style="list-style-type: none"> • Ingewikkelde architectuur, dus hogere ontwikkelkosten systeem. Daarnaast negatieve effecten op beheersbaarheid en testbaarheid

	<p>Afkomstig uit architectuurbeschrijving Stater:</p> <ul style="list-style-type: none"> • De keuze voor een <u>process engine</u>, waarin de processen gemodelleerd kunnen worden door het samenstellen van een procesmodel uit losse, onafhankelijke diensten. De process engine is in staat het procesmodel uit te voeren en daarbij de regie over het proces te voeren. • De keuze voor de <u>integration bus</u>, waardoor het transparant wordt of interne dan wel externe services worden aangeroepen. • Het opdelen van de geautomatiseerde ondersteuning in herbruikbare, losse, onafhankelijke services.
Deployment	Communicate

Productflexibiliteit

General	
Short name	Productflexibiliteit
Description	Er mogen geen technische redenen aanwezig zijn die een bepaalde combinatie van productkenmerken verhinderen. Legale combinaties van productkenmerken worden via instellingen/inregelingen gedefinieerd. De ICT-oplossing moet voorbereid zijn op het toevoegen van rekenregels zonder aanpassing van bestaande software. De flexibiliteit van de ICT-oplossing moet zodanig zijn dat niet-woninggebonden leningen ondersteund kunnen worden.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Strategy
Construction abstraction	Construction
Physical abstraction	Conceptual
Enablement abstraction	Application
Organisational range	Information system
Aspects of dynamic systems	Passive structure
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Always
Obedience level	Strict
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater • Estate programma (het informatiesysteem)
Context	
Motivation	<p>Regulating propositions (-)</p> <p>[In de hypotheek branche worden constant nieuwe producten bedacht en de ICT-oplossing moet hier op voorbereid zijn en hier flexibel mee kunnen om gaan. De onderstaande doelen dienen als motivatie voor dit principe (Afkomstig uit de Stater Architectuurbeschrijving):</p> <p>Strategisch:</p> <ul style="list-style-type: none"> • Stater wil klanten flexibele oplossingen bieden met een kostenefficiënte dienstverlening. • Stater wil de ondersteuning van nieuwe klanten, nieuwe producten en nieuwe landen, evenals het doorvoeren van wijzigingen van bestaande ondersteuning, sneller kunnen realiseren. <p>Operationeel:</p> <ul style="list-style-type: none"> • Nieuwe dienstverlening moet op een effectieve en kostenefficiënte wijze kunnen worden geïmplementeerd. <p>]</p>
Impact	<p>Afkomstig uit de Stater Architectuurbeschrijving:</p> <ul style="list-style-type: none"> • Al het gedrag in de ICT-oplossing dat afhankelijk is van productparameters wordt daadwerkelijk gestuurd door inregelingen op één centrale plek: De product services.

	<ul style="list-style-type: none"> • De bouwstenen voor de functionele verwerking van producten en de daaraan gerelateerde inregelingen worden gedefinieerd in het Stater domeinmodel. <p>Aangedragen door Wim van Stokkum:</p> <ul style="list-style-type: none"> • Dus ook aanvullende interfaces tussen productmodel en applicatie. Gescheiden beheerorganisatie. Die van beheerparameters en die van functionaliteit op mid-office.
Deployment	Communicate

Flexibele tijdslijnen

General	
Short name	Flexibele tijdslijnen
Description	De periodieke (administratieve) afsluiting kan per individuele financiële afspraak ingeregeld worden. Dagelijks vindt afsluiting van relevante afspraken plaats.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Operations
Construction abstraction	Function
Physical abstraction	Conceptual
Enablement abstraction	Application
Organisational range	Ecology
Aspects of dynamic systems	Behaviour
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Always
Obedience level	Strict
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater & Klant • Estate programma (het informatiesysteem)
Context	
Motivation	<p>Regulating propositions (-)</p> <p>[De klant moet de mogelijkheid hebben om zelf te bepalen wanneer een afsluiting plaats vindt. De motivatie voor dit principe is het bereiken van het algemene doel: flexibiliteit. De onderstaande doelen dienen als motivatie voor dit principe (Afkomstig uit de Stater Architectuurbeschrijving):</p> <p>Strategisch:</p> <ul style="list-style-type: none"> • Stater wil klanten flexibele oplossingen bieden met een kostenefficiënte dienstverlening. • Stater wil de ondersteuning van nieuwe klanten, nieuwe producten en nieuwe landen, evenals het doorvoeren van wijzigingen van bestaande ondersteuning, sneller kunnen realiseren. <p>]</p>
Impact	<p>Afkomstig uit de Stater Architectuurbeschrijving:</p> <ul style="list-style-type: none"> • De uitwerking van dit principe mag geen negatieve invloed hebben op de beschikbaarheid van het systeem. <p>Aangedragen door Wim van Stokkum:</p> <ul style="list-style-type: none"> • Dus eis 24/7 beschikbaarheid. Hieraan hangt een behoorlijk prijskaartje aan.
Deployment	Communicate

Billing

General	
Short name	Billing
Description	De ICT-oplossing maakt het mogelijk om op basis van een verzameling gebruiksgegevens verschillende vormen van tarifiering aan te bieden.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Strategy
Construction abstraction	Valuation
Physical abstraction	Conceptual
Enablement abstraction	Information
Organisational range	Information system
Aspects of dynamic systems	Behaviour
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Always
Obedience level	Strict
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater • Estate programma (het informatiesysteem)
Context	
Motivation	Regulating propositions (-) [In het huidige systeem dat Stater gebruikt is de tarifiering niet in verhouding met de mate van inspanning van de activiteit. Dit principe is opgesteld omdat het belangrijk is om een tarifiering te kunnen bieden die aansluit bij de geleverde inspanning. Dit principe is opgesteld vanuit het strategische doel; Stater wil klanten flexibele oplossingen bieden met een kostenefficiënte dienstverlening.]
Impact	<p>Afkomstig uit de Stater Architectuurbeschrijving:</p> <ul style="list-style-type: none"> • De modules van de ICT-oplossing produceren voldoende fijnmazige gebruiksgegevens die flexibele billing mogelijk maken. <p>Aangedragen door Wim van Stokkum:</p> <ul style="list-style-type: none"> • Dus heel veel loggen en complexe rapportage/billing algoritme. Verlies van performance.
Deployment	Communicate

Systemmodulariteit

General	
Short name	Systeemmodulariteit
Description	De ICT-oplossing is onderverdeeld in diensten/modules die onderling zijn ontkoppeld en in isolatie kunnen worden afgenomen.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Strategy
Construction abstraction	Construction
Physical abstraction	Conceptual
Enablement abstraction	Application
Organisational range	Information system
Aspects of dynamic systems	Active structure
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Usually
Obedience level	Overridable
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater & Klant • Estate programma (het informatiesysteem)
Context	
Motivation	Regulating propositions (risk-based) [risico is hier dat er klanten (omzet) misgelopen kunnen worden als de klanten alleen de keuze zouden hebben om een compleet systeem af te moeten nemen. Dezelfde motivatie als bij procesmodulariteit is hier van toepassing. Stater wil een flexibele ICT-oplossing aan bieden en het mogelijk maken om de klanten delen van het proces te laten automatiseren door middel van de ICT-oplossing. Door de verschillende delen van het proces (een deel wordt hier dienst/module genoemd) als losstaande module aan te bieden wordt dit mogelijk. Dit principe is opgesteld vanuit het strategische doel; Stater wil klanten flexibele oplossingen bieden met een kostenefficiënte dienstverlening.]
Impact	Aangedragen door Wim van Stokkum: <ul style="list-style-type: none"> • Complexere architectuur. • Verlies van efficiency, hogere ontwikkelkosten. Echter wel minder beheerskosten.
Deployment	Communicate

Inregelbaarheid

General	
Short name	Inregelbaarheid
Description	De ICT-oplossing heeft meerdere niveaus van inregelbaarheid, waarbij de definities op generiekere niveaus overdraagbaar zijn naar specifiekere niveaus. Zo'n niveau wordt een <u>aspectlevel</u> genoemd. De basis voor dit model is onderscheid tussen generieke inregelingen, landspecifieke inregelingen, klantspecifieke inregelingen.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Strategy
Construction abstraction	Construction
Physical abstraction	Conceptual
Enablement abstraction	Application
Organisational range	Information system
Aspects of dynamic systems	Behaviour
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Always
Obedience level	Strict
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater • Estate programma (het informatiesysteem)
Context	
Motivation	Regulating propositions (risk-based) [De kosten van onderhoud en het doorvoeren van wijzigingen moeten zo weinig mogelijk tijd en kosten met zich meebrengen. Het idee achter dit principe is dat configuratie op een abstracter niveau (bijv. 'land') wordt overerft op een concreter niveau (bijv. 'geldgever') waardoor de configuratie-items beter onderhoudbaar zijn.]
Impact	Aangedragen door Wim van Stokkum: <ul style="list-style-type: none"> • Elke component in de architectuur moet dit principe kunnen invullen, anders ontstaat er een zwakke schakel. Niet elke standaard oplossing/tool kan echter hiermee omgaan. Beperkt de keuze van standaard oplossingen.
Deployment	Communicate

Ketenintegratie

General	
Short name	Ketenintegratie
Description	De choreografie van de activiteiten in het totale hypotheekproces is per klant inregelbaar en kan zowel intern als extern zijn belegd.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Strategy
Construction abstraction	Construction
Physical abstraction	Conceptual
Enablement abstraction	Business
Organisational range	Ecology
Aspects of dynamic systems	Active structure
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Always
Obedience level	Strict
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater & Klant • Estate programma (het informatiesysteem)
Context	
Motivation	Regulating propositions (Refinement-based) [Dit principe is een uitbreiding op en een aanvulling van <u>procesmodulariteit</u> en <u>systeemmodulariteit</u> . Dit principe is daarnaast opgesteld vanuit het strategische doel; Stater wil klanten flexibele oplossingen bieden met een kostenefficiënte dienstverlening.]
Impact	Afkomstig uit architectuurbeschrijving Stater: <ul style="list-style-type: none"> • De samenstelling van activiteiten in de keten kan variëren per klant; additionele activiteiten en dus modules zijn mogelijk. • De configuratie en aansturing van de modules in een keten is gescheiden van de modules en instelbaar per keten. • Informatie over de processen en de bijbehorende gegevens moet uitwisselbaar zijn. Dit komt in de Functionele Applicatiearchitectuur tot uiting in de proces engine waar de procesdefinities belegd zijn en de €LF waarmee informatie-uitwisseling gefaciliteerd wordt.
Deployment	Communicate

Straight Through Processing

General	
Short name	Straight Through Processing
Description	De verwerking van aangeboden verzoeken (zoals aanvragen) dient door de ICT-oplossing automatisch uitgevoerd te kunnen worden.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Operations
Construction abstraction	Function
Physical abstraction	Conceptual
Enablement abstraction	Application
Organisational range	Information system
Aspects of dynamic systems	Active structure
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Sometimes
Obedience level	Overridable
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater • Estate programma (het informatiesysteem)
Context	
Motivation	Regulating propositions (risk-based) [Kostenefficiëntie. Verzoeken/activiteiten die automatisch afgehandeld kunnen worden zouden ook automatisch afgehandeld moeten worden. Deze activiteiten kunnen afgehandeld worden zonder dat hier menselijke handelingen aan te pas komen. Dit scheelt tijd en geld. De achterliggende motivatie is afkomstig van het strategische doel; Stater wil de volledige automatische verwerking van businessprocessen optimaliseren (Straight Through Processing).]
Impact	Afkomstig uit architectuurbeschrijving Stater: Dit principe stelt vooral eisen aan de inrichting van processen waarvoor STP gewenst is. Processen zullen dan gebruik moeten maken van geautomatiseerde services. Foutsignalering (uitval door bijvoorbeeld incomplete gegevens), traceerbaarheid (waar is het fout gegaan?) en correctie/herstartmogelijkheden zijn hierbij belangrijke aandachtspunten. Deze aandachtspunten gelden voor de hele keten (en daarmee ook voor externe partijen).
Deployment	Communicate

Beschikbaarheid

General	
Short name	Beschikbaarheid
Description	De ICT-oplossing moet zodanig ingericht zijn dat er geen technische beperkingen zijn aan de beschikbaarheid ervan voor geautoriseerde gebruikers en processen.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Structuring
Construction abstraction	Valuation
Physical abstraction	Conceptual
Enablement abstraction	Application / Infrastructure
Organisational range	Information system
Aspects of dynamic systems	Active structure
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Usually
Obedience level	Overridable
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater • Estate programma (het informatiesysteem)
Context	
Motivation	Regulating propositions (risk-based) [Het risico dat de ICT-oplossing niet bereikbaar is wordt hier mee afgevangen. De ICT-oplossing moet zoveel mogelijk beschikbaar zijn om operationele processen niet te verstoren.]
Impact	Afkomstig uit architectuurbeschrijving Stater: <ul style="list-style-type: none"> • Dit principe heeft een relatie met schaalbaarheid. • De online beschikbaarheid van de ICT-oplossing mag niet beïnvloed worden door batchverwerkingen. • De beschikbaarheid van externe diensten beïnvloedt mogelijk de beschikbaarheid van eigen diensten
Deployment	Communicate

Traceerbaarheid

General	
Short name	Traceerbaarheid
Description	Handelingen van gebruikers en services dienen controleerbaar te zijn en gelogd te worden. Fouten dienen geregistreerd te worden
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Operations
Construction abstraction	Function
Physical abstraction	Conceptual
Enablement abstraction	Information
Organisational range	Enterprise
Aspects of dynamic systems	Behaviour
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Usually
Obedience level	Guiding
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater • Estate programma (het informatiesysteem)
Context	
Motivation	Regulating propositions (Refinement-based) [Dit principe is gedeeltelijk noodzakelijk om te voldoen aan het compliance principe. Bepaalde informatie dient te worden bijgehouden en traceerbaar te zijn om aan de wetgeving te voldoen. Daarnaast is het belangrijk om van fouten of afhandelingen de oorzaak te kunnen achterhalen.]
Impact	<p>Afkomstig uit architectuurbeschrijving Stater:</p> <p>Dit principe komt in de Functionele Applicatiearchitectuur tot uiting door de keuze van de volgende componenten:</p> <ul style="list-style-type: none"> • de system service logging. • de integration bus <p>Aangedragen door Wim van Stokkum:</p> <ul style="list-style-type: none"> • En nog vele andere plaatsen waarin acties en gegevens worden gelogd.
Deployment	Communicate

White-/private-labeling

General	
Short name	White-/private-labeling
Description	Alle user-interfaces en alle brieven en rapportages van alle modules kunnen worden ingesteld naar het uiterlijk en de wensen van de klant.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Strategy
Construction abstraction	Function
Physical abstraction	Conceptual
Enablement abstraction	Application
Organisational range	Information system
Aspects of dynamic systems	Passive structure
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Usually
Obedience level	Overridable
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater & Klant • Estate programma (het informatiesysteem)
Context	
Motivation	Regulating propositions (-) [De klanten van Stater maken gebruik van de ICT-oplossing en de ICT-oplossing moet aansluiten bij de 'look-and-feel' van de klant. Op alle output die de ICT-oplossing genereerd is het van belang dat het duidelijk is dat het afkomstig is van de klant. Activiteiten kunnen intern of extern belegd zijn. Ten aller tijden moet echter zichtbaar zijn voor wie de activiteit uitgevoerd wordt en dat de output voorzien is van de juiste labeling.]
Impact	<p>Afkomstig uit architectuurbeschrijving Stater:</p> <p>In de Functionele Applicatiearchitectuur heeft dit impact op de functionele modules, de rapportages, de brieven en het eventuele gebruik van communicatiekanalen (bijvoorbeeld email).</p> <p>Aangedragen door Wim van Stokkum:</p> <ul style="list-style-type: none"> • Maakt systeemontwerp enorm veel complexer en duurder te realiseren
Deployment	Communicate

Rapportage

General	
Short name	Rapportage
Description	De ICT-oplossing bevat geen ingebouwde kennis over rapportages. De rapportage functionaliteit is ontkoppeld van de geautomatiseerde ondersteuning van het primaire proces. Alle geautoriseerde partijen krijgen toegang tot hun rapportagedata, in de door hen gewenste vorm van doorsneden en aggregaties.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Structuring
Construction abstraction	Function
Physical abstraction	Logical
Enablement abstraction	Application
Organisational range	Information system
Aspects of dynamic systems	Passive structure
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Always
Obedience level	Strict
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater & Klant • Estate programma (het informatiesysteem)
Context	
Motivation	Regulating propositions (refinement-based) [Er is in het project een ontwerp beslissing genomen dat geen enkele rapportage wordt aangemaakt vanuit het systeem, maar dat dit wordt aangeleverd aan een anoniem datawarehouse-achtige rapportage interface.]
Impact	Afkomstig uit architectuurbeschrijving Stater: In de Functionele Applicatiearchitectuur komt dit principe tot uiting in het datawarehouse dat de primaire bron is voor de rapportages.
Deployment	Communicate

Koppelbaarheid

General	
Short name	Koppelbaarheid
Description	De ICT-oplossing moet eenvoudig kunnen worden gekoppeld aan andere systemen, softwarepakketten, database en netwerken.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Strategy
Construction abstraction	Construction
Physical abstraction	Conceptual
Enablement abstraction	Application
Organisational range	Information system
Aspects of dynamic systems	Active structure
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Usually
Obedience level	Overridable
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater & Klant • Estate programma (het informatiesysteem)
Context	
Motivation	Regulating propositions (refinement-based) [Stater heeft verschillende klanten in meerdere landen. De manier waarop het hypotheekstelsel werkt in deze landen verschilt onderling. In elk land en voor elke klant kan het proces anders verlopen. De ICT-oplossing moet aansluiten bij deze situatie. De klant moet in staat zijn om zelf het proces samen te stellen en kunnen kiezen welke deelprocessen ze willen automatiseren door middel van de ICT-oplossing van Stater. Daarnaast wordt in het kader van keten integratie verwacht dat er steeds meer koppelingen plaats zullen gaan vinden. Dit principe is refinement-based (op basis van het principe Procesmodulariteit).]
Impact	<p>Aangedragen door Wim van Stokkum:</p> <p>Positief:</p> <ul style="list-style-type: none"> • Verdere ketenintegratie. • Het kunnen integreren van klant systemen (commercieel sterk). <p>Negatief:</p> <ul style="list-style-type: none"> • Geen koppelen moet je altijd. • Elke koppeling is toename van complexiteit en heeft impact op beheersbaarheid.
Deployment	Communicate

Multi-channeling

General	
Short name	Multi-channeling
Description	De modules van de ICT-oplossing kunnen via diverse (technische) kanalen worden gebruikt, en worden niet kanaalspecifiek geïmplementeerd. De modules kunnen zowel interactief (via een user-interface) als programmatisch worden gebruikt.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Strategy
Construction abstraction	Construction
Physical abstraction	Logical
Enablement abstraction	Infrastructure
Organisational range	Information system
Aspects of dynamic systems	Active structure
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Usually
Obedience level	Overridable
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater • Estate programma (het informatiesysteem)
Context	
Motivation	Regulating propositions (refinement-based) [De ICT-oplossing moet flexibel zijn. Dit principe is een verfijning van verschillende principes die met flexibiliteit te maken hebben zoals systeemmodulariteit.]
Impact	Afkomstig uit architectuurbeschrijving Stater: Dit betekent dat elke modules zowel een service interface moet bieden als (voor zover relevant) een apart gescheiden user-interface. Uitgangspunt hierbij is dat er gebruik gemaakt wordt van internetstandaarden.
Deployment	Communicate

Autorisatie en Beveiliging

General	
Short name	Autorisatie en Beveiliging
Description	Toegang tot functies en gegevens van de ICT-oplossing dient op verschillende niveaus (gebruiker, geldgever, tussenpersoon, etc.) te worden geautoriseerd. Hierbij spelen zaken als single logon, beveiligde interacties tussen de partijen en 'Identity and access management'.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Structuring
Construction abstraction	Construction
Physical abstraction	Conceptual
Enablement abstraction	Application
Organisational range	Information system
Aspects of dynamic systems	Active structure
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Always
Obedience level	Strict
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater & Klant • Estate programma (het informatiesysteem)
Context	
Motivation	Regulating propositions (risk-based) [Bepaalde activiteiten mogen alleen door de juiste personen uitgevoerd worden.]
Impact	Afkomstig uit architectuurbeschrijving Stater: Dit principe komt in de Functionele Applicatiearchitectuur tot uiting door de keuze van de volgende componenten: <ul style="list-style-type: none"> • de access controller • authorization modeler
Deployment	Communicate

Meertaligheid

General	
Short name	Meertaligheid
Description	De ICT-oplossing biedt ondersteuning voor meerdere talen – zowel interactief, in rapportages als in ondersteuning (Help functies). De gekozen taal kan afhankelijk zijn van verschillende aspecten (gebruiker, klant, land, etc.)
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Strategy
Construction abstraction	Construction
Physical abstraction	Conceptual
Enablement abstraction	Application
Organisational range	Information system
Aspects of dynamic systems	Passive structure
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Usually
Obedience level	Overridable
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater & Klant • Estate programma (het informatiesysteem)
Context	
Motivation	Regulating propositions (risk-based) [Stater begeeft zich o.a. op de Europese markt. Om de ICT-oplossing in te kunnen zetten voor klanten in deze landen is het noodzakelijk dat de ICT-oplossing in meerdere talen te gebruiken is.]
Impact	<p>Afkomstig uit architectuurbeschrijving Stater:</p> <p>Meertaligheid komt terug in de user-interfaces, brieven en rapportages. Aandachtspunt hierbij is dat er meerdere combinaties van talen binnen één omgeving aanwezig zijn. Bijvoorbeeld: Een Nederlandstalige consument koopt een huis in Frankrijk met een Nederlandse hypotheek te passeren bij een Franse notaris.</p> <p>Aangedragen door Wim van Stokkum:</p> <p>Alle teksten, foutboodschappen, schermlabels, helpteksten etc. worden allemaal in een content management systeem opgeslagen en dynamisch opgehaald bij opbouw scherm/berichten.</p>
Deployment	Communicate

Multicurrency

General	
Short name	Multicurrency
Description	De ICT-oplossing biedt ondersteuning voor meerdere valuta – zowel interactief, in rapportages als in ondersteuning (Help functies). Binnen één omgeving (VCE) van een geldgever wordt met één valuta gewerkt.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Strategy
Construction abstraction	Construction
Physical abstraction	Conceptual
Enablement abstraction	Application
Organisational range	Information system
Aspects of dynamic systems	Passive structure
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Always
Obedience level	Strict
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater & Klant • Estate programma (het informatiesysteem)
Context	
Motivation	Regulating propositions (risk-based) [Omdat Stater in verschillende landen actief is waar verschillende valuta gebruikt worden is het van belang dat de ICT-oplossing dit gegeven ondersteunt.]
Impact	Aangedragen door Wim van Stokkum: Niet elementair datatype: currency. Deze heeft het gedrag van de float, echter het datatype zelf werkt met een base currency en conversionrates.
Deployment	Communicate

Compliance

General	
Short name	Compliance
Description	De ICT-oplossing conformeert zich aan nationale en internationale wet- en regelgeving (IAS/IFRS, BASEL II, WFD, SOXA, SAS70).
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Actionable
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Structuring
Construction abstraction	Valuation
Physical abstraction	Conceptual
Enablement abstraction	Information
Organisational range	Information system
Aspects of dynamic systems	Behaviour
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Always
Obedience level	Strict
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater • Estate programma (het informatiesysteem)
Context	
Motivation	<p>Regulating propositions (risk-based)</p> <p>[Stater heeft verschillende klanten in meerdere landen, zowel in Europa als de Verenigde Staten. In deze landen zijn verschillende wetgevingen van kracht en Stater is verplicht om zich hier aan te houden. De ICT-oplossing moet daarom voldoen aan de eisen die gesteld worden door deze wetgevingen en moet in staat zijn om de voor de betreffende wetgevingen geëiste informatie te kunnen leveren. Afkomstig uit 'Stater – een niet-lineaire architectuur beschrijving':</p> <p>Dit betekent dat het systeem de door deze wetgeving geëiste informatie moet kunnen leveren. Zo stelt SAS70 bijvoorbeeld eisen aan de informatie die gelogd dient te worden gedurende het gebruik van het systeem.]</p>
Impact	<p>Afkomstig uit architectuurbeschrijving Stater:</p> <p>Extra informatie dient gelogd te worden om te voldoen aan de nationale- en internationale wet- en regelgeving.</p> <p>Aangedragen door Wim van Stokkum:</p> <p>Positief:</p> <ul style="list-style-type: none"> • Commercieel is transparantie een positief aspect. <p>Negatief:</p> <ul style="list-style-type: none"> • Performance: extra verslaglegging en negatief complexiteit: de systeemontwikkeling wordt complexer.
Deployment	Communicate

Schaalbaarheid

General	
Short name	Schaalbaarheid
Description	De ICT-oplossing is in staat meerdere miljoenen leningen te beheren zonder substantieel verlies van online- en batchperformance. Schaalbaarheid heeft ook betrekking op andere aspecten, waaronder aantal landen, aantal producten, aantal rapportages, aantal gebruikers, etc.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Operations
Construction abstraction	Construction
Physical abstraction	Conceptual
Enablement abstraction	Application
Organisational range	Information system
Aspects of dynamic systems	Behaviour
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Always
Obedience level	Strict
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater • Estate programma (het informatiesysteem)
Context	
Motivation	Regulating propositions (risk-based) [<ul style="list-style-type: none"> • Estate moet schaalbaar zijn, zodat een groei naar meerdere miljoenen te beheren leningen, zonder grote technische ingrepen, mogelijk is. • De technische performance van de ICT-oplossing moet blijvend acceptabel zijn.]
Impact	Afkomstig uit architectuurbeschrijving Stater: Concreet betekent dit dat services parallel uitgevoerd moeten kunnen worden waardoor dezelfde functionaliteit voor verschillende processen tegelijkertijd beschikbaar is. Dit wordt bereikt door het inzetten van extra hardware componenten (Meerdere applicationsservers op een of meerdere machines. Meerdere applicaties te verdelen over meerdere applicationsservers.)
Deployment	Communicate

Onderhoudbaarheid

General	
Short name	Onderhoudbaarheid
Description	De structuur van de ICT-oplossing is zodanig opgezet dat wijzigingen beperkt blijven tot een minimaal aantal componenten en eenvoudig zijn door te voeren.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Strategy
Construction abstraction	Construction
Physical abstraction	Conceptual
Enablement abstraction	Application
Organisational range	Information system
Aspects of dynamic systems	Passive structure
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Always
Obedience level	Strict
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater • Estate programma (het informatiesysteem)
Context	
Motivation	<p>Regulating propositions (risk-based)</p> <p>[Onderhoud van ICT-voorzieningen brengen grote kosten met zich mee. Door dit gegeven is het belangrijk dat de onderhoudbaarheid van de ICT-oplossing optimaal is.</p> <ul style="list-style-type: none"> • Stater wil de ondersteuning van nieuwe klanten, nieuwe producten en nieuwe landen, evenals het doorvoeren van wijzigingen van bestaande ondersteuning, sneller kunnen realiseren. • Stater richt zich op verdere vermindering van de ICT-kosten.]
Impact	<p>Afkomstig uit architectuurbeschrijving Stater:</p> <p>Dit komt terug in de inrichting van de <u>Functionele Applicatiearchitectuur</u> door de splitsing in aparte componenten en het onderscheiden van processengine en product services.</p> <p>Aangedragen door Wim van Stokkum:</p> <p>Daarnaast een heel stelsel aan maatregelen in document vorm. Probleem is alleen dat het nooit genoeg is en dat je meerdere doelgroepen hebt.</p>
Deployment	Communicate

Beheerbaarheid

General	
Short name	Beheerbaarheid
Description	De kosten en benodigde inspanning voor het beheren van de ICT-oplossing dienen minimaal te zijn.
Source	Stater Architectuurbeschrijving V1.1
Form	
Level of precision	Informal
Level of actionability	Definite
Object	
Time	Het principe is van toepassing/geldig zolang het Estate-programma in ontwikkeling en/of in gebruik is.
Control abstraction	Strategy
Construction abstraction	Construction
Physical abstraction	Conceptual
Enablement abstraction	Application
Organisational range	Business unit
Aspects of dynamic systems	-
Systemic order	Operational system
Validity	
Intricity	Desired
Probability	Always
Obedience level	Strict
Actors	
Actors	<ul style="list-style-type: none"> • Architectuurteam & ontwerpteam. • Stater • Estate programma (het informatiesysteem)
Context	
Motivation	<p>Regulating propositions (risk-based)</p> <p>[Net zoals het onderhoud van ICT-voorzieningen brengt ook het beheer hiervan grote kosten met zich mee. Door dit gegeven is het belangrijk dat het beheer van de ICT-oplossing zo min mogelijk tijd en kosten met zich meebrengt.]</p> <ul style="list-style-type: none"> • Stater richt zich op verdere vermindering van de ICT-kosten.]
Impact	<p>Afkomstig uit architectuurbeschrijving Stater:</p> <ul style="list-style-type: none"> • Vanuit technisch perspectief betekent dit één beheerlocatie, één installatie en uniforme standaardtechnologie. • Beheerbaarheid zal een stempel drukken op het aankoopbeleid van platformen en ontwikkel- en deployment-omgevingen • Beheerbaarheid zal een stempel drukken op het ontwerpen, bouwen en testen van software. • De verwachting is dat dit principe een positief effect heeft op de kosten. (verlaging van de TCO.)
Deployment	Communicate

Bijlage C: herformuleren architectuurprincipes

Inleiding

In deze bijlage wordt een reeks van experimenten belicht die de basis en start vormen van deel 2 van deze afstudeerscriptie. Er wordt kort ingegaan op de betreffende casus (Stater) en de algemene problemen die onderkend zijn bij het gebruiken van de huidige architectuurprincipes.

Architectuurproces Stater NV

In dit hoofdstuk zal in grote lijnen het architectuurproces van Stater NV worden beschreven. Hiermee wordt een beeld geschetst van de wijze waarop zij architectuur zien, wat men verstaat onder architectuurprincipes, hoe men om gaat met architectuur en hoe dit alles zich tot nu toe heeft ontwikkeld binnen Stater NV.

Architectuur binnen Stater

Architectuur is iets nieuws binnen Stater NV. Een kleine groep mensen zijn dit aan het opzetten. De achtergrond van deze mensen is systeemontwikkeling. De motivatie achter het willen werken onder architectuur was dat men een ICT-oplossing wilde ontwikkelen die meer flexibiliteit en transparantie bewerkstelligt en minder beheer en onderhoudskosten heeft.

De architectuurprincipes vormen een verzameling van principes uit cursussen OO en CBD en zijn gecombineerd met actuele problemen. Er is gekeken naar OO, Archimate. Ze zijn begeleid door het kenniscentrum Cibit. De definitie die gehanteerd is voor architectuurprincipe is afkomstig van DYA.

Binnen Stater NV zijn ze zoekende geweest. De ontwikkeling van het gedachtegoed architectuur heeft parallel plaatsgevonden met vernieuwing van het systeem. Het is dus een leerproces geweest met afstemmingsproblemen gedurende het traject.

Werken onder architectuur is ook voor de Business van Stater nog onwennig, alsmede ook voor de ontwikkelaars van het huidige systeem iSHS. Het nut is niet altijd even duidelijk. Architectuur is iets abstracts. Het architectuurteam heeft door middel van presentaties proberen duidelijk te maken wat ze aan doen zijn en wat de meerwaarde is.

De algemene principes zijn aan het begin van het project mondeling met elkaar afgestemd in workshops. Er is mondeling verzekerd van het feit dat iedereen de principes hetzelfde geïnterpreteerd heeft. Het project dat verantwoordelijk is voor de nieuwe ICT-oplossing heeft de principes zelf vertaald naar bouw richtlijnen en principes, die ontwikkelaars zelf altijd in praktijk brengen bij diverse projecten.

Tijdens het traject heeft regelmatig afstemming plaats gevonden en is de architectuur meegenomen in de ontwikkeling. Er is getoond hoe de architectuurrichtlijnen in praktijk gebracht dienen te worden. Daarnaast hebben regelmatig audits plaatsgevonden van externe architectuur organisaties als Gartner. Voor deze audits zijn echter niet de Stater principes gebruikt, maar hun eigen frameworks. Ze hebben hierbij echt gekeken naar de architectuur, de opzet van componenten, en ook naar de programmatuur zelf. Ook de projectorganisatie is belicht.

Daarnaast hebben de architecten van Stater zelf laat in het traject een audit uitgevoerd o.b.v de architectuurprincipes die ze tot dan toe hadden opgesteld.

Algemene observaties en knelpunten binnen architectuurproces Stater NV

De volgende observaties en knelpunten zijn te benoemen:

- **Tijdigheid:** Uitwerking van de architectuurprincipes niet gereed of beschikbaar bij aanvang groot automatiseringstraject. Hierdoor loopt architectuur achter de feiten aan.
- **Mate van detail:** Uitvoerende partij heeft veel interpretatie ruimte. Dit is goed opgepakt door het ervaren ontwikkelteam, maar had bij een ander partij makkelijk tot grote afwijkingen kunnen leiden.

- Onduidelijkheid semantiek: Maandelijks veranderende betekenis van bepaalde begrippen en de daarbij gestelde doelstellingen. Men is tijdens het traject nog zoekende.
- Architectuur wordt uitgelegd o.b.v. prototypes in .Net. Erg kostbaar en niet echt effectief
- Geen verankering architectuur in business Stater. D.w.z dat het nastreven van een bepaald architectuurprincipe zou leiden tot onacceptabele hoge doorlooptijd en kosten voor aspecten die nu commercieel door business niet als belangrijk of haalbaar worden geacht. M.a.w. men heeft getracht geprobeerd door middel van de opgestelde architectuur zich in te dekken voor alle mogelijke toekomstige wellicht handige situaties.
- Vanwege niet op tijd en concreet hebben richtlijnen kwamen requirements/architectuur richtlijnen achteraf. Kost veel doorlooptijd en kosten om deze door middel van rework weer in te voeren.
- Geen concrete en op de praktijk afgestemde richtlijnen hebben bij ontwikkelteam, dat verantwoordelijk is voor middleware, tot ingrijpend rework geleid. Consequentie vertraging en extra kosten. Dit is rechtstreeks het gevolg van onvolledige formulering architectuurprincipes en pas achteraf aanleveren aanvullende onverwachte requirements.
- Weinig pragmatisch: D.w.z dat veel nadruk wordt gelegd op het realiseren van flexibiliteit door middel van architectuurprincipes, terwijl onzeker is of dit ooit nodig is en waarvan duidelijk is dat eerste X klanten er weinig behoefte aan hebben.
- De architecten waren niet nauw genoeg betrokken in het project waardoor men de feeling verloor met praktijk problemen, business context en project problematiek en daardoor niet kunnen sturen en het proces eigenlijk alleen frustreren ondanks goede bedoelingen.

Dragon1 Stappenplan voor het Opstellen, Formuleren en Normaliseren van Principes v0.10c

Om een beter inzicht te verkrijgen in het herformuleren van architectuurprincipes is gekozen om een architectuurprincipe van Stater te herformuleren door de hierboven genoemde versie van het Dragon1 Stappenplan voor het opstellen, formuleren en normaliseren van principes. Dit experiment is tevens uitgevoerd om meer inzicht te verkrijgen omtrent de problematiek die kan optreden bij het formuleren van architectuurprincipes. De onderstaande uitwerking van het architectuurprincipe 'Procesmodulariteit' is in samenwerking met Wim van Stokkum tot stand gekomen.

Experiment

Attribuut	Beschrijving
Universele ID	PRINC_PROCES_1
Universele naam	Het procesmodulariteitsprincipe voor Stater informatiesystemen
Systeem, concept of stijlverwijzing	Binnen een informatiesysteem moeten procesdefinities gescheiden zijn van domeingegevens structuur definities en van definities van functionaliteit of Systeem
Omschrijving	<p>Hieronder volgt de eerste uitwerking van het principe:</p> <p>Doordat:</p> <p>1.Processen modulair zijn opgebouwd. Processen bestaan uit sequenties van processtappen. Processtappen bestaan op hun beurt uit sequenties van handelingen. Handelingen vormen de kleinste eenheid van procesdefinities en zijn daarom ondeelbaar.</p> <p>2. Processen, processtappen en handelingen geen "kennis" van andere processen, processtappen en handelingen, dit wordt stateless genoemd. Met kennis bedoelen wij dat het niet veronderstelt dat een ander proces/processtap/handeling een bewerking heeft gedaan.</p> <p>Kunnen</p> <ol style="list-style-type: none"> 1. processen, processtappen en handelingen worden hergebruikt 2. processen processtappen en handelingen worden belegd bij interne of externe partij naar keuze. <p>2^e uitwerking van het principe:</p> <p>Het procesmodulariteitsprincipe zegt dat door processen modulair op te bouwen en processen, processtappen en handelingen stateless zijn dat processen, processtappen en handelingen hergebruikt worden en belegd worden bij een interne of externe partij naar keuze.</p>
Klasse	Structuurprincipe
Type (activiteit)	Constructieprincipe
Soort beschouwingsniveau	<p>Vanuit bedrijf gezien: bedrijfsactiviteitprincipe. De activiteit is dan het ontwikkelen van een informatiesysteem.</p> <p>Vanuit systeem gezien: CTI-Infrastructuurprincipe: het beschrijft de opbouw van de structuur van een ICT systeem.</p>
Eigenaar/beheerder	Eigenaar is de business architectuur unit van Stater.
Issues en conflicten	<p>Haken en ogen:</p> <p>Dit principe dient door alle lagen van een informatiesysteem te worden doorgevoerd.</p>

D.w.z. op procesniveau, op processtap niveau maar ook binnen handelingniveau. Het onafhankelijk maken van de procesitems heeft als consequentie dat:

- a) Proceselement A geen kennis mag hebben van de gegevensstructuur van proceselement B. Dit betekent dat er geen globaal datamodel mag zijn binnen de applicatie, maar dat gegevens moeten worden uitgewisseld middels een algemeen communicatieprotocol, waarbij elk element zijn eigen vertaling maakt naar zijn eigen datastructuur.
- b) Binnen elementen mogen bepaald activiteiten niet uitgevoerd worden of worden bepaalde activiteiten anders uitgevoerd afhankelijk van de status van de case die wordt afgehandeld. De handeling " binnenboeken van ingekomen documenten" kunnen aanvullende stukken als bewijslast worden opgevoerd. Echter dit kan niet meer nadat de lening geaccepteerd is en de case zich in het notaristraject bevindt. De verleiding is om dan te sturen op processtap. Dus als huidige processtap = notaristraject dan.... Echter dit is fout. Gestuurd moet worden op status. Dus indien lening.status = geaccepteerd.
- c) Er iets geregeld moet worden m.b.t. kwaliteit van gegevensoverdracht. In principe kan elk element extern worden belegd. De controle over de gegevens komt daarmee dan extern te liggen. Nadat het externe proces is afgerond wordt vervolgens het proces weer intern voortgezet. Het issue is dan de kwaliteit (volledigheid en consistentie) van de overgedragen gegevens.

Probleem met toepassen bovenstaande voorwaarden is:

Bij a) Er moet een onafhankelijk communicatieprotocol worden ontwikkeld en worden beheerd. Dit is een extra beheerslast. Daarnaast is de transformatie nodig bij elke service aanroep per element. Dit maakt de systeemontwikkeling moeilijker en heeft negatieve effecten op de doorlooptijd van het systeemontwikkelingsproject. Aangezien de ontkoppeling op elk element moet plaatsvinden (dus ook op handeling niveau, heeft dit negatieve effecten op de performance van het systeem.

Bij b): Er dient een gemeenschappelijk statusovergangen model ontwikkeld en centraal beheerd te worden. Is een extra beheerslast. Daarnaast is het maken van condities een stuk lastiger. Het probleem is het vinden van de juiste combinatie van statussen om aan te kunnen duiden waar je bent in de lifecycle van de case. Doorgaans zijn aanvullende interne statussen nodig om dit goed te kunnen regelen.

Bij c) in SLA overeenkomsten moet geregeld worden welke partij aansprakelijk is voor verkeerde gegevens. Het valideren van de consistentie en volledigheid van gegevens is vaak erg duur (qua proces) en heeft dus negatieve effecten op de performance.

Conflicten:

Dit principe is inherent in conflict met het principe beheersbaarheid. De inspanningen voor het beheer van het systeem nemen aanzienlijk toe vanwege:

- Beheer centraal domeinmodel (protocol)
- Beheer statusovergangen
- Beheer proces management systeem omgeving
- Beheer services.

Deze extra beheersinspanningen drukken initieel op de organisatie en dienen terugverdiend te worden door:

- Vermindering inspanningen realisatie (alternatieve) nieuwe processen, vanwege hergebruik
- Reductie inspanningen aanpassen intern gedrag van elementen

	<ul style="list-style-type: none"> - Verkopen afzonderlijke elementen als oplossingen aan potentieel grote klantenkring. Dit i.t.t. verkopen “alles” of “niets” . <p>De regels werken niet in een ICT infrastructuur waarbij:</p> <ol style="list-style-type: none"> 1. de procesgang niet door een aparte procescomponent wordt afgehandeld. Bijvoorbeeld in traditionele systemen waarbij processen activiteiten en gegevens niet gescheiden zijn. 2. de proceselementen niet apart kunnen worden aangeroepen (geactiveerd), door middel van services. 3. omgevingen waarin geen “communicatiebus” is die de interne en externe communicatie tussen elementen kan regelen.
Beschouwingssituatie	Principe moet gaan gelden voor alle toekomstig te realiseren systemen binnen Stater (TO-BE)
Effect-type	De handhaving mechanismen: <ul style="list-style-type: none"> - Gebruik centraal communicatie model - Gebruik statussen ipv verwijzingen naar proces/stap/handelingen - Regel verantwoordelijkheid gegevens in SLA
First principle	Nee
Status	Onderkennen
Bronvermelding	Bewezen algemene architectuurrichtlijnen in kader van : <ul style="list-style-type: none"> - Procesarchitectuur - Service Oriënted Architecture - Communication Core
Rationale	Reden van dit principe: <ol style="list-style-type: none"> 1. Door delen van een proces extern te kunnen beleggen is het mogelijk dat klanten van Stater het systeem in delen kunnen afnemen. Dit heeft als voordelen: <ol style="list-style-type: none"> a) Klanten hebben vaak eigen rekenmodules, beoordelingmodules en andere componenten die zij qua onderhoud graag in eigen beheer houden. Dit wordt mogelijk. b) Klanten willen (delen) systemen kunnen draaien op hun eigen technische infrastructuur. Dit is nu onmogelijk. c) Klanten willen een fijnmazigere knip tussen wat Stater qua proces uitvoert en wat de klanten zelf doen. M.a.w. de klant kan zelf zijn business knip vaststellen. d) Klanten kunnen later in het proces instappen of uitstappen, bijvoorbeeld voor bankacceptatie of niet. e) Stater kan door deze flexibiliteit meer klanten aan zich binden. 2. Hergebruik: <p>Proceselementen kunnen in meerder toepassingen worden hergebruikt waardoor systeemontwikkeltijden kunnen worden gereduceerd en waardoor er onderhoud op een plaats (single point of maintenance) kan worden gerealiseerd.</p>
Veranderdoelen	Dit principe is nieuw binnen Stater. Wordt wellicht incidenteel toegepast binnen afzonderlijke projecten op eigen initiatief. Veranderdoel is dit principe als hoofddoel te laten zijn bij alle toekomstige projecten. <p>Huidige systemen zijn niet flexibel genoeg om hergebruik te kunnen realiseren en ontbreekt het aan mechanismen om delen van het proces extern te kunnen beleggen. Doel is om hier verandering in aan te brengen.</p>
Kwaliteitseisen	Extern kunnen beleggen: <ul style="list-style-type: none"> - Elk proceselement heeft een nauwkeurige beschrijving met pre- en postcondities en service signatuur.

	<ul style="list-style-type: none"> - Elk proceselement moet worden opgenomen in een catalogus. <p>Er dient een implementatiedocument te komen voor de business knip. Waarmee bij aansluiting nieuwe klant precies kan worden ingeregeld welke elementen intern en extern worden belegd.</p>
Bedrijfsvoordelen	<ol style="list-style-type: none"> 1. Reductie ontwikkelkosten informatie systemen door hergebruik. Dit kan op langere termijn nadat er eerst een aantal systemen zijn opgezet volgens het principe. 2. Vergroten commerciële daadkracht, doordat Stater zich flexibeler kan opstellen zullen minder klanten “afzien” van het afnemen van Stater diensten. 3. Vergroten commerciële creativiteit. Door samenstellen (hergebruik) van onderdelen kunnen nieuwe combinaties worden gevormd en deze nieuwe configuraties kunnen als “nieuwe dienst/nieuw product” worden aangeboden.
Implementatie impact	<p>a) proceselement A geen kennis mag hebben van de gegevensstructuur van proceselement B. Dit betekent dat er geen globaal datamodel mag zijn binnen de applicatie, maar dat gegevens moeten worden uitgewisseld middels een algemeen communicatieprotocol, waarbij elk element zijn eigen vertaling maakt naar zijn eigen datastructuur.</p> <ul style="list-style-type: none"> - Kosten 4 man fulltime aan ontwikkeling en beheer = 4 man*200 mandagen *8 uur *75 euro = 480k per jaar - Licentie communicatie tool BizTalk: 1,2 mln - Implementatie Biztalk = 3,2 mln eenmalig <p>b) Binnen elementen mogen bepaald activiteiten niet uitgevoerd worden of worden bepaalde activiteiten anders uitgevoerd afhankelijk van de status van de case die wordt afgehandeld. De handeling “ binnenboeken van ingekomen documenten” kunnen aanvullende stukken als bewijslast worden opgevoerd. Echter dit kan niet meer nadat de lening geaccepteerd is en de case zich in het notaristraject bevind. De verleiding is om dan te sturen op processtap. Dus als huidige processtap = notaristraject dan.... Echter dit is fout. Gestuurd moet worden op status. Dus indien lening.status = geaccepteerd.</p> <ul style="list-style-type: none"> - Beheer status = 0,5 man continue = 0,5 * 200 * 8 * 75= 60 k per jaar <p>c) Er iets geregeld moet worden m.b.t. kwaliteit van gegevensoverdracht. In principe kan elk element extern worden belegd. De controle over de gegevens komt daarmee dan extern te liggen. Nadat het externe proces is afgerond wordt vervolgens het proces weer intern voortgezet. Het issue is dan de kwaliteit (volledigheid en consistentie) van de overgedragen gegevens.</p> <ul style="list-style-type: none"> - Ontwikkelen nieuw SLA = 2 man * 100 dagen * 8 uur * 75 euro = 120k eenmalig - Ontwerp en implementatie validatiemodule = 200 uur * 200 euro * 2 = 80 k eenmalig - Beheer validatie = 0,5 manjaar continue = 60 k <p>d) Beheer procesarchitectuur</p> <ul style="list-style-type: none"> - Elk proceselement heeft een nauwkeurige beschrijving met pre- en postcondities en service signatuur. - Elk proceselement moet worden opgenomen in een catalogus. Ontwikkeling catalogus = 2 man * 6 maanden = 120 k eenmalig Beheer = 60 k continue <p>e) Vergroten doorlooptijd initiële ontwikkelproces vanwege leercurve eerste grote project= 1,2 jaar + 10 man * 200 dagen * 8 * 175 k = 3,3 mln</p>

	<p>Leerkosten = 3,3 mln Totaal eenmalig = 120k + 80 k +120 k + 4,4 mln = 4,72 mln Terugkerend = 60k + 60 K + 60 K + 480 k = 660 k</p> <p>Opmerking Wim van Stokkum:</p> <p>Dit attribuut kent erg veel meerwaarde. Het forceert architectuur na te denken over impact van een voorgenomen principe. Het brengt realiteit bij plannen. Principes waar een hoog kostenplaatje aan verbonden is moet je echt wel willen. Daarnaast scheidt het goede verwachten naar de projectmanagers die dit principe moeten toepassen en die zelf verantwoordelijk zijn voor aanvragen/bewaken budget. Daarnaast kan het worden meegenomen in plannings. Tevens geeft het input aan Return of Investment berekeningen. Wanneer hebben we de voordelen van dit principe eruit?</p>
Contextual artifacts	<p>Principe kan gelden binnen elke informatiesysteem, zowel klein als groot. In Abstract niveau kan principe worden toegepast op groter niveau in ketenintegratieprojecten. Boven processen komen dan ketenprocessen als extra modulaire laag.</p>

Evaluatie

Op basis van dit experiment zijn een aantal bevindingen gedaan. Deze dienen als input voor de diverse hoofdstukken, maar vooral om het stappenplan verder uit te breiden.

1. Onzekerheid over correct invullen attributen:
 - a. 3, 4, 7, 12, (15/16)
2. Sommige attributen lijken overlappend. Deze bevatten dan ook dezelfde teksten voor het grootste deel. Onduidelijk is wat verschil is tussen combinaties:
 - a. 15 en 16
 - b. 15, 18

Daarnaast veel overlap tussen 9 (conflicten) en 20 impact

3. Onduidelijk in welke diepgang er beschreven moet worden. Met nam bij omschrijving. Dient daar bijvoorbeeld een theoretisch verhaal te komen over service oriënted architectuur. Definities van handelingen, processtappen, definities en gebruik van statussen? Of moet dit worden verondersteld als voorkennis. Een uitgebreidere beschrijving is wel makkelijker te hanteren door degene die het uit moet voeren.
4. Je hebt middelen om een doel te bereiken, bijvoorbeeld gebruik statussen i.p.v. namen van processtappen. Echter dit kan worden gezien als een handhavingmechanisme (afdwingen door systeem) of als een verander doel (voortaan niet meer sturen op procesnaam) of als een implementatie impact onderdeel. Hierdoor krijg je veel herhaling. De vraag is hier nu: moeten dit niet aparte principes worden. Zo ja dan heb je een mechanisme nodig waarmee je principes onderling kan associëren/aggregeren. Principe 1 wordt dus gerealiseerd door principe 2 en 3 en principe 2 kent meerder verschijningsvormen namelijk optie 2a en optie 2b. Je hebt dan een principe diagram nodig met aggregatie/overerving etc. Elk proceselement heeft een nauwkeurige beschrijving met pre- en postcondities en service signatuur. Elk proceselement moet worden opgenomen in een catalogus.
5. De kosten schatting is zeer waardevol, zie commentaar bij attribuuttype.
6. Onduidelijk is nog de meerwaarde van de attributen: 5, 6, 7, 11. Waarvoor worden deze gebruikt?
7. Kwaliteitseisen gaan ons inziens wel erg diep. Je hebt altijd een vertaling van principe naar praktijk. Per vertaling heb je vanwege andere context andere kwaliteitseisen. Dus lastig om kwaliteitseisen "geldig voor alle situaties" vooraf te kunnen beschrijven.

8. Wel: betere uitwerking van architectuurprincipe. Ook de doelen die er mee worden bereikt en de rationale erachter. Geeft de uitvoerende meer inzicht en context. Echter in een regulier projectteam zal een bovenstaande uitwerking niet voldoende zijn of aanlanden want:
 - a. Bedrijfsdoelen vindt niet iedereen interessant. Niet elke programmeur zal zich interesseren of iets kunnen voorstellen van lange termijn bedrijfsfilosofie, of heeft affectie met commerciële overwegingen/processen.
 - b. Investerings is voor programmamanagement interessant maar niet voor veel andere rollen.
 - c. Kwaliteitsregels zijn weer abstract en vaak lastig toe te passen in de rol van individuele analist/programmeur.

Kortom de regels moeten worden doorvertaald naar ontwerp/bouw richtlijnen van concreter niveau. Beschrijving van de regels is dus ook cruciaal. Echter hiervoor is geen attribuut voorhanden? Anders kan je alsnog niet bewaken dat een regel wordt geïmplementeerd.

Conclusie

Op basis van het experiment, de bevindingen en de evaluatie zijn de volgende algemene aandachtspunten / inzichten geformuleerd:

- Een aantal attributen die beschreven dienen te worden zijn zonder achtergrond informatie en kennis op het gebied van EA lastig te begrijpen. Hier kan een duidelijke omschrijving van wat er verwacht wordt hulp bieden.
- Een bijkomstigheid van bovenstaand punt is dat een aantal attributen lijken te overlappen. De verschillen zijn niet in één oogopslag duidelijk waardoor het gevoeg gecreëerd wordt dat sommige informatie twee keer wordt gevraagd. Een duidelijkere volgorde voor het invulling geven aan de verschillende attributen zou hier bij kunnen helpen.
- De diepgang waarmee de attributen beschreven te worden is niet beschreven in het stappenplan. Dit is natuurlijk de keuze van degene die de architectuurprincipes opstelt, maar een suggestie / indicatie zou de onervaren architect helpen.
- Het beschrijven van het handhavingmechanisme blijkt lastig. Is het architectuurprincipe zelf een handhavingmechanisme? Dit zou het eigenlijk wel moeten zijn.
- Het beschrijven van de impact van een architectuurprincipe en de mogelijke kosten (een inschatting) kunnen zeer waardevol zijn bij het beslissen of een architectuurprincipe onderkend moet gaan worden door een organisatie.
- Van een aantal attributen is de meerwaarde niet duidelijk. Hieraan moet meer aandacht worden besteed bij het uitleggen van wat de verschillende attributen inhouden en waar deze voor dienen.
- De gedetailleerde beschrijving van het architectuurprincipe levert veel meer informatie en inzicht op in wat het architectuurprincipe nu inhoud dan het oorspronkelijke architectuurprincipe.
- Er wordt gesteld dat niet alle informatie even relevant is voor iedere doelgroep. Echter is wel alle informatie nodig om alle doelgroepen te voorzien van voldoende informatie omtrent het architectuurprincipe.
- De relatie en wijze waarop de architectuurprincipes kunnen worden doorvertaald naar regels etc. ontbreekt in het stappenplan.

Deze aandachtspunten zijn besproken met Mark Paauwe (auteur Dragon1). Op basis hiervan is een kort nieuwe stappenplan opgesteld met Mark Paauwe dat na overleg en aanpassingen gebruikt is om een drietal architectuurprincipes te herformuleren. Dit stappenplan is specifiek gericht op het herformuleren van architectuurprincipes. Het stappenplan is opgenomen in 'Dragon1 Methode Stappenplan - Architectuurprincipes opstellen, reviewen en herformuleren v0.15.2c'.

Dragon1 Stappenplan - Architectuurprincipes opstellen, reviewen en herformuleren v0.15.2c

Zoals besproken in het vorige hoofdstuk is er een (deel)stappenplan toegevoegd aan het document waarin het complete stappenplan is beschreven. Hieronder wordt deze toevoeging weergegeven:

Het normaliseren van architectuurprincipes

Er zijn vijf stappen voor het normaliseren van de formulering van een principe.

Hieronder staan deze stappen weergegeven.

Stap 1: Het selecteren van een uitspraak die bestaat uit een beschrijving met een titel

Selecteer een (deel van een) (samengestelde) uitspraak uit een lijst met uitspraken, waarvan men zegt dat het principes zijn of moeten worden. De geselecteerde uitspraak noemen we vanaf nu een RAW-statement, met een RAW-title en een RAW-name. Schrijft het RAW-statement op of type het in en voor het statement van een RAW-ID.

Stap 2: Het herkennen, uiteenrafelen of identificeren van het soort uitspraak

Hierbij is het van belang om te weten wat er nodig is. Wat wil men gaan doen met de statements? Waar wil men ze voor gaan gebruiken?

We gaan hier ervan uit dat de statements qua formulering echte principes, ontwikkelprincipes of nog beter architectuurprincipes moeten worden, en geen regels of uitgangspunten. We gaan er daarbij ook vanuit dat de geselecteerde RAW-statement een regel (veelal een ontwerpregel) kan zijn, of een uitgangspunt (veelal een strategisch uitgangspunt of een beleidsuitgangspunt), of een axioma of een echt principe (veelal een ontwikkelprincipe of een architectuurprincipe).

Criteria voor het herkennen van een uitspraak:

- Een goed geformuleerd principe beschrijft de gehandhaafde werking van iets (of een deel daarvan), zoals een concept, systeem of fenomeen. Een ontwikkelprincipe gaat over de wijze waarop iets (of een deel daarvan) kan worden ontwikkeld. Principes beginnen met het woord **DOOR** en beschrijven het gevolg of het resultaat van de werking van iets en wat daar dan mee mogelijk wordt gemaakt.
- Een goed geformuleerde regel beschrijft de afspraak (vaak een relatie tussen entiteiten) die door twee of meer partijen wordt gemaakt. Het beschrijft hoe iets moet worden gestructureerd of worden gedaan. Op het niet nakomen of naleven van de regel hoort een sanctie te staan. Regels hebben soms ook de vorm van een gedragsregel (norm) of het opleggen van een standaard. Een regel waar geen sanctie op staat is een richtlijn.
- Een goed geformuleerd uitgangspunt beschrijft vaak wat een organisatie(onderdeel) nastreeft. Bijvoorbeeld op identiteits- of klantgebied. Uitgangspunten beginnen met de woorden "We streven ernaar..."

Schrijf achter het RAW-statement op wat voor soort uitspraak het nu is en welke soort uitspraak het moet worden. Dit kunnen dezelfde soorten zijn, maar ook verschillende soorten. Stel dat het RAW-statement nu een slecht geformuleerde ontwerpregel is die een architectuurprincipe moet worden. Dan is het verstandig er eerst een goede ontwerpregel van te maken. Van een ontwerpregel kun je afleiden welk principe er achter zit.

Stap 3: Het achterhalen wat het huidige doel is van het statement en wat de bron is van het statement of waar het van is afgeleid.

Zoek de oorspronkelijke documenten of literatuur op die men heeft gebruikt om het RAW-statement op te schrijven of te bedenken. Schrijft het doel op van het statement en waar het in bij of op moet worden gebruikt.

Soms heb je dit allemaal nodig om het uitgangspunt, ontwerpregel, axioma of principe beter te kunnen formuleren. We stellen hier dat een principe eigenlijk altijd in literatuur terug te vinden moet zijn, omdat het onderzoek en testtijd vergt voordat een principe ook echt als onweerlegbaar mag worden gekwalificeerd.

Vaak komen er in uitspraken fenomenen, systemen en concepten voor waarover iets wordt gezegd. Soms is het verstandig om het statement helemaal opnieuw te formuleren en dan is het het eenvoudigste om het vanuit het concept, systeem of fenomeen te formuleren.

Ook kan het nodig zijn om naar abstractere, hogere klassen of meer conceptuele fenomenen, concepten en systemen te kijken. Hoe algemener de uitspraak of statement, des te groter is het gebied dat het afdekt. Het is de wens om hiernaar te streven. Een organisatie kan dan met minder statements uit de voeten en het zorgt ervoor om onnodige uitzonderingssituaties op te heffen, want die kosten vaak alleen maar onnodig veel tijd, geld, capaciteit, aandacht en resources.

Stap 4: Herformuleer het RAW-statement naar het gewenste soort uitspraak

Maak het RAW-statement SMART als het een regel of principe is.

Kijk of het statement een beschrijving op fysiek niveau is waarbij het wellicht beter is om het statement op logisch niveau te formuleren (afhankelijkheid naar product, technologie of leverancier weghalen) indien nodig of gewenst.

Als het statement een principe moet worden formuleer dan wat er (steeds) gebeurt, wat dat als gevolg of resultaat heeft en wat dat dan weer mogelijk of onmogelijk maakt.

Als het statement een architectuurprincipe dient te worden, probeer dan zoveel mogelijk context, scope of kader beperkingen weg te halen of weg te laten. Een architectuurprincipe dient holistisch te zijn.

Als het statement een uitgangspunt moet worden probeer dan in 1 of 2 zinnen te formuleren wat de organisatie dient na te streven. Probeer in geval van een strategische uitgangspunt een strategisch thema te identificeren en daarna het uitgangspunt op het juiste niveau te formuleren.

Stap 5: Voorzie het RAW-statement van een geformuleerde naam en titel die idealiter uit 1 tot 5 woorden bestaat.

Experiment

Het bovenstaande stappenplan is toegepast door middel van het herformuleren van drie architectuurprincipes van Stater. Het doel van dit experiment is geweest te kijken in hoeverre het toepasbaar is en welke inzichten / bevindingen over architectuurprincipes bevestigd kunnen worden door middel van deze praktijkoefening. Per architectuurprincipe is een uitwerking te vinden met commentaar en wijze waarop is geherformuleerd en welke keuzes hierbij zijn gemaakt. In de evaluatie wordt beschreven welke constatering m.b.t. architectuurprincipes zijn af te leiden van dit experiment.

Herformuleren architectuurprincipe: Flexibele tijdslijnen

Stap 1:

RAW-ID	1
RAW-title	Flexibele tijdslijnen
RAW-statement	De periodieke (administratieve) afsluiting kan per individuele financiële afspraak ingeregeld worden. Dagelijks vindt afsluiting van relevante afspraken plaats.

Stap 2:

De uitspraak bestaat uit twee delen namelijk:

1. De periodieke (administratieve) afsluiting kan per individuele financiële afspraak ingeregeld worden.
2. Dagelijks vindt afsluiting van relevante afspraken plaats.

Als gekeken wordt naar wat hier is opgeschreven kan het eerste deel van deze uitspraak als een regel geclassificeerd worden. Het beschrijft een afspraak (relatie) tussen de entiteiten periodieke afsluiting en individuele financiële afspraak. Het betreft hier een regel die optioneel is (het woord 'kan' wordt gebruikt). Het tweede deel van deze uitspraak is opgeschreven alsof dit altijd geldt. Dit is een uitspraak waaruit blijkt dat er elke dag een afsluiting plaats vindt van relevante afspraken. Deze uitspraak moet een architectuurprincipe worden.

Stap 3:

Het doel van deze uitspraak is echter dat de ICT-oplossing rekening moet houden met een requirement afkomstig uit de omgeving van het bedrijf. Deze requirement is gebaseerd op een concept uit de omgeving/markt waarin het bedrijf opereert. Deze constatering is o.a. gebaseerd op het volgende citaat van Wim van Stokkum uit het document Classificatie Architectuurprincipes Stater NV:

"Het is een requirement vanuit de omgeving van het bedrijf. Het informatiesysteem moet volgen."

Daarnaast is er een functionele eis in deze uitspraak verwerkt volgens de onderstaande tekst afkomstig uit de Stater Architectuurbeschrijving:

"Aan de ene kant is dit een functionele eis. Daarom komt dit principe terug in de uitwerking van het Stater domeinmodel. Aan de andere kant stelt dit principe ook eisen aan de performance van het systeem. De uitwerking van dit principe mag geen negatieve invloed hebben op de beschikbaarheid van het systeem."

Uit deze achtergrond informatie blijkt dat het eerste deel van de uitspraak juist een concept bevat wat afkomstig is uit de omgeving van het bedrijf en te definiëren is als:

De periodieke administratieve afsluiting wordt per individuele financiële afspraak ingeregeld.
--

Dit is een uitspraak die als waar beschouwd kan worden als het concept daadwerkelijk te vinden is in de omgeving van het bedrijf. Het is schijnbaar zo dat in de hypothecaire wereld de periodieke afsluiting per individuele financiële afspraak ingeregeld zou moeten kunnen worden.

Achterliggende redenen/motivatie van dit concept zijn o.a.:

Consumenten willen graag :

- *Rente en aflossing soms van verschillende rekeningen af laten boeken*
- *Op door hun gewenste tijden (periodiciteit en afboekdag) laten afboeken*
- *Willen incasso's verdelen over meerdere schuldenaren (deel ouders, deel zelf, deel bedrijf)*

De volgende opmerking is echter van toepassing op dit concept:

“Grappig is dat in praktijk deze wens slechts sporadisch voorkomt. In minder dan 5% van de af te lossen hypotheek wordt daadwerkelijk door de consument deze flexibiliteit gewenst. Een vermoeden van WVS is dat dit wel handig kan zijn als je i.p.v. consumptieve hypothecaire leningen ook leningen wil gaan verstrekken aan bedrijven. Dan heb je ingewikkeldere betaal afspraken nodig. Echter dit is een lange termijn optie voor Stater en is nog niet binnen 5 jaar daadwerkelijk relevant.”

Uiteindelijk kan geconcludeerd worden dat in werkelijkheid deze vorm van flexibiliteit nauwelijks gewenst is en dat de onderbouwing van deze uitspraak niet voldoende is en daardoor deze uitspraak niet onderkend had mogen worden.

Het tweede deel van de uitspraak blijkt een eis/requirement aan de ICT-oplossing te zijn. Dit is dus een regel afgeleid van het eerste deel van deze uitspraak. Men wil graag dat dagelijks een afsluiting van relevante afspraken plaats vind aangezien dit één van de vereisten is om het principe te kunnen handhaven binnen de ICT-oplossing. Deze regel is te definiëren als:

Dagelijks vindt administratieve afsluiting van relevante financiële afspraken plaats.

Stap 4:

Op basis van de analyse bij stap 3 is het lastig om een architectuurprincipe te formuleren. Je zou kunnen zeggen dat dit principe in deze vorm niet geformuleerd had mogen worden. Op basis van de volgende uitspraken is geprobeerd het daadwerkelijke concept/principe te herleiden:

“Het echte principe is eigenlijk: flexibiliteit in operatie op alle mogelijke aspecten waarop consumenten/banken zouden kunnen gaan variëren.”

“De periodieke administratieve afsluiting wordt per individuele financiële afspraak ingeregeld.”

“Grappig is dat in praktijk deze wens slechts sporadisch voorkomt. In minder dan 5% van de af te lossen hypotheek wordt daadwerkelijk door de consument deze flexibiliteit gewenst. Een vermoeden van WVS is dat dit wel handig kan zijn als je i.p.v. consumptieve hypothecaire leningen ook leningen wil gaan verstrekken aan bedrijven. Dan heb je ingewikkeldere betaal afspraken nodig. Echter dit is een lange termijn optie voor Stater en is nog niet binnen 5 jaar daadwerkelijk relevant.”

Deze informatie is op de volgende wijze in te passen in de door Dragon1 voorgestelde vorm van het formuleren van een principe:

Door [flexibiliteit in operatie op alle mogelijke aspecten waarop consumenten/banken zouden kunnen gaan variëren] wordt het mogelijk [de periodieke administratieve afsluiting wordt per individuele financiële afspraak ingeregeld] en daardoor [Rente en aflossing soms van verschillende rekeningen af laten boeken] en [Op door hun gewenste tijden (periodiciteit en afboekdag) laten afboeken] en [Willen incasso's verdelen over meerdere schuldenaren (deel ouders, deel zelf, deel bedrijf)]

Het uiteindelijke architectuurprincipe:

Door met alle mogelijke aspecten waarop consumenten/banken zouden kunnen gaan variëren rekening te houden bij flexibiliteit in operatie is het mogelijk de periodieke administratieve afsluiting per individuele

financiële afspraak in te regelen en daardoor wordt het mogelijk rente en aflossing van verschillende rekeningen af te laten boeken, op de door de consument/bank gewenste tijden (periodiciteit en afboekdag) laten afboeken en incasso's te verdelen over meerdere schuldenaren.

Stap 5:

Bij het principe is de titel Flexibele tijdslijnen nog steeds van toepassing en dekt de lading van het principe.

Commentaar Wim van Stokkum

Deze uitspraak is voortgekomen uit een huidig probleem van het oude systeem van Stater. Stater kan slechts voor alle leningen alle incassosoorten, rente en aflossing als een geheel incasseren bij slechts één persoon via slechts één rekeningnummer op maandelijkse basis.

Consumenten willen graag :

- Rente en aflossing soms van verschillende rekeningen af laten boeken
- Op door hun gewenste tijden (periodiciteit en afboekdag) laten afboeken
- Willen incasso's verdelen over meerdere schuldenaren (deel ouders, deel zelf, deel bedrijf)

Eigenlijk is dit gewoon een systeemrequirement, waar het label architectuurprincipe is opgeplakt. Het echte principe is eigenlijk: flexibiliteit in operatie op alle mogelijke aspecten waarop consumenten/banken zouden kunnen gaan variëren.

Grappig is dat in praktijk deze wens slechts sporadisch voorkomt. In minder dan 5% van de af te lossen hypotheekleningen wordt daadwerkelijk door de consument deze flexibiliteit gewenst. Een vermoeden van WVS is dat dit wel handig kan zijn als je i.p.v. consumptieve hypotheekleningen ook leningen wil gaan verstrekken aan bedrijven. Dan heb je ingewikkeldere betaalafspraken nodig. Echter dit is een lange termijn optie voor Stater en is nog niet binnen 5 jaar daadwerkelijk relevant.

Om duidelijk te maken wat ze willen bij architectuur, is hier in maanden tijd een prototype voor ontwikkeld.

Evaluatie

Problematiek omtrent principe:

- Uitspraak is geen principe maar een requirement/functionele eis.
- Uitspraak is gebaseerd op een concept wat in werkelijkheid niet bestaat of in ieder geval een andere vorm heeft.
- Uitspraak is pas van toepassing over een x aantal jaar of mogelijk nooit van toepassing.
- Uitspraak is geformuleerd alsof het een waarheid betreft.
- Prototype gebouwd om te laten zien wat er bedoeld wordt met deze uitspraak.

Had deze problematiek voorkomen kunnen:

Wanneer de Dragon1 methode gebruikt zou zijn bij het opstellen en formuleren van de architectuurprincipes had deze problematiek voorkomen kunnen worden:

- Het concept waarop deze uitspraak gebaseerd is blijkt niet waar te zijn. De Dragon1 methode raad aan alleen principes of concepten te hanteren die waar zijn.

- Indien de beschouwingssituatie van deze uitspraak beschreven zou zijn had men tot de conclusie kunnen komen dat deze uitspraak pas over een x aantal jaar van toepassing zou kunnen zijn geweest.
- Indien men de implementatie impact van deze uitspraak had beschreven zou men tot de conclusie gekomen kunnen zijn dat de kosten niet opwegen tegen de baten.

Herformuleren architectuurprincipe: Straight Through Processing

Stap 1:

RAW-ID	1
RAW-title	Straight Through Processing
RAW-statement	De verwerking van aangeboden verzoeken (zoals aanvragen) dient door de ICT-oplossing automatisch uitgevoerd te kunnen worden.

Stap 2:

Het betreft hier een slecht geformuleerde regel. Het beschrijft hoe verzoeken verwerkt moeten worden. Deze regel is afgeleid van het algemene concept/principe automatiseren (automatisering). Deze uitspraak moet een architectuurprincipe worden.

Stap 3:

Deze uitspraak mist te veel informatie om er een goed architectuurprincipe van te maken. De volgende informatie ontbreekt:

- Motivatie en/of achterliggende reden en/of doel.
- Wat er mogelijk of onmogelijk wordt hierdoor.

Deze informatie moet eerst worden verzameld om het juiste architectuurprincipe te kunnen formuleren. Uit het bestuderen van achtergrondinformatie en gesprekken met de contactpersonen van Stater is de onderstaande informatie naar voren gekomen.

Het doel van dit 'Statement' is:

- Stater wil de volledige automatische verwerking van businessprocessen optimaliseren.
- Kostenefficiëntie.

Dit wil men bereiken door zoveel mogelijk verzoeken automatisch te verwerken zonder dat hier een trigger (handeling) van een mens voor nodig is. De volgende uitspraken hebben betrekking op deze uitspraak (afkomstig uit Classificatie Architectuurprincipes Stater NV en Stater Architectuurbeschrijving):

"Waar het mogelijk is daar toepassen"

"Verzoeken/activiteiten die automatisch afgehandeld kunnen worden zouden ook automatisch afgehandeld moeten worden. Deze activiteiten kunnen afgehandeld worden zonder dat hier menselijke handelingen aan te pas komen. Dit scheelt tijd en geld."

Stap 4:

Nu we de ontbrekende informatie hebben en weten wat voor uitspraak het moet worden kan begonnen worden met het herformuleren. De uitspraak is te herformuleren naar het volgende architectuurprincipe:

Door de verwerking van aangeboden verzoeken te automatiseren, kunnen verzoeken zonder tussenkomst (trigger) van personeel en op elk moment verwerkt worden, zodat verzoeken sneller en efficiënter verwerkt kunnen worden en de kosten per verzoek afnemen.

Deze eerste uitwerking is echter nog niet scherp genoeg. De beschrijving van het architectuurprincipe bevat het woord 'kunnen' op meerdere plaatsen. Het woord 'kunnen' moet echter vermeden worden aangezien dit aangeeft dat het niet altijd zo hoeft te zijn. Ook beschrijft deze uitwerking nog niet de kern van het principe Straight Through Processing. De beschrijving moet dus nog verder worden aangepast.

Het uiteindelijke architectuurprincipe:

Door aangeboden verzoeken automatisch te verwerken zonder tussenkomst van een handeling van een mens, worden verzoeken op elk moment, sneller en efficiënter verwerkt en nemen de kosten per verzoek af.

("elk moment" moet hier gelezen worden als 24 uur per dag, 7 dagen in de week, het gehele jaar door.)

Stap 5:

De titel blijft hetzelfde: Straight Through Processing

Evaluatie

Als je puur kijkt naar wat hier is opgeschreven zegt de oorspronkelijke uitspraak eigenlijk vrij weinig. Het beschrijft half het concept automatiseren. Het uiteindelijke architectuurprincipe is eigenlijk een (mogelijke) beschrijving van het concept automatiseren en is dus nog steeds erg algemeen. Door het beschrijven van het resultaat, de motivatie en iets te zeggen over hoe het resultaat wordt bereikt wordt in ieder geval duidelijk waarom dit architectuurprincipe onderkend is door Stater. De volgende conclusies kunnen worden getrokken uit het herformuleren van dit architectuurprincipe:

- Het concept automatiseren is beschreven in andere woorden. Oftewel bestaande theorie is eigenlijk opnieuw opgeschreven. Dit architectuurprincipe wordt in ieder informatiesysteem toegepast.
- Het onderkennen van dit architectuurprincipe in deze vorm heeft eigenlijk weinig meerwaarde. Men had hier specifiekere moeten zijn en vooral aandacht moeten geven aan hoe het resultaat dat beschreven is in het architectuurprincipe bereikt gaat worden en wat dit mogelijk maakt.

Herformuleren architectuurprincipe: Billing

Stap 1:

RAW-ID	2
RAW-title	Billing
RAW-statement	De ICT-oplossing maakt het mogelijk om op basis van een verzameling gebruiksgegevens verschillende vormen van tarifiering aan te bieden.

Stap 2:

Het RAW-statement is een regel. Deze regel moet een architectuurprincipe worden.

Stap 3:

Het achterliggende doel van deze uitspraak is dat men door flexibiliteit in de ICT-oplossing meer transparantie mogelijk wil maken. De volgende informatie is van toepassing op deze uitspraak:

In het huidige systeem dat Stater gebruikt is de tarifiering niet in verhouding met de mate van inspanning van de activiteit. Dit principe is opgesteld omdat het belangrijk is om een tarifiering te kunnen bieden die aansluit bij de geleverde inspanning. Dit principe is opgesteld vanuit het strategische doel; Stater wil klanten flexibele oplossingen bieden met een kostenefficiënte dienstverlening.

[bron: classificatie_architectuurprincipes_Stater_NV_v1.doc]

Stap 4:

De eerste stap voor het herformuleren is het SMART formuleren van het RAW-statement. Hieronder worden beschreven hoe dit gedaan kan worden:

Aangezien de uitspraak maar op één informatiesysteem (ICT-oplossing) van toepassing is kan hier de naam van dit informatiesysteem genoemd worden. Daarnaast kan aangegeven worden om welke verzameling gebruiksgegevens het hier gaat en uit welke gegevens deze verzameling bestaat. Ook kan worden aangegeven op welke wijze de tarifiering dan bepaald dient te worden. Door dit te beschrijven wordt de uitspraak specifiek en beter meetbaar. De uitspraak is acceptabel. Het is afgeleid van een hoger strategisch doel waar het bedrijf achter staat. De uitspraak is realistisch en tijdsgebonden door het feit dat het men de uitspraak waar wil maken binnen de ontwikkeltijd die staat voor het informatiesysteem.

Door het SMART formuleren van de uitspraak wordt een strakker geformuleerde regel verkregen. De uitspraak dient echter een architectuurprincipe te formuleren. In de huidige formulering wordt alleen aandacht besteed aan wat er gebeurt. Wat het gevolg en resultaat is, en wat het mogelijk en/of onmogelijk maakt is niet beschreven. Door hier invulling aan te geven kan het volgende architectuurprincipe geformuleerd worden:

Door [op basis van een verzameling gebruiksgegevens verschillende vormen van tarifiering te bieden] [sluit de tarifiering aan bij de inspanning van de geleverde diensten en wordt transparantie gecreëerd in de wijze waarop de tarifiering is bepaald]. [Hierdoor wordt het voor de klant inzichtelijk op welke wijze de tarifiering wordt bepaald, voor welke diensten men betaald en hoeveel men betaald voor deze diensten].

Blokhaak rood: wat gebeurt er

Blokhaak blauw: wat is het gevolg en resultaat

Blokhaak groen: wat maakt het mogelijk en onmogelijk

Deze formulering lijkt al een stuk meer op een architectuurprincipe dan de oorspronkelijke uitspraak. De nieuwe wijze van formuleren is o.a. context onafhankelijk. In deze formulering zijn echter nog een aantal zwakke punten waardoor men niet kan stellen dat dit architectuurprincipe waar is. Het beschreven gevolg en

resultaat in dit principe is niet direct af te leiden van wat er gebeurt. De vraag hoe en of er transparantie wordt gecreëerd is niet te beantwoorden (verantwoorden). Om deze vragen te beantwoorden moet het duidelijk zijn wie transparantie wenst en op welke wijze zij dit verkrijgen. In dit geval is dit de klant. De wijze waarop transparantie verkregen wordt is door de wijze van tarifiering inzichtelijk te maken en te communiceren naar de klant. Ook de vraag waarom de tarifiering beter aan zou sluiten bij de inspanning van de geleverde diensten is niet te beantwoorden. Hier kan wel naar gegist worden maar dit biedt geen garantie. Het moet duidelijk worden beschreven welke gebruiksgegevens hiervoor gebruikt worden en waar deze aan gekoppeld worden om de tarifiering te bepalen. In dit geval is gekozen voor frequentie en mate van inspanning (er zullen nog meer gebruiksgegevens een rol spelen maar dit geeft een goede indicatie). In de volgende uitwerking zijn deze zwakke punten afgevangen:

Door [op basis van de frequentie en de mate van inspanning van een geleverde dienst verschillende vormen van tarifiering te bieden] [wordt het inzichtelijk op welke wijze de tarifiering bepaald wordt en sluit de tarifiering beter aan bij de inspanning van de geleverde diensten met als resultaat meer transparantie op het gebied van tarifiering]. [Hierdoor wordt het mogelijk om de wijze waarop de tarifiering wordt bepaald communiceerbaar te maken naar de klant en heeft de klant een duidelijk beeld voor welke diensten men betaald, hoeveel men betaald voor deze diensten en waarom men deze hoeveelheid betaald].

Het bovenstaande statement is een architectuurprincipe. Dit architectuurprincipe beschrijft een deel van het concept tarief (of tarifiering).

Stap 5:

De titel van dit architectuurprincipe moet zijn:

Flexible billing

Evaluatie

De volgende bevindingen zijn gedaan:

- Het oorspronkelijke architectuurprincipe is zeer onvolledig geformuleerd. Maar een deel van het resultaat en de wijze waarop het resultaat bereikt dient te worden is beschreven. Dit leidt tot een uitspraak die weinig inhoud heeft. Er is geen invulling gegeven aan de context en daardoor veel te algemeen.
- Het gebruiken van de redeneerstrategie die bij propositiologica gehanteerd wordt is in te zetten om een architectuurprincipe scherper te definiëren. Door te kijken of het resultaat bereikt kan worden door middel van de beschreven aanpak kan geverifieerd worden of dit logisch is of aangescherpt dient te worden.
- De begrippen; resultaat, gevolg, mogelijkheden en onmogelijkheden, hebben te veel overlap en kunnen synoniem zijn voor elkaar.

Conclusie

Op basis van de reeks experimenten zijn een aantal inzichten / problemen m.b.t. architectuurprincipes en het formuleren hiervan bevestigd en geconstateerd. Deze zullen in onderstaand overzicht nog puntsgewijs benoemd worden:

Over architectuurprincipes

- Uitspraken die in de praktijk architectuurprincipes genoemd worden zijn vaak regels, eisen en wensen of deelbeschrijvingen van architectuurprincipes.
- Architectuurprincipes zijn vaak zeer onvolledig geformuleerd. Maar een deel van het resultaat en de wijze waarop het resultaat bereikt dient te worden is beschreven.
- Dient een architectuurprincipe zelf een handhavingmechanisme te zijn of moet er een handhavingmechanisme afgedwongen worden voor een architectuurprincipe?
- Veel architectuurprincipes worden in de praktijk niet vertaald naar de toepassing ervan in de context. Hierdoor wordt eigenlijk (bestaande) theorie opgeschreven, zonder een praktijkvertaling te maken. Dit leidt tot een uitspraak die weinig inhoud heeft. Er is geen invulling gegeven aan de context en daardoor veel te algemeen. De architect laat beslissingen over aan anderen en laat dus anderen voor architect spelen en neemt te weinig verantwoordelijkheid.
- De attributen / eigenschappen die worden toegekend aan architectuurprincipes lijken in sommige opzichten te overlappen. Denk hierbij bijvoorbeeld aan het verschil tussen impact en voor-/nadelen. Een voor- of nadeel kan ook gezien worden als impact.
- Architectuurprincipes worden opgeschreven alsof het een waarheid betreft terwijl dit niet onderbouwd kan worden.
- Een architectuurprincipe heeft de vorm van een gevolgtrekking.
- De beschrijvingen van architectuurprincipes bieden onvoldoende duidelijkheid, een alternatieve methode om het architectuurprincipe te verduidelijken blijkt nodig.
- De tijdspanne waarin een architectuurprincipe van kracht is (gaat worden) is belangrijk te onderkennen.
- De begrippen; resultaat, gevolg, mogelijkheden en onmogelijkheden, hebben te veel overlap en kunnen synoniem zijn voor elkaar.

Over architectuurprincipes formuleren (op basis van Dragon1)

- Een aantal attributen die beschreven dienen te worden zijn zonder achtergrond informatie en kennis op het gebied van EA lastig te begrijpen. Hier kan een duidelijke omschrijving van wat er verwacht wordt hulp bieden.
- Een bijkomstigheid van bovenstaand punt is dat een aantal attributen lijken te overlappen. De verschillen zijn niet in één oogopslag duidelijk waardoor het gevoel ontstaat dat sommige informatie twee keer wordt gevraagd. Een duidelijkere volgorde voor het invulling geven aan de verschillende attributen zou hier bij kunnen helpen.
- De diepgang waarmee de attributen beschreven te worden is niet beschreven in het stappenplan. Dit is natuurlijk de keuze van degene die de architectuurprincipes opstelt, maar een suggestie / indicatie zou de onervaren architect helpen.
- Het gebruiken van de redeneerstrategie die bij propositiologica gehanteerd wordt is in te zetten om een architectuurprincipe scherper te definiëren. Door te kijken of het resultaat bereikt kan worden door middel van de beschreven aanpak kan geverifieerd worden of dit logisch is of aangescherpt dient te worden.
- Het beschrijven van de impact van een architectuurprincipe en de mogelijke kosten (een inschatting) kunnen zeer waardevol zijn bij het beslissen of een architectuurprincipe onderkend moet gaan worden door een organisatie.
- Van een aantal attributen die worden toegekend aan architectuurprincipes is de meerwaarde niet duidelijk. Hieraan moet meer aandacht worden besteed bij het uitleggen van wat de verschillende attributen inhouden en waar deze voor dienen.
- De gedetailleerde beschrijving van het architectuurprincipe levert veel meer informatie en inzicht op in wat het architectuurprincipe nu inhoud dan het oorspronkelijke architectuurprincipe.

- Er wordt gesteld dat niet alle informatie even relevant is voor iedere doelgroep. Echter is wel alle informatie nodig om alle doelgroepen te voorzien van voldoende informatie omtrent het architectuurprincipe.
- De relatie en wijze waarop de architectuurprincipes kunnen worden doorvertaald naar regels etc. ontbreekt in het stappenplan.

Bijlage D: Attributen van een architectuurprincipe Dragon1

Dragon1 kent o.a. de volgende attributen toe aan een architectuurprincipe. Deze lijst is afkomstig uit het Dragon1 Stappenplan - Architectuurprincipes opstellen, reviewen en herformuleren v0.15.2c.

Attribuut	Uitleg
1. Een unieke universele ID	Vanwege de (miljoenen) principes in een enterprise architectuur dient een principe een unieke identificatie te hebben.
2. Een unieke universele Naam	Een kort statement van één tot tien woorden die duidelijk maken wat de context en werking van het principe is. Bijvoorbeeld: het zwaartekracht-principe of het gedigitaliseerde werkprocessenprincipe.
3. Systeem, Concept of Stijl verwijzing	<p>Een principe is altijd onderdeel van een systeem, concept, fenomeen of stijl.</p> <p>Een systeem is een geheel van onderdelen die samenwerken teneinde een gemeenschappelijk doel te realiseren.</p> <p>Een concept is een gestructureerde, planmatige of systematische werkwijze.</p> <p>Een fenomeen is een systeem dat in de natuur voor komt en dat ongekende krachten in zich heeft en zeer tot de verbeelding spreekt.</p> <p>Een stijl is een verzameling van concepten met gemeenschappelijke (stijl)elementen. We maken onderscheid in systemen, concepten en stijlen voor constructie, ontwikkeling en verandering. Voorbeelden van gangbare stijlen zijn service oriëntatie, component oriëntatie, proces oriëntatie, project oriëntatie, data oriëntatie, event oriëntatie, engine oriëntatie en taakgericht werken. Per stijl is het hoofdstijlelement benoemd.</p>
4. Statement (Omschrijving of literatuur verwijzing)	<p>De formulering van het principe statement (de omschrijving van een principe) komt nogal nauw. De context waarbinnen het principe geldt en de periode en het abstractieniveau waarin het principe geldig is, dient duidelijk te worden afgebakend.</p> <p>Omdat een principe een waarheid betreft, zijn woorden zoals <i>mogen</i>, <i>willen</i>, <i>kunnen</i>, <i>zouden</i> en <i>moeten</i> niet gewenst in de naam of omschrijving. De formulering van een principe dient niet stellend te zijn, maar een werking te beschrijven. De formulering dient ultimo een oorzaakgevolg relatie te beschrijven maar dient minimaal een volledige entiteitenrelatie te beschrijven.</p> <p>De omschrijving van een principe gaat in op het mechanisme of de werking van een concept, systeem, fenomeen of stijl.</p> <p>De omschrijving van een principe moet zoveel mogelijk S.M.A.R.T. zijn.</p>
5. Klasse	<p>Is het een architectuur, constructie of structuurgericht principe?</p> <p>Is het een kringgedrag principe, taak/volgorde principe of actie-reactie principe?</p>
6. Type (Activiteit)	Is het een planningsprincipe, ontwikkel(bouw/ontwerp)principe, transformatie/veranderprincipe of beheer/instandhoudingprincipe?
x. abstractieniveau	Is het een conceptueel, conceptueel, logisch of fysiek principe?
x. Soort bedrijfssysteem-functie	Is het een inkoop, verkoop, marketingcommunicatie, productie, financiën, ICT (IV, ICT-Infra)
x. Soort	Is bijvoorbeeld een consumentenmarkt principe, een back office principes, een

organisatiedomein	winkelformuleprincipe of een assortiments-principe
7. organisatiesysteem-beschouwingsniveau	Is het een ondernemingsprincipe, bedrijfsprincipe, bedrijfsfunctieprincipe, bedrijfsproces, bedrijfstaak, handeling.
8. Eigenaar en beheerder	Wie de eigenaar en wie is de beheerder van het principe?
9. Issues en Conflicten	Wat zijn haken en ogen aan het principe om het in de organisatie te hanteren? Waar conflicteert dit principe met andere principes, of waar werken regels niet vanwege de verkeerde huidige principes?
10. Beschouwings-situatie	We maken onderscheid in situatietypen principes: AS-IS principles, Plateau n - principles, TO-BE principles en Envision principles. <ul style="list-style-type: none"> • Een AS-IS principe is een principe dat zich in de huidige situatie (heden - 1 jaar) daadwerkelijk voordoet, waarmee men er verstandig aan doet er rekening mee te houden. • Een plateau n principe is een principe dat zich tijdelijk voordoet (jaar X). • Een TO-BE principe is een principe dat zich in de toekomstige situatie (3-5 jaar) zal gaan voordoen, realistisch en haalbaar is, waarmee men er verstandig aan doet er in de toekomst rekening mee te houden. • Een Envision principe is een principe dat zich in de verre toekomst zal gaan voordoen, maar voor de AS-IS, Plateau1 en TO-BE situatie onrealistisch is.
11. Effect-type	We maken onderscheid in twee effect-type principes: true-principles en false-principles. Een true-principle is een principe waarvan het handhavingmechanisme effectief is, en zorgt voor de handhaving van het principe. Een false-principle is een principe waarvan het handhavingmechanisme faalt, en daarmee het principe niet wordt gehandhaafd.
12. First principle [Ja/Nee]	Ieder systeem, stijl, fenomeen of concept heeft een kern (van waarheid, of waar het om draait) die door middel van een principe kan worden geformuleerd. Dit principe is het first principle. Van alle systemen, concepten en stijlen die in een architectuur worden gebruikt dient het first principle te worden geformuleerd.
13. Status	We maken onderscheid in de volgende statusovergangen met bijbehorende status van een principe: voorstellen, kandideren, in overweging nemen, onderkennen.
14. Bron(-vermelding)	Voorbeelden van bronnen van principes zijn: de natuur, ervaring, een door de industrie ontwikkelde technologie, een best practice, een bewezen concept. De bronvermelding voor een principe is nooit een wens, eis, requirement of doelstelling. Het eisen van een bepaalde kwaliteit of een veranderdoel zijn wel aanleidingen om een bepaald concept (of principes) te willen of te moeten implementeren.
15. Rationale	De achterliggende reden of onderbouwing (bijvoorbeeld een hogere orde principe) voor het principe
16. Exploitatie en veranderdoelen	Welke doelen liggen ten grondslag aan de keuze voor het (concept van het) principe?
17. (Kwaliteit)eisen	De (kwaliteit)eisen (zie ISO 9126) die worden gesteld door belanghebbenden aan een systeem of oplossing vormen de reden dat met een principe rekening gehouden moet

	<p>worden of dat een principe geïmplementeerd dient te worden.</p> <p>Elk kwaliteitsaspect kan worden gezien als een hogere orde principe van een concept.</p>
18. Bedrijfsvoordelen /nadelen	Het voordeel of nadeel van een principe maakt dat men kan onderbouwen waarom dit principe nodig is of bijdraagt aan het realiseren van de gestelde kwaliteitseisen.
19. Handhaving-mechanisme	Dit is nodig om het principe binnen een bepaalde context altijd te kunnen af te dwingen. Voorbeelden van handhavingmechanismen voor principes zijn: toezicht, controle, sociale druk, beloning, bestraffing en dwingende structuren.
20. Implementatie impact	Wat dient er te worden aangepast om een principe waar te laten zijn in een context. Hoeveel tijd, geld en middelen kost het principe om (qua handhaving en structuurwijziging) te implementeren?
21. Contextual artifacts	Voorbeelden van onderdelen in de contexten waar een principe voor kan gelden zijn: systeem, systeemonderdeel, organisatie, onderneming, instelling, organisatiedomein, project, proces, product