

Bruikbaarheid van ER-modelmetriekeken

Peter Tenbült

15 november 2008



Colofon

Auteur: Peter Tenbült
Studentnummer: 0652962
Opleiding: Informatiekunde
Afstudeernummer: 85 IK
Specialisatie: Model Based System Engineering
Opdracht: "Bruikbaarheid van ER-modelmetrieken"

Universiteit: Radboud Universiteit Nijmegen (RU)
Comeniuslaan 4
6525 HP Nijmegen

Postbus 9102
6500 HC Nijmegen

Telefoon: 024-3616161
Fax: 024-3564606
Internet: www.ru.nl

Faculteit: Faculteit Natuurwetenschappen, Wiskunde & Informatica (FNWI)
Instituut: Nijmeegs Instituut voor Informatica en Informatiekunde (NIII)

Afstudeerbegeleider: Dr. Patrick van Bommel
Tweede lezer: Dr. Stijn Hoppenbrouwers

Plaats, datum: Nijmegen, zaterdag 15 november 2008
Versie, status: 1.01, Definitief

Voorwoord

Met trots presenteer ik u mijn afstudeerscriptie. Formeel ben ik 8 september 2008 begonnen, echter vanaf april 2008 heb ik gewerkt aan de voorbereidingen voor dit onderzoek. Om na mijn vakantie in augustus 2008 al te beginnen met de literatuurstudie. Ik heb ervoor gekozen om een intern onderzoek te doen. Een voorwaarde voor het afstuderen is dat een scriptie beschikt over voldoende wetenschappelijk niveau. Met deze scriptie heb ik naar mijn smaak de wetenschap bijgedragen op het gebied van conceptuele modellen en modelmetriek.

Graag wil ik een aantal mensen bedanken voor de bijdrage die zij hebben geleverd aan het totstand komen van deze scriptie. Allereerst gaat mijn dank uit naar mijn afstudeerbegeleider Patrick van Bommel voor de constructieve en prettige gesprekken die we gevoerd hebben tijdens het verloop van het onderzoek. Ook voor zijn goede begeleiding en feedback. Een ding vraag ik me toch nog wel af, of ik hem na elke tweewekelijkse bezoek nog niet ben gaan vervelen.

Ook wil ik Marc Pieterse bedanken, die een grote rol heeft gespeeld in de totstandkoming van de conceptuele modellen die in deze scriptie aanbod zullen komen. Ik wil hem niet alleen bedanken voor het meedenken en ontwerpen van de modellen, maar ook voor de tips en feedback die hij tijdens het schrijven van deze scriptie heeft gegeven.

Ik wil ook Carolien Gerards van het Academisch schrijfcentrum bedanken voor het beantwoorden van mijn vragen en voor je geduld, aangezien mijn Nederlands niet mijn sterkste punt is. Tijdens het schrijven van mijn scriptie leek het alsof ik aan jou tweede begeleider had. In ieder geval hartstikke bedankt voor je tijd en moeite.

Verder wil ik mijn familie, vrienden en vriendin bedanken voor hun geduld en steun. Ik weet dat ik de laatste tijd veel bezig ben geweest met het schrijven van deze scriptie, maar jullie begrip heeft me enorm gesteund.

Veel plezier gewenst bij het lezen van deze scriptie.

Peter Tenbült

Nijmegen, Zaterdag 15 november 2008

Abstract

Het gebruiken van conceptuele modellen is in de IT-wereld niet meer weg te denken. Voor het programmeren, ontwerpen van databases, in kaart brengen van bedrijfsprocessen, overal in de IT wordt er gebruik gemaakt van conceptuele modellen. Maar hoe kom je te weten of deze conceptuele modellen nu van goede kwaliteit zijn?

In dit onderzoek wordt er gekeken naar de modelmetrieken van drie verschillende onderzoekers. Met deze modelmetrieken zou het mogelijk zijn om kwaliteitseigenschappen tijdens het ontwikkelen van conceptuele modellen te kunnen evalueren. Om uiteindelijk de kwaliteit van de conceptuele modellen te verbeteren. Er zijn verschillende onderzoekers bezig geweest met het ontwikkelen van modelmetrieken, maar dit onderzoek zal zich beperken tot alleen modelmetrieken ontwikkeld voor ER-modellen. Waarbij de hoofdvraag van dit onderzoek als volgt luid:

Kunnen de modelmetrieken die zijn ontwikkeld voor ER-modellen worden toegepast op UML-modellen?

Het onderzoek zal een korte beschrijving geven van de modelmetrieken die door Moody, Maes en Poels zijn ontwikkeld. Om met deze basis de modelmetrieken te kunnen toepassen op twee ER-modellen, twee procesmodellen en één UML-model. Op basis van de resultaten die deze modelmetrieken opleveren kunnen er twee dingen worden geconcludeerd.

Allereerst de informele modelmetrieken kunnen makkelijker worden toegepast op een UML-model in verhouding tot de formele modelmetrieken. Dit komt omdat de formele modelmetrieken gebaseerd zijn op ER conventies. UML-modellen hebben deze conventies niet.

Ten tweede kan er na de toepassing van deze modelmetrieken worden geconcludeerd dat er geen oplossing wordt aangedragen om de kwaliteit van een conceptueelmodel te verbeteren. Het komt erop dus op neer dat men een modelmetriek kan uitvoeren en wat de uitkomst ook is (goed of slecht), qua verbetering zal er niks worden geopperd.

Kernwoorden:

Conceptual modelling, Metrics, Model Metrics, ER modelling, UML modelling

Inhoudsopgave

1 Inleiding	7
1.1 Aanleiding	7
1.2 Leeswijzer	10
 I Introductie	 12
2 Probleemstelling	13
2.1 Onderzoeksvraag	14
3 Verantwoording	15
3.1 Waarom dit onderzoek?	15
3.2 Theoretische relevantie	15
3.3 Praktische relevantie	16
3.4 Maatschappelijke relevantie	16
3.5 Doelgroep	17
4 Theoretisch Kader	18
4.1 Kennisgebied	18
4.1.1 Softwaremetriekeken versus Modelmetriekeken	18
4.1.2 Formele versus Informele modelmetriekeken	19
4.2 Termen	19
4.3 Keuzes en Veronderstellingen	21
4.4 Onderzoek tot nu toe	21
5 Onderzoeksstructuur	24
5.1 Onderzoeksmethode	24
5.2 Onderzoekstypering	24
5.2.1 Typeringen	24
5.2.2 Breedte vs. Diepgang	24
5.2.3 Kwantificering vs. Kwalificering	25
5.2.4 Veldonderzoek vs. Bureauonderzoek	25
5.3 Onderzoeksaanpak	26
5.3.1 Beantwoording hoofd- en deelvragen	26
5.4 Onderzoeksstrategie	27
 II Dataverzameling	 28
6 Beschrijving modelmetriekeken	29
6.1 Metriekeken ontwikkeld door Moody	29
6.1.1 Compleetheid	31
6.1.2 Integriteit	31

6.1.3	Flexibiliteit	32
6.1.4	Begrijpbaarheid	33
6.1.5	Correctheid	34
6.1.6	Eenvoud	36
6.1.7	Integratie	36
6.1.8	Implementeerbaarheid	37
6.2	Metrieken door Ann Maes	39
6.2.1	Perceived Ease of Understanding	41
6.2.2	Perceived Usefulness	41
6.2.3	Perceived Semantic Quality	42
6.2.4	User Satisfaction	43
6.3	Metrieken door Geert Poels	44
III	Casus	47
7	Inleiding	48
7.1	De reisorganisatie	48
8	Bevindingen na toepassing modelmetrieken	53
8.1	Compleetheid	53
8.2	Integriteit	56
8.3	Flexibiliteit	57
8.4	Begrijpbaarheid	58
8.5	Correctheid	61
8.6	Eenvoud	62
8.7	Integratie	64
8.8	Implementeerbaarheid	65
9	Toepassing metrieken op een UML model	68
IV	Conclusie	72
10	Beantwoording hoofdvraag	75
10.1	Aanbeveling	78
10.2	Persoonlijke evaluatie	78
11	Vervolgonderzoek	80
V	Appendix	82
12	Appendix - A - Resultaten	83
12.1	Resultaten: toepassing metrieken op de ER modellen	83
12.2	Resultaten: toepassing metrieken op het UML Model	90
13	Appendix - B - Modellen	95

Hoofdstuk 1

Inleiding

1.1 Aanleiding

Volgens Poels en Maes [21][32] zijn conceptuele modellen tegenwoordig de hoeksteen van alle informatiesystemen. Deze modellen helpen ons inzicht te creëren in de structuur en activiteiten van organisaties, de 'stroom' en de aard van de gegevens die een bedrijf heeft en hoe de verschillende software-elementen (objecten, entiteitstypen, functies, taken, enz.) met modelelementen binnen en buiten het definitieve informatiesysteem communiceren. Een conceptueel model is het voornaamste middel om met stakeholders te communiceren die betrokken zijn bij het ontwerp van een informatiesysteem.

Door het dynamisch (het snel kunnen inspelen op de veranderingen van de markt) werken van bedrijven worden bestaande conceptuele modellen al snel overbodig. De conceptuele modellen worden flexibel opgezet om de modellen zo snel mogelijk aan te passen mochten er veranderingen optreden. Hierdoor kan het bedrijf snel en makkelijk inspelen op veranderingen in de markt. Maar door flexibele modellen te creëren wil nog niet zeggen dat ze gemakkelijk aanpasbaar of begrijpbaar zijn. In 2002 is door Olive geobserveerd dat het draait om de kwaliteit van het bestaande conceptueel model [28].

Het volledig opnieuw te gebruiken, in stand te houden, en evolueren van conceptuele modellen zijn volgens Moody [24] cruciaal voor het succes van het geïnstalleerde informatiesysteem. Als het oorspronkelijke model onjuist is, dan zijn alle andere activiteiten die zijn gebaseerd op het conceptueel model vaak onjuist. De kosten om de fouten te corrigeren die over het hoofd zijn gezien tijdens het ontwikkelen van het model stijgt in elke volgende fase van ontwikkeling. Als het model niet aan de eisen van een stakeholder voldoet dan zal het ontwikkelingsproces opnieuw gestart moeten worden, om de eisen als nog in het informatiesysteem te verwerken.

Ondanks dat de eisen van een conceptueel model steeds hoger worden is het moeilijk te achterhalen of een conceptueel model van echte goede kwaliteit is. Het onderzoek naar de kwaliteit van conceptuele modellen staat eigenlijk nog in de kinderschoenen. Het evalueren van de kwaliteit van de conceptuele modellen is nog vaak meer een kunst dan echte wetenschap. Hoewel er verschillende onderzoekers [7][14][23] midden jaren negentig al modelmetrieën ontwikkelde voor Extended Relationship diagrammen. Is er pas recentelijk dit onderzoek uitgebreid en verder gedefinieerd [16][17][24].

Het verbeteren van de kwaliteit van conceptuele modellen is tevens ook een verbetering van de kwaliteit van het te leveren systeem. Terwijl een goed conceptueel model, slecht geïmplementeerd kan worden, kan een slecht model tegen lagere kosten in een later stadium verbeterd worden. Als er vanaf het begin gewerkt wordt met een hoog kwalitatief conceptueel model leidt dit tot een hogere kwaliteit van het uiteindelijke informatiesysteem. Zo kan het conceptueel model van invloed zijn op zowel de efficiëntie (tijd, kosten en inspanning) en de effectiviteit (kwaliteit van resultaten) van een informatiesysteem in ontwikkeling. Bijvoorbeeld een conceptueel model dat kwalitatief slecht is kan zowel de ontwikkeling als de inspanning verhogen (het opsporen en verbeteren van gebreken). Het zou ook kunnen zijn dat het hele systeem niet voldoet aan de eisen van de gebruikers of stakeholders, met als gevolg dat het systeem opnieuw ontwikkeld moet worden om de eisen als nog in het systeem te verwerken.

In een recente studie over de impact van requirementsfouten van Lauesen en Vinter [20] is gebleken dat in de praktijk deze fouten vaak niet worden verbeterd. Dit komt door vaste tijdschema's en begrotingen. Hierdoor is het vaak te duur en / of politiek onaanvaardbaar om requirementsfouten na de eisenanalyse te verbeteren. Het resultaat is dat het uiteindelijke software door de requirementsfouten van lagere kwaliteit zal zijn.

Het probleem zit denk ik in de definitie van kwaliteit. Als je in omgeving vraagt wat mensen verstaan onder kwaliteit, heeft iedereen een andere definitie. Als er een kwaliteitsdefinitie word gegeven dan zijn ze meestal vaag, ingewikkeld en/of is er geen onderliggende structuur aanwezig [32]. Voorgaande studies tonen aan dat de kwaliteit van een conceptueel model de key factor is voor de kwaliteit van het uiteindelijk te implementeren systeem [9]. Wat gebeurt er als je niet weet of de kwaliteit van je model goed is? Dan zal misschien het systeem uiteindelijk wel falen. Er zijn metrieken ontwikkeld om bijvoorbeeld de kwaliteit objectief te kunnen meten [14][21][23]. Deze metrieken zouden ontwikkelaars kunnen helpen om in een vroeg stadium het model te kunnen evalueren. Na de evaluatie van het model zou het model aangepast en verbeterd kunnen worden. Hoewel het meten soms ook gebeurt door een expert, zal de expert altijd een subjectief oordeel geven. Volgens Maes en Poels [21] is de expert niet altijd betrokken bij het ontwikkelen van een informatiesysteem, hierdoor zou je een verkeerd subjectief oordeel kunnen krijgen, omdat de expert gewoon niet bekend is binnen het Universe of Discourse (UoD).

Doordat dit onderzoeksgebied nog in de kinderschoenen staat, is volgens Moody [22] iedere onderzoeker maar ergens mee bezig. Moody [22] beschrijft ook dat er teveel onderzoekers bezig zijn met het onderzoeken naar frameworks. Deze frameworks zijn ook belangrijk voor het evalueren van kwaliteit van conceptuele modellen. Hierdoor weet je niet welke framework je moet gaan gebruiken. Inmiddels zijn er volgens Moody [22] al rond de 50 quality frameworks ontwikkeld, Moody vindt dat de onderzoekers zich bezig moeten houden met een standaard voor conceptuele modellen met als uitgangspunt de syntax en semantiek. Doordat er nog geen standaard is, word het onderzoeksgebied voor conceptuele modellen onduidelijk. Dit geldt hetzelfde voor de modelmetrieken. Want er is een standaard nodig om modelmetrieken te ontwerpen. Pas als er een standaard voor conceptuele modellen is zal het ontwikkelen van modelmetrieken makkelijker worden. Hoewel het onderzoeksgebied wordt uitgebreid door modelmetrieken en zijn er al verschillende programma's in ontwikkeling om modellen te evalueren op kwaliteit [19], maar het blijft een onoverzichtelijk geheel.

Eigen ervaring:

Bij mijn studie Hogere Informatica in Den Bosch, moest ik veel werken met UML modellen. De modellen gebruikte ik om programma's en databases te ontwerpen. Het viel me toen al op dat er iedereen een eigen kijk heeft hoe een UML-model eruit moet zien. Geef tien mensen de opdracht om een database model te maken van een voetbalvereniging en je zult zien dat iedereen een ander model ontwikkelt. Het probleem wat hier aan de orde komt welk model is nu goed? Modelmetrieken zorgen ervoor dat je in een vroeg stadium kan controleren op fouten en gebreken. Zodat je deze fouten en / of gebreken niet hoeft op te zoeken of op te lossen als je klaar bent met ontwikkelen. Het is alleen niet duidelijk welke modelmetrieken er op dit moment ontwikkeld zijn. Het is belangrijk om een overzicht te maken van de ontwikkelde modelmetrieken, om zo te voorkomen dat niet dezelfde metrieken worden ontwikkeld. Of dat metrieken voor ER modellen ook gebruikt kunnen worden voor UML modellen. Met de uitkomst van dit onderzoek zou er meer gericht kunnen worden op het verder ontwikkelen van metrieken of het valideren van de metrieken.

1.2 Leeswijzer

Hieronder wordt in het kort toegelicht hoe de structuur van deze afstudeerscriptie is opgebouwd. Dit document bestaat uit vier delen, namelijk:

Deel I - Introductie

In het introductie gedeelte wordt de probleemstelling gedefinieerd, de verantwoording voor dit onderzoek gegeven, het theoretische kader voorgelegd en wordt er beschreven hoe het onderzoek is verricht. Het doel van dit deel is de lezer meer inzicht te geven in de onderzoeksmethodiek.

- **Hoofdstuk 2 - Probleemstelling**

In dit hoofdstuk wordt de centrale onderzoeksvraag gepresenteerd, komen de belangrijkste deelvragen aan bod en wordt de probleemstelling en de onderzoeksdoelstelling van deze afstudeerscriptie beschreven.

- **Hoofdstuk 3 - Verantwoording**

In dit hoofdstuk geef ik de reden voor dit onderzoek aan. Zo worden onder andere het theoretische, praktische en maatschappelijke relevantie van het onderzoek toegelicht.

- **Hoofdstuk 4 - Theoretisch kader**

Hier wordt het kennisgebied van het onderzoek gespecificeerd, komen de relevante inhoudelijke keuzes en veronderstellingen aanbod, het onderzoek tot nu toe op het gebied van modelmetrieën en worden de belangrijkste termen / concepten die in dit onderzoek gebruikt worden toegelicht.

- **Hoofdstuk 5 - Onderzoeksstructuur**

In dit hoofdstuk wordt de structuur van het onderzoek nader toegelicht. Zo zal de onderzoeksmethode worden beschreven, word de onderzoekstypering uitgelegd, de onderzoeksaanpak zal worden beschreven, hoe de deelvragen worden behandeld en wordt er verteld hoe de centrale onderzoeksvraag beantwoord wordt.

Deel II - Dataverzameling

In deel twee komt de dataverzameling aanbod. De naam zegt het eigenlijk al, in dit deel van de scriptie zal de gevonden literatuur (de data) worden samengevat en verwerkt. Ook worden in dit deel twee deelvragen beantwoord.

- **Hoofdstuk 6 - Beschrijving modelmetrieën**

De beschrijving van de modelmetrieën wordt per auteur in dit hoofdstuk behandeld. Door middel van deze beschrijving zullen twee deelvragen worden beantwoord. Zo zal er naar voren komen welke kwaliteitseigenschappen de modelmetrieën evalueren.

Deel III - Casus

In deel drie komt de casus aan bod. Op basis van deze casus worden verschillende modellen geïntroduceerd. De modelmetrieën worden op de geïntroduceerde modellen toegepast. Door middel van de bevindingen worden er in hoofdstuk acht en negen samen drie deelvragen beantwoord.

- **Hoofdstuk 7 - Inleiding**

Dit hoofdstuk is een korte beschrijving van de casus en wat u verder kunt verwachten.

- **Hoofdstuk 8 - Bevindingen na toepassing modelmetrieën**

Na het toepassen van de modelmetrieën word er in dit hoofdstuk de bevindingen gepresenteerd. Ook word er ingegaan op overeenkomsten en verschillen

tussen de metrieken onderling. Dit hoofdstuk word afgesloten met een kort overzicht tussen de verschillen en overeenkomsten in tabelvorm.

- **Hoofdstuk 9 - Toepassing metrieken op een UML model**

De ER metrieken zullen nogmaals worden toegepast maar dan op een UML-model. In dit hoofdstuk zal er dus een UML-model worden gepresenteerd. Die ook ontwikkeld is op basis van de casus beschreven in hoofdstuk 7.

Deel IV - Conclusie

De conclusie van het onderzoek zal in dit deel worden beschreven. Om te beginnen met een algemene conclusie over de werking van de metrieken om later dieper in te gaan op de onderzoeksvraag.

- **Hoofdstuk 10 - Beantwoording hoofdvraag**

De hoofdvraag zal op basis van de beantwoorde deelvragen hier beantwoord worden. Ook is er een aanbeveling te lezen en een persoonlijke evaluatie over de gemaakte keuzes.

- **Hoofdstuk 11 - Vervolgonderzoek**

Het onderzoek word afgesloten met een beschrijving over mogelijke vervolgonderzoeken.

Deel V - Appendix

In de appendix staan alle resultaten en modellen die nodig zijn voor het beantwoorden van verschillende deelvragen.

- **Hoofdstuk 12 - Appendix - A - Resultaten**

De gedetailleerde resultaten van het toepassen van de modelmetrieken kunnen in deze appendix worden terug gelezen.

- **Hoofdstuk 13 - Appendix - B - Modellen**

De modellen die zijn ontwikkeld door Marc Pieterse zijn terug te vinden in deze appendix.

Deel I

Introductie

Hoofdstuk 2

Probleemstelling

Binnen de ICT wereld wordt vaak gebruik gemaakt van modellen [32], om bijvoorbeeld de overzichtelijkheid van een programma te verbeteren of om de begripbaarheid te verhogen. Dan is de keuze voor een goede representatie van data in een conceptueel model waarschijnlijk een van de cruciale taken in het systeemontwerpproces. Want als de eisen van een informatiesysteem niet of gebrekkig in het model zijn vertegenwoordigd kan dit voor extra ontwikkelkosten zorgen. Deze gebreken zullen na het ontwikkelen als nog in gebouwd moeten worden. Hoewel het modelleren een kleine fase is in het hele ontwerpproces heeft fase deze waarschijnlijk de grootste impact [24][39], vergeleken met de andere fasen in het ontwerpproces.

Waarom is het maken van modellen zo belangrijk?

De kwaliteit van software is meestal gebaseerd op het eindproduct [1] [41]. Maar uit studies blijkt dat meer dan de helft van de fouten die optreden tijdens de ontwikkeling van systemen requirementsfouten zijn [9][20]. Requirementsfouten zijn de meeste voorkomende oorzaak van verstoringen tijdens ontwikkeling van systemen en projecten [9]. De kosten van een fout loopt exponentieel toe tijdens het ontwikkelen. Het is duurder om een defect na de totstandkoming van het systeem te repareren dan tijdens de eisenanalyse [2].

Poels [31] beschrijft dat begripbaarheid van model een belangrijke factor is van de onderhoudbaarheid. Om onderhoud te kunnen plegen op een model dien je eerst het model uitvoerig te begrijpen voordat je überhaupt kan beginnen met het aanpassen van het model. Modellen spelen een belangrijke rol bij systeemontwikkeling, hoe sneller er een model geëvalueerd kan worden des te groter de kwaliteit van het uiteindelijke systeem wordt. Het evalueren van modellen tijdens het ontwikkelen heeft als voordeel dat je fouten er al uit kunt halen voordat het informatiesysteem wordt gebouwd. De zoektocht naar goede kwaliteitsmetingen en metrieken is nog steeds in volle gang. [32]

Zonder aan experts te vragen of de modellen goed zijn, is er geen manier om zelf te achterhalen of het model dat ontwikkeld is ook daadwerkelijk het doel behaald wat de maker voor ogen heeft. Modelmetrieken bieden hier een uitkomst. Met modelmetrieken zou u bijvoorbeeld kunnen berekenen of de begripbaarheid van een model hoog is. Zodat aan het einde van de ontwikkeling geen missers zijn gemaakt. Of door eisen die niet zijn opgenomen en hierdoor voor extra ontwikkelkosten zorgen. Want de eisen die niet zijn opgenomen dienen in een later stadium als nog toegevoegd te worden. Niet alle metrieken zijn toepasbaar op verschillende typen modellen. Stel dat er een modelmetriek voor ER-diagrammen is ontwikkeld, geldt dat deze metriek ook toe te passen is op UML-diagrammen?

Op de studie van Poels [31] na zijn er eigenlijk geen onderzoeken gedaan naar modelmetrieken. Poels heeft in zijn studie wat modelmetrieken bekeken en uiteindelijk er zelf voor gekozen om modelmetrieken te ontwikkelen voor ER-modellen. In deze studie van Poels kwam naar voren dat er bijna geen validatie was van de modelmetrieken, want de ontwikkelde modelmetrieken zijn bijna tot niet toegepast in de praktijk. Doordat de modelmetrieken niet worden toegepast blijven onderzoekers modelmetrieken ontwikkelen. Met als resultaat dat het onderzoeksgebied onoverzichtelijk wordt. Nu wil ik met dit onderzoek gaan kijken naar de ontwikkelde modelmetrieken en deze toepassen op een zelf gemaakt ER- en UML-model. Om uiteindelijk van de toegepaste modelmetrieken een evaluatie te geven.

2.1 Onderzoeksvraag

De onderzoeksvraag die centraal staat in deze scriptie is:

Kunnen de modelmetrieken die zijn ontwikkeld voor ER-modellen worden toegepast op UML-modellen?

Om het beantwoorden van de onderzoeksvraag makkelijker te maken heb ik verschillende deelvragen bedacht die me helpen om uiteindelijk de onderzoeksvraag te beantwoorden. Ik maak gebruik van de volgende deelvragen:

1. Welke metrieken zijn er door verschillende auteurs ontwikkeld?
2. Voor welke doeleinden zijn de metrieken ontwikkeld?
3. Wat zijn de overeenkomsten en verschillen tussen de verschillende metrieken?
4. Zijn de ER metrieken toepasbaar op een UML-model?
5. Welke problemen kunnen optreden tijdens het toepassen van ER metrieken op UML-modellen?

Hoofdstuk 3

Verantwoording

3.1 Waarom dit onderzoek?

Het onderzoek dat ik uitvoer dient duidelijk te maken welke modelmetrieken voor ER-modellen in de loop der jaren zijn ontwikkeld [7][14][23]. Doordat verschillende onderzoekers modelmetrieken hebben ontwikkeld is het niet duidelijk welke modelmetrieken er nu ontwikkeld zijn. Om overzichtelijkheid over modelmetrieken te scheppen is dit onderzoek opgestart. Dit is om te voorkomen dat onderzoekers niet dezelfde metrieken gaan ontwikkelen. Om ervoor te zorgen dat de onderzoekers de bestaande modelmetrieken verder ontwikkelen en toepassen, in plaats van nieuwe metrieken te ontwerpen.

Aan de hand van een eigen casus (zie deel drie) zal ik bekijken of de ontwikkelde modelmetrieken makkelijk in de praktijk te gebruiken zijn. Om hier een aanbeveling op te baseren. Ook word er gekeken naar de modelmetrieken die bijvoorbeeld voor ER-diagrammen zijn ontwikkeld of deze ook gebruikt kunnen worden op UML modellen, zodat de onderzoekers de metrieken beter kunnen ontwikkelen in plaats van nieuwe te ontwerpen. Het onderzoek zal zich voornamelijk richten op ER en EER aangezien hier de meeste modelmetrieken voor ontwikkeld zijn die ik in de literatuur heb kunnen vinden.

3.2 Theoretische relevantie

Conceptuele modellen beginnen steeds meer terrein te winnen binnen de ontwikkeling van informatiesystemen. Er zijn onderzoeken geweest naar de correctheid en kwaliteit van informatiesystemen als evaluatie van het project, maar deze onderzoeken worden pas gestart aan het einde van de ontwikkeling. Volgens Moody [22] is het dringend nodig dat er een standaard komt hoe je conceptuele modellen moet opzetten. Voor "Software Quality" [13] is er al een standaard te vinden, opgezet door International Standards Organisation (ISO) en International Electrotechnical Commission (IEC). Dit wilt Moody [22] ook voor conceptuele modellen, omdat elke onderzoeker nu maar wat doet, in plaats van onderzoek doen naar een goede standaard.

Met conceptuele modellen loopt men achter de feiten aan. Door gebrek aan goede evaluatie technieken weet men niet of een conceptueel model kwalitatief goed is. Door het gebrek aan goede evaluatie middelen zullen de conceptuele modellen die over het algemeen de aanzet zijn voor het te ontwikkelen informatiesysteem niet in kwaliteit verbeteren. Met als resultaat dat het informatiesysteem gebrekkig of

volkomen verkeerd geïmplementeerd word, met de nodige kosten van dien om alles als nog te verbeteren.

Daarom is het belangrijk dat er in kaart word gebracht welke modelmetrieken er zijn en hoe deze te gebruiken zijn. Het houdt natuurlijk niet op bij modelmetrieken, want er is sowieso een grote slag te slaan op het gebied van evalueren van conceptuele modellen. U moet hierbij denken aan frameworks en tools of programma's om conceptuele modellen te evalueren. Dit onderzoek kan dienen als basis voor uitbreiding van al bestaande modelmetrieken of modelmetrieken die kunnen worden verwerkt in een programma of tool.

3.3 Praktische relevantie

Indien uit dit onderzoek blijkt dat de modelmetrieken makkelijk en nuttig zijn om conceptuele modellen op kwaliteit te beoordelen, zal dit gunstig zijn het ontwikkelproces voor informatiesystemen. Hiermee kun je tijdens de ontwerpfase van de conceptuele modellen al toetsen op kwaliteit om zo onnodige fouten te voorkomen. Deze kunnen wellicht in een later stadium voor extra kosten kunnen zorgen, bijvoorbeeld doordat de fouten opgezocht dienen te worden of dat bepaalde eisen niet terugkomen in het informatiesysteem.

De kennis uit dit onderzoek kan worden gebruikt hoe modelmetrieken kunnen worden toegepast op een model. Ook zal dit onderzoek modelmetrieken van ER-modellen toepassen op een UML-model, om zo te kijken of het mogelijk is modelmetrieken toe te passen op een ander model dan waarvoor het eigenlijk ontwikkeld is. De reden dat in dit onderzoek een ER metriek word toegepast op een UML model, is om te kijken of er daadwerkelijk nieuwe metrieken ontwikkeld moeten worden of dat de bestaande metrieken voldoen of alleen maar aangepast dienen te worden.

3.4 Maatschappelijke relevantie

De maatschappelijke relevantie van dit onderzoek zit hem in het te ontwikkelen informatiesysteem. Als het conceptuele model niet gecontroleerd of geevalueerd kan worden zou dit slechte software op kunnen leveren. Aangezien we in ons dagelijks leven steeds meer afhankelijk worden van informatiesystemen is het wenselijk dat de software die we gebruiken in bijvoorbeeld ziekenhuizen van goede kwaliteit is. Conceptuele modellen worden al gebruikt voor het ontwikkelen voor informatiesystemen toch is het lastig om te bekijken of een conceptueel model nu van goede kwaliteit is. Modelmetrieken zorgen ervoor dat conceptuele modellen eerder in het ontwikkelproces geevalueerd kunnen worden. Hierdoor zou de software in kwaliteit verbeterd kunnen worden en hoeven er ook minder kosten gemaakt te worden. Dit komt omdat er achteraf geen eisen toegevoegd kunnen worden.

Aan het eind van het onderzoek zal voor eenieder die met ER-modellen werkt handreikingen geven welke modelmetrieken je kunt gebruiken om conceptuele modellen te kunnen evalueren. Ook zal uit dit onderzoek voortkomen welke modelmetrieken makkelijk toepasbaar zijn op verschillende modellen. Met als uiteindelijke doel dat de software die geproduceerd word voor maatschappelijke doeleinden van betere kwaliteit is.

3.5 Doelgroep

Dit onderzoek zal voor mensen die gebruik maken van UML- en ER-modellen interessant zijn. Of voor mensen die gebruik maken van conceptuele modellen in het algemeen, omdat er een overzicht wordt gepresenteerd van alle ontwikkelde modelmetrieken op dit moment. Aan de hand van het resultaat kunnen zij bestaande modellen evalueren en indien nog verbeteren. Tevens zal dit onderzoek een aanbeveling geven voor een aantal metrieken die toegepast zijn aan de hand van een casus (zie deel drie van deze scriptie).

Hoofdstuk 4

Theoretisch Kader

4.1 Kennisgebied

Het kennisgebied wat hoort bij dit onderzoek is informatiekunde, met name Model Based Software Engineering (MBSD). Hoe de metrieken opgebouwd zijn is voor dit onderzoek is minder van belang. Aangezien dit onderzoek een overzicht zal geven over de verschillende modelmetrieken die ontwikkeld zijn voor ER modellen en hoe deze toe te passen zijn op UML modellen.

4.1.1 Softwaremetrieken versus Modelmetrieken

Een veel voorkomend probleem bij software-engineering is dat de software fouten bevat. Het voorkomen van deze fouten is volgens Raed [35] zeer lastig of bijna onmogelijk. Als het niet lukt om deze fouten te voorkomen, kan er op zijn minst geprobeerd worden om deze fouten achteraf op te sporen en op te lossen. Het testen van software om deze fouten op te sporen en te verbeteren heeft een grote impact op de resources en tijd. Daarom is het wenselijk als de tester word geholpen met gerichte informatie over de fouten. Softwaremetrieken worden gebruikt om de tester van deze informatie te voorzien.

Softwaremetrieken zijn bijna net zo oud als software-engineering. De studies die zijn uitgevoerd in der loop der jaren op het gebied van softwaremetrieken hebben weinig tot geen impact gehad op de praktijk. Terwijl de softwaremetrieken bedoeld zijn om de software te verbeteren [10]. Ten opzichte van softwaremetrieken bestaan modelmetrieken nog niet heel erg lang. De ontwikkeling van modelmetrieken is eigenlijk 'net' begonnen. Moody [23] is een van de eerste in 1998 die informele modelmetrieken heeft ontwikkeld. In een later onderzoek [22] geeft hij aan dat er bijna tot geen vooruitgang is geboekt op het evalueren van kwaliteit van een conceptueel model. Dit komt doordat er nog geen goede standaard zou zijn voor het maken van conceptuele modellen, terwijl de standaard voor softwaremetrieken al langer bestaan. Hierdoor is het risico groot dat de conceptuele modellen van slechte kwaliteit zijn.

Het software-engineeringsproces begint tegenwoordig vaak met het ontwikkelen van conceptuele modellen [21][32]. Aan de hand van deze conceptuele modellen word de software ontwikkeld. Wat gebeurt er als het conceptuele model van slechte kwaliteit is? Zonder dat je kunt meten en weten of het conceptueel model in eerste instantie slecht is. Stel er wordt software ontwikkeld op basis van dit model, dan kan het zijn dat de software uiteindelijk ook van slechte kwaliteit is. Het gebruik maken van soft-

waremetrieken zouden fouten kunnen opsporen, maar deze fouten kunnen wellicht voorkomen kunnen worden door het conceptueel model op kwaliteit te beoordelen.

De kwaliteit van software ligt voor een groot deel bij conceptuele modellen. Ten behoeve van de kwaliteit van software, is het dus belangrijk dat het conceptueel model geëvalueerd moet worden op kwaliteit. Om er zeker van te zijn dat de software in de toekomst minder fouten bevatten dienen de modelmetrieken verder onderzocht en ontwikkeld te worden. Dit onderzoek bekijkt aan de hand van een ER en UML diagram drie verschillende metrieken [21][23][31].

4.1.2 Formele versus Informele modelmetrieken

Dit onderzoek is voornamelijk gebaseerd op informele modelmetrieken met andere woorden beschrijvende modelmetrieken. Het verschil tussen formele en informele modelmetrieken is de exacte benadering van de formele modelmetrieken. De formele modelmetrieken hebben ook een andere eigenschap, de uitkomsten van deze modelmetrieken zijn meestal objectief. De uitkomst van een formele modelmetriek is meestal een getal of een waarde. Deze modelmetrieken zijn daarom handig te verwerken in een programma of tool. Formele modelmetrieken leveren daarbij ook altijd hetzelfde antwoord, het maakt dus ook niet uit wie deze metrieken gebruikt.

Formele modelmetrieken zijn gepresenteerd in verschillende vormen, van simpele getallen [31], in algebra [4] tot predicaatlogica [7]. De informele modelmetrieken zijn subjectief dit wil zeggen dat er een persoon nodig is die de metrieken uitvoert. Subjectiviteit is voornamelijk gebaseerd op de interpretatie van de uitvoerder. Het kan dus voorkomen dat de uitkomst per keer dat het model door de uitvoerder bekeken wordt verandert. Nu lijkt het net als informele modelmetrieken niet bruikbaar zijn omdat de interpretatie van het conceptueel model belangrijk is bij het toepassen van de metriek.

Het ontwerpen van conceptuele modellen is een denkproces hoe een bepaald probleemdomen (UoD) in een conceptueel model word gegoten. Informele metrieken kunnen helpen bij dit denkproces. Hiermee wil ik zeggen dat de informele modelmetrieken je instaat stellen om het conceptueel model vanuit een ander standpunt te bekijken. Of om op je conceptueel model te bekijken met een kritische blik, door af te vragen waarom bepaalde modelmetrieken zijn gemodelleerd en waarom niet. Formele metrieken geven een getal of een waarde, op dat moment zou je daar niet zoveel mee kunnen. Pas als het conceptueel model af is zouden de informele modelmetrieken bruikbaar zijn.

De auteurs Moody [23] en Maes [21] hebben geheel informele metrieken ontwikkeld. Kesh [14] heeft een combinatie tussen formeel en informeel ontwikkeld. Poels [31] heeft formele metrieken ontwikkeld.

4.2 Termen

Tijdens deze scriptie zult u een aantal termen kunnen tegenkomen waarvan u mogelijk de betekenis niet van weet. Om duidelijkheid te verschaffen over deze termen word er in deze sectie de termen opgesomd en uitgelegd wat deze precies betekenen.

Wat is een model?

Een model is een schematisch weergave van de werkelijkheid. Volgens Erik Proper

[34] is een model gebaseerd op een geabstraheerd domein of een deel van een domein. Dit domein is eigenlijk het gebied waar de stakeholder in is geïnteresseerd. Als ik bijvoorbeeld geïnteresseerd ben in voetbal dan is mijn domein voetbal. Het is moeilijk om alles in een keer te bevatten wat met voetbal te maken heeft. Door middel van een conceptueel model kan je op een simpele manier inzicht krijgen wat voetbal precies inhoudt. In de rest van mijn scriptie gebruik ik de definitie die ik hierboven van Erik Proper geciteerd heb.

Wat zijn metrieken nu eigenlijk?

Nu weten we wat een model precies inhoudt, maar we hebben nog geen inzicht wat nu precies een metriek is. Mijn eigen idee bij een metriek is een manier om eigenschappen van model te kunnen uitdrukken of evalueren. Een model heeft verschillende eigenschappen bijvoorbeeld; begrijpbaarheid, correctheid, uitbreidbaarheid, etc. Met behulp van metrieken kunnen je deze verschillende eigenschappen toetsen. Een metriek is dus een middel om conceptuele modellen te toetsen.

Wat zijn requirements?

Requirements zijn de eisen die stakeholders kunnen hebben bij een bepaald project. Deze eisen dienen opgenomen te worden in het conceptueel model. Eisen vertegenwoordigen stakeholders in een conceptueel model, zonder deze eisen op te nemen zal het uiteindelijke informatiesysteem niet te gebruiken zijn voor die stakeholder(s).

Wat is Universe of Discourse (UoD)?

Het UoD word in het Nederlands ook wel Universum of Discussie genoemd. Bij het modelleren is het universum een probleemdomein waar het conceptueel model voor word gemaakt. Om het voorbeeld van de voetbalvereniging er weer bij te halen. Het probleemdomein zal alles bevatten wat een voetbalvereniging ook bevat. Door te discussieren over dit probleemdomein krijg je dus een discussie over een bepaald universum, dit universum is dus de voetbalvereniging.

Wat houdt ER in?

ER is een afkorting voor Extended Relationship diagrammen. ER diagrammen is een taal voornamelijk om database te ontwikkelen op basis van een model. De entiteitstypen in het model zijn de tabellen in de database en de attributen zijn de kolommen / velden in een database.

Wat houdt UML in?

Net zoals ER is UML een modelleertaal. UML is een afkorting voor Unified Modelling Language, dit houdt in dat er verschillende typen modellen te maken zijn in UML. De verzamelnaam om via deze "taal" conceptuele modellen te maken, worden UML-modellen genoemd.

Wat is een viewpoint?

Zonder dat we het weten gebruikt iedereen viewpoints. Viewpoints zijn te omschrijven als standpunten die een persoon inneemt tijdens het bekijken van objecten, tijdens het communiceren met mensen, tijdens het werken en voornamelijk tijdens het modelleren [34]. Vooral de laatste definitie van een viewpoint is belangrijk. Want iedereen heeft een eigen interpretatie van een conceptueel model. Dit betekent dat iemand een standpunt inneemt hoe hij of zij het model wilt zien.

Wat zijn business rules?

Business rules zijn de bedrijfsregels van een organisatie.

Wat zijn stakeholders?

Stakeholders zijn personen die een bepaald belang hebben in een project. Ook kunnen stakeholders baat hebben bij een bepaald project. De stakeholders kunnen individuen, afdelingen, organisaties, etc zijn.

4.3 Keuzes en Veronderstellingen

Veronderstellingen:

Met dit onderzoek denk ik te kunnen aantonen dat de beschreven metrieken toepasbaar zijn op een UML-model. Maar ik denk dat de metrieken die ontwikkeld zijn voor een bepaald model en ook alleen maar toepasbaar zijn op dit model. Bijvoorbeeld de ER metrieken die gebaseerd zijn op ER conventies zullen niet toepasbaar zijn op een UML model. De metrieken die wel toepasbaar zijn zullen voornamelijk de informele metrieken zijn.

Keuzes:

Ik heb een keuze gemaakt voor ER metrieken aangezien hier de meeste metrieken tot nu toe voor ontwikkeld zijn. Daarbij is het ontwerpen en het gebruiken van ER en UML modellen me bekend door voorgaande studies. Op basis van het onderzoek van Poels [31] heb ik de metrieken uitgekozen die voor ER diagrammen ontwikkeld zijn.

Er zullen twee vormen van ER modellen en procesmodellen worden gepresenteerd op basis van dezelfde casus. Daarbij wordt er één UML model gepresenteerd ook aan de hand van deze casus. Ik heb gekozen voor de twee ER modellen zodat ik een betere vergelijkingen kan maken, na de toepassing van de metrieken. Het UML model is ontworpen door Marc Pieterse, ik heb dit aan hem gevraagd zodat ik het ontwerp niet kon beïnvloeden met de metrieken die ik al had toegepast op de ER modellen.

In de casus zijn er geen business scenario's en budgetten gemaakt. Aangezien dit niet relevant is voor dit onderzoek. Het gaat om de toepassing en de bevindingen van de metrieken niet zozeer om uitkomsten / resultaten.

Het zijn voornamelijk de informele metrieken die in deze scriptie toegepast worden komen omdat ik denk dat deze metrieken ook toepasbaar zijn op UML modellen.

De eerste twee en de laatste twee deelvragen van dit onderzoek staan in elkaars verlengde. Daarom worden de deelvragen per twee in een hoofdstuk beantwoord.

4.4 Onderzoek tot nu toe

Poels [31] beschrijft dat er aantal kwaliteitsframeworks zijn voortgekomen in het onderzoek van Moody [22]. Volgens Moody bieden de meeste frameworks kwaliteitsdefinities en criteria voor de conceptuele modellen; enkele van deze frameworks bevatten ook kwantitatieve manieren om conceptuele modellen te evalueren op kwaliteit en op een objectieve en kostenefficiënte manier. Poels [31] heeft een overzicht gemaakt van het onderzoek naar model metrieken voor conceptuele modellen tot nu toe. Zie tabel 4.1 op bladzijde 22.

Eick [7] beschrijft één metriek voor S-diagrammen. Eick is op basis van drie kwaliteitsaspecten te werk gegaan. Expressiviteit, complexiteit en normality zitten in deze ene metriek. Gray et al. [lit] heeft een voorstel gedaan van metrieken voor de

evaluatie van kwaliteit van ER modellen. Poels beschrijft dat het verband met ER diagrammen niet terug te vinden is. Kesh [14] gebruikt één metriek om de kwaliteit te meten van ER diagrammen. Kesh combineert de verschillende metrieken op basis van Likert grafieken die op een subjectieve manier beoordeelt dienen te worden. Moody [24] heeft een set van metrieken [23] ontwikkeld en deze toegepast. Uit de studie van Moody [24] is gebleken dat vier van de 25 metrieken die zijn voorgesteld in [23], opwegen tegen de kosten van het verzamelen. Cherfi et al. [4] heeft ook een framework voorgedragen met een corresponderende set van metrieken. Niet alle voorgedragen metrieken zijn objectief. Maes en Poels [21] hebben metrieken ontwikkeld op basis van vier aspecten. Deze vier aspecten zijn op hun beurt weer gerelateerd aan gebruikers standpunten.

Poels [31] en Moody [22] beschrijven beide dat er te weinig concrete voorstellen zijn toegepast in de praktijk en dat er bijna tot geen validatie van de model metrieken. Tevens is het aantal ontwikkelde frameworks volgens Moody uit de hand gelopen. Hierdoor is het onderzoeksgebied verzadigd geraakt, terwijl het nodig is om een standaard te ontwikkelen voor conceptuele modellen.

Tabel 4.1: Poels [31] ontwikkelde modelmetrieën

Authors	Quality Focus	Scope	Objective/Subjective?	Validation?	Tool?
Eick [7]	Expressiveness, complexity, normalizedness	S-Diagrams	Objective	No	No
Gray [2]	Complexity, deviation from third normal form	ER diagram	Objective	Unknown	Yes
Kesh [14]	Ontological quality, behavioural quality	ER diagram	Objective and subjective	No	Yes
Moody [23]	Completeness, integrity, flexibility, understandability, correctness, simplicity, integration, implementability	ER diagram	Objective and subjective	Number of entities and relationships metric validated in an action research study	Yes
Cherfi [4]	Specification dimension: legibility, expressiveness, simplicity, correctness Usage dimension: completeness, understandability Implementation dimension: implementability, maintainability	Extended ER diagram	Objective and subjective	No	Yes
Maes [21]	Perceived semantic quality, ease of use, usefulness, user satisfaction	Graphical conceptual models	Subjective	Yes (via two controlled experiments)	No

Hoofdstuk 5

Onderzoeksstructuur

5.1 Onderzoeksmethode

Volgens de classificatie van Verschuren [42] is dit onderzoek, deels probleemsig-nalerend en deels een evaluatieonderzoek. Probleemsig-nalerend omdat het onder-zoek zich richt op het evalueren van kwaliteit van conceptuele modellen, dat op dit moment bijna tot niet mogelijk is om te evalueren. Hierbij kan er een onderscheid worden gemaakt tussen de huidige situatie waar geen kwaliteit te evalueren valt en de wenselijke situatie waar kwaliteit te meten zou zijn aan de hand van modelme-trieken.

De andere helft van dit onderzoek is evaluatie gericht. Onderzoekers hebben model-metriekeken bedacht om de kwaliteit van conceptuele modellen te kunnen evalueren, een soort interventie voordat er verder word ontwikkeld op de slechte kwaliteit con-ceptuele modellen.

5.2 Onderzoekstypering

5.2.1 Typeringen

Verschuren gebruikt de volgende definitie voor het typeren van een onderzoek:

De meest bepalende beslissing die u bij het maken van een technisch ontwerp voor uw afstudeerproject moet nemen is de keuze van een aanpak, ofwel onderzoeksstrate-gie. Onder een onderzoeksstrategie verstaan wij een geheel van met elkaar samen-hangende beslissingen over de wijze waarop u het onderzoek gaat uitvoeren.

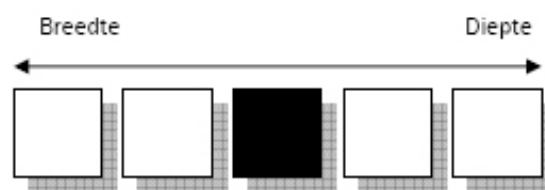
Hierin wordt beschreven dat de wijze waarop een onderzoek dient te worden uit-gevoerd gebaseerd is op een aantal kernbeslissingen. Deze worden bepaald door de breedte of diepgang, kwantificering of kwalificering en veldonderzoek of bureauon-derzoek. Deze kernbeslissingen zijn ook toegepast op dit onderzoek en worden hi-eronder nader beschreven.

5.2.2 Breedte vs. Diepgang

Hoewel dit onderzoek niet de intentie heeft volledig te zijn, aangezien er nog voort-durend onderzoek wordt gedaan na modelmetriekeken, wordt er wel geprobeerd enige diepgang mee te geven. Desondanks is ook de breedte van belang om een helder

beeld te scheppen van de al ontwikkelde modelmetriekeken. Dit wordt gedaan door de modelmetriekeken toe te lichten en toe te passen op een casus (zie deel drie). De diepgang wordt verkregen door dat het onderzoek een duidelijk overzicht schept van de ontwikkelde modelmetriekeken. Deze zullen opgesomd worden in deel twee van deze scriptie. De breedte wordt gehaald uit het feit dat zowel de beschrijving van de modelmetriekeken aanbod komt maar ook hoe je deze kan toepassen. Om inzicht te krijgen over alle ontwikkelde modelmetriekeken is er een brede kijk nodig.

De diepgang wordt verkregen door dat het onderzoek een duidelijk beeld schept van de ontwikkelde modelmetriekeken. Waarvan drie onderzoekers zijn uitgekozen om deze modelmetriekeken toe te passen en uitvoerig te beschrijven wat de bevindingen zijn na het toepassen.



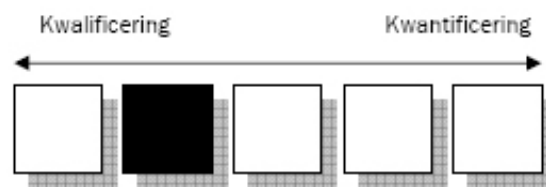
Figuur 5.1: Breedte versus Diepgang

5.2.3 Kwantificering vs. Kwalificering

Verschuren gebruiken de volgende definitie voor de uiteenzetting van kwalitatief en kwantitatief onderzoek:

Hebben we het over het feitelijke proces van meting (kwantitatief onderzoek) of registratie en beschrijving (kwalitatief onderzoek).

Dit onderzoek zal meer naar het kwalitatieve onderzoek leunen dan naar het kwantitatieve. Dit komt doordat in dit onderzoek een beschrijving van de gekozen modelmetriekeken wordt gegeven en daarbij de beschrijving van alle bevindingen, nadat de modelmetriekeken zijn toegepast.



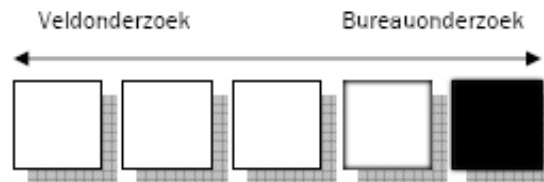
Figuur 5.2: Kwalificering versus Kwantificering

5.2.4 Veldonderzoek vs. Bureauonderzoek

Verschuren beschrijft de volgende definitie in zijn boek over veld- en bureauonderzoek:

Het bureauonderzoek speelt zich hoofdzakelijk af achter het bureau, waarbij veldonderzoek in de praktijk of het 'veld' plaatsvindt.

Dit onderzoek heeft het meest weg van een bureauonderzoek, omdat er uit bestaande literatuur informatie van modelmetriekeken wordt gehaald. Deze modelmetriekeken zullen worden toegepast op een eigen ontworpen casus. Hiervoor zal ik dus niet het 'veld' in hoeven.



Figuur 5.3: Veldonderzoek versus Bureauonderzoek

5.3 Onderzoeksaanpak

Het onderzoek is opgedeeld in twee delen. Het eerste deel van deze scriptie is de beschrijving van modelmetriekeken ontwikkeld door drie verschillende onderzoekers. De modelmetriekeken zullen per onderzoeker worden behandeld. Het tweede deel van deze scriptie zal de toepassing van de modelmetriekeken worden behandeld met daarbij een uitgebreide beschrijving van de bevindingen.

5.3.1 Beantwoording hoofd- en deelvragen

In dit hoofdstuk laat ik zien hoe ik de deelvragen en uiteindelijk de hoofdvraag zal beantwoorden. De deelvragen zullen eerst beantwoord worden.

1. **Welke metriekeken zijn er door verschillende auteurs ontwikkeld?**
Door de gevonden literatuur te bestuderen, zal het antwoord op deze deelvraag een duidelijk beeld worden geschept welke metriekeken er ontwikkeld zijn.
2. **Voor welke doeleinden zijn de metriekeken ontwikkeld?**
Tijdens het beschrijven van de eerste deelvraag is ook duidelijk voor welk doel de metriekeken ontwikkeld zijn.
3. **Wat zijn de overeenkomsten en verschillen tussen de verschillende metriekeken?**
Na het toepassen van de metriekeken op twee ER-modellen en procesmodellen, zal het makkelijker zijn om de verschillen en overeenkomsten qua werking te beschrijven. De verschillen en overeenkomsten zullen eerst beschreven worden per kwaliteitseigenschap die door Moody word aangehouden. Daarna zal er in tabelvorm en kort overzicht worden gepresenteerd.
4. **Zijn de ER metriekeken toepasbaar op een UML-model?**
Dezelfde metriekeken die op de ER-modellen en procesmodellen zijn toegepast, worden nu toegepast op het UML-model. Het toepassen van de metriekeken op het UML-model hebben een uitkomst, aan de hand van deze uitkomst word er gekeken of de metriekeken toepasbaar zijn.

5. Welke problemen kunnen optreden tijdens het toepassen van ER metrieken op UML-modellen?

Nu de uitkomsten van de metrieken bekend zijn, zal er worden gekeken welke problemen mogelijk kunnen optreden. Het antwoord op deze deelvraag is een beschrijving van mogelijke problemen.

Na het beantwoorden van alle bovenstaande deelvragen kan de hoofdvraag worden beantwoord. De hoofdvraag in de onderzoek luidt:

Kunnen de modelmetrieken die zijn ontwikkeld voor ER-modellen worden toegepast op UML-modellen?

Het antwoord op de onderzoeksvraag zal een beschrijving opleveren of modelmetrieken die ontwikkeld zijn voor ER-modellen toepasbaar zijn op UML-modellen. Daarbij wordt er terug gegrepen op de antwoorden die voortvloeien uit de deelvragen.

5.4 Onderzoeksstrategie

De onderzoekstrategie is volgens Verschuren:

Een geheel van met elkaar samenhangende beslissingen over de wijze waarop het onderzoek moet worden uitgevoerd. Bij deze uitvoering wordt voornamelijk bedoeld op het vergaren van relevant materiaal en de verwerking van dit materiaal tot antwoorden op de vragen uit de vraagstelling.

De strategie die voor dit onderzoek wordt gebruikt is die van bureauonderzoek. Dit soort onderzoek speelt zich hoofdzakelijk af, achter het bureau, bibliotheek en archieven. Bureauonderzoek wordt ook wel *literatuurstudie* genoemd, waarbij gebruik wordt gemaakt van door andere ontwikkelde kennis vastgelegd in vakliteratuur.

Het literatuuronderzoek wordt hoofdzakelijk gebruikt bij de beschrijving van de modelmetrieken dat terug komt in deelvragen één, twee en drie. Deelvragen één en twee komen terug in deel twee van deze scriptie. Deelvraag drie wordt beantwoord in deel drie.

Deel II

Dataverzameling

Hoofdstuk 6

Beschrijving modelmetrieken

In dit deel van deze scriptie zal ik twee deelvragen tegelijk beantwoorden. De deelvragen die in dit deel worden beantwoord zijn:

1. Welke metrieken zijn er door verschillende auteurs ontwikkeld?
2. Voor welke doeleinden zijn de metrieken ontwikkeld?

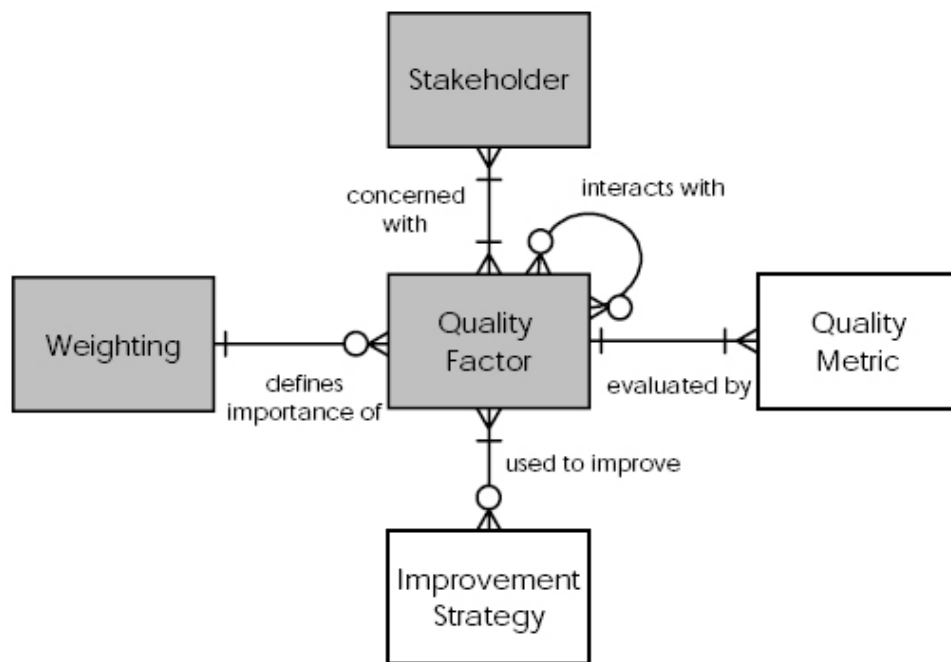
Het beantwoorden van bovenstaande deelvragen is belangrijk voor de opbouw van deze scriptie. De deelvragen dienen eerst beantwoord te worden om ervoor te zorgen dat er genoeg aanknopingspunten zijn voor het volgende deel van deze scriptie. In het volgende deel zal er een voorbeeld een ER en UML model worden gepresenteerd, om op deze voorbeelden zullen de gevonden modelmetrieken worden toegepast. Verder zullen de modelmetrieken worden genummerd en genaamd per onderzoeker. Bijvoorbeeld een metriek van Moody heet MoMetriek, voor Maes is dit MaMetriek en voor Poels is PoMetriek. Dit is gedaan zodat u makkelijk terug kunt kijken wat de metriek precies inhoudt bij de bevindingen die later aanbod zullen komen.

6.1 Metrieken ontwikkeld door Moody

Moody is geen onbekende op het gebied van conceptuele modellen en modelmetrieken. De literatuur die ik heb gebruikt bij het maken van dit hoofdstuk zijn de volgende stukken: [22] [23] [24] [25] en [26]

Moody gebruikt een framework (zie afbeelding 6.1) op de volgende bladzijde dat ontwikkeld is door hemzelf en Shanks. Dit framework is de basis van de ontwikkelde metrieken. Ik zal aan de hand van het framework de entiteitstypen uitleggen die ik vind dat ze van toepassing zijn voor deze scriptie. De overige entiteitstypen die ik niet behandel in deze scriptie, zijn naar mijn mening niet nodig om uit te leggen omdat ze weinig te maken hebben met modelmetrieken.

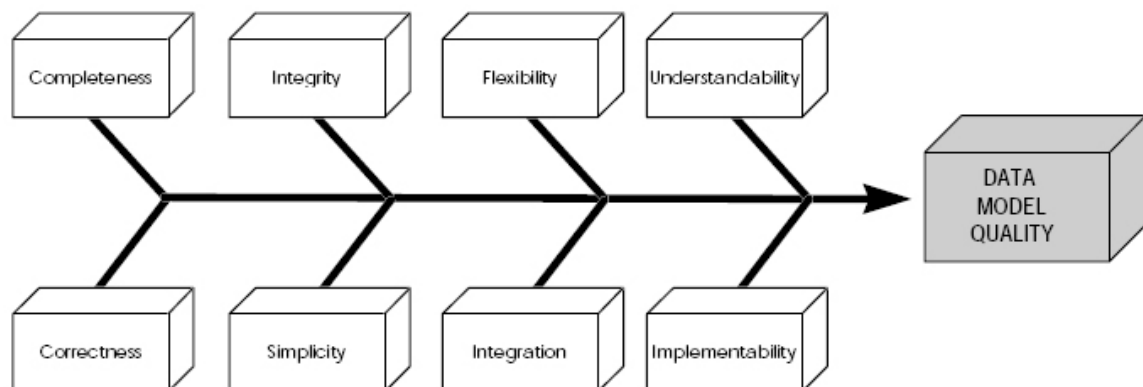
- Quality Factors: Dit zijn de eigenschappen die een rol spelen bij de kwaliteit van het te evalueren model. Ze beantwoorden de vraag, "Wat maakt een model nu goed?". Een quality factor kan een positieve of negatieve rol spelen bij andere factoren. Bijvoorbeeld als je binnen een conceptueel model de begrijpbaarheid wilt verhogen kan dit van invloed zijn op de complexiteit of correctheid. Nadat het de begrijpbaarheid is verhoogd dient het model opnieuw geëvalueerd worden om te zorgen dat er geen factoren negatief zijn beïnvloed.



Figuur 6.1: Data Model Quality Evaluation Framework [23]

- Stakeholders: Zijn mensen die zijn betrokken bij het ontwerpen van een model. De stakeholders hebben baat bij een model dat van goede kwaliteit is. Verschillende stakeholders hebben meestal interesse in verschillende eigenschappen van het model.
- Quality metrics: Zijn de metrieken waarmee je de eigenschappen van een model kunt evalueren. Verschillende metrieken kunnen worden gebruikt voor dezelfde eigenschap.

Aan de hand van het framework heeft Moody verschillende eigenschappen die je kan toekennen aan een model samengevat. Deze gedefinieerde eigenschappen zijn in afbeelding 6.2 samengevat. Aan de hand van deze afbeelding zal ik de eigenschappen opsommen en kort toelichten wat ze volgens Moody inhouden.



Figuur 6.2: Data model quality factors [23]

6.1.1 Compleetheid

Moody heeft de volgende definitie van compleetheid gebruikt. Compleetheid is volgens Moody het belangrijkste aspect van een conceptueel model. Als de eisen van de stakeholders niet terug komen in het conceptueel model dan zijn alle overige kwaliteitsfactoren overbodig. Je kunt een begrijpbaar model hebben, of een zeer simplistisch model, uiteindelijk zal het model niet compleet zijn, want er zijn dus bepaalde eisen weggelaten. Met als resultaat dat het uiteindelijke informatiesysteem voor een bepaalde groep stakeholders niet bruikbaar is. Compleetheid houdt dus in dat aan alle behoeften van stakeholders moet worden voldaan.

Hoe evalueer je nu of een conceptueel model compleet is? In principe kan je compleetheid controleren door na te gaan of elke stakeholder wordt vertegenwoordigd in het model. Dit doe je door elk element in het model te bekijken of dat overeenkomt met de eisen van een stakeholder. Het praktische probleem volgens Moody is dat er geen externe bron is met alle eisen van stakeholders. De eisen bestaan dus in de geest van een stakeholder. Om compleetheid te evalueren dien je dus uitvoerig met de stakeholders te overleggen.

Moody heeft op basis van de bovenstaande definitie van compleetheid vier metrieken ontwikkeld. De eerste vier metrieken die hieronder aanbod komen zijn dus bedoeld om de compleetheid van een conceptueel model te evalueren.

- **MoMetriek 1: Niet overeenkomende modelelementen**

Het aantal modelelementen dat niet overeenkomt met de behoeften van de stakeholders. Dit kan leiden tot onnodige ontwikkelingskosten en extra (onnodige) inspanning.

- **MoMetriek 2: Ontbrekende eisen**

Het ontbreken van eisen van stakeholders kan ervoor zorgen dat het uiteindelijke systeem niet geaccepteerd wordt. Of deze eisen zullen in een later stadium toegevoegd moeten worden wat weer extra kosten met zich meebrengt.

- **MoMetriek 3: Foutief gedefinieerde modelelementen**

Het aantal objecten of modelelementen dat overeenkomt met de behoeften van stakeholders maar onjuist gedefinieerd is. Dit kan ervoor zorgen dat de eisen in een later stadium aangepast moeten worden. Ook dit zorgt voor extra kosten en er is een mogelijkheid dat het systeem niet wordt geaccepteerd.

- **MoMetriek 4: Inconsistenties ten opzichte van het procesmodel**

Het aantal inconsistenties met het procesmodel. Een kritische taak in het controleren van de compleetheid is het conceptueel model tegen de bedrijfsprocessen te leggen die het informatiesysteem moet gaan ondersteunen. Dit zorgt ervoor dat er gecontroleerd wordt of aan alle eisen worden voldaan door het conceptueel model, ten opzichte van het procesmodel.

6.1.2 Integriteit

Integriteit wordt gedefinieerd met de mate waarin de business rules van toepassing zijn op het conceptueel model. Business rules definiëren wat wel en niet dient te gebeuren met de gegevens. Business rules zijn nodig voor het behoud van de samenhang en integriteit van gegevens die zijn opgeslagen.

Volgens de ISO [13] zijn business rules alle regels die van toepassing zijn op de

gegevens en moet worden gedocumenteerd in het conceptueel model om te zorgen dat ze consequent worden toegepast in het informatiesysteem.

Het evalueren van integriteit gebeurt feitelijk op dezelfde manier als de kwaliteitsfactor compleetheid. Ook integriteit dient in nauw samenwerkingsverband met de stakeholders geëvalueerd worden. De business rules kan je vertalen naar natuurlijke taal. Met de vertaalde regels kan je aan de hand van het conceptueel model controleren of elke regel terug te vinden is. Doordat de zakelijke gebruikers vaak geen idee hebben wat een conceptueel model precies inhoudt, zijn de regels hier een uitkomst. Want natuurlijke taal kunnen zakelijke gebruikers ook begrijpen. Zo kun je dus met de stakeholders samen kijken of het conceptueel model integer is. Deze controle is noodzakelijk voor de integriteit van het conceptueel model, want zakelijke gebruikers hebben vaak moeite met het begrijpen van bijvoorbeeld kardinaliteit.

Op basis van deze integriteit definitie heeft Moody een tweetal metrieken ontwikkeld. Die ervoor zorgen dat de modellen integer zijn. De twee metrieken voor integriteit komen hieronder aanbod.

- **MoMetriek 5: Niet afgedwongen business rules**

Het aantal business rules die niet worden afgedwongen door het conceptueel model. Als de regels niet worden vertegenwoordigd in het conceptueel model dan kan dit ernstige gevolgen hebben voor integrity en/of operationele problemen.

- **MoMetriek 6: Niet overeenkomende business rules**

Het aantal business rules die niet overeenkomen of niet nauwkeurig zijn in overeenstemming met het proces model. Een business rule kan op twee manieren foutief zijn:

- Te zwak: De business rule laat ongeldige gegevens toe.
- Te sterk: De business rule laat geen geldige gegevens toe. Hierdoor zal de gebruiker voor een workaround moeten zorgen, dit zorgt voor beperkingen.

6.1.3 Flexibiliteit

Flexibiliteit is volgens Moody gedefinieerd als het gemak waarmee het conceptueel model aangepast kan worden aan het de verandering van bedrijf. Het doel van flexibiliteit is om het conceptueel model te kunnen aanvullen en / of veranderingen door te voeren. Het conceptueel model is een belangrijke factor die bijdrage levert aan de flexibiliteit van het informatiesysteem [22]. Een gebrek aan flexibiliteit in een conceptueel model kan tot gevolg hebben dat:

- Het erg moeilijk wordt om onderhoud te plegen op een conceptueel model. Onderhoud van alle mogelijke veranderingen (modelstructuur, organisatie, enz.) kunnen tot gevolg van extra kosten ter weeg brengen.
- De snelheid waarmee een organisatie zichzelf kan aanpassen. Door starre systemen worden wijzigingen binnen een organisatie afgeremd. Om bijvoorbeeld een nieuw product op de markt te brengen dient het informatiesysteem ook mee te kunnen veranderen.

Moody vindt dat flexibiliteit de moeilijkste factor is om te beoordelen, vanwege de inherente problemen wat er kan gebeuren in de toekomst. Evaluatie van flexibiliteit vereist in kaart te kunnen brengen welke eisen in de toekomst kunnen veranderen,

hoe ze kunnen veranderen en wat voor invloed dit heeft op het conceptueel model.

Het evalueren van flexibiliteit is een manier om de impact van veranderingen tot een minimum te beperken. Rekening houdend met de kans dat het model veranderd in de toekomst en met de eventuele (toekomstige) kosten die gemaakt moeten worden. Moody heeft voor flexibiliteit drie metrieken ontwikkeld die hieronder staan beschreven. Met deze metrieken kan er worden gekeken waar het model mogelijk instabiel zou zijn met betrekking tot de eventuele veranderingen in de toekomst.

- **MoMetriek 7: Het aantal veranderende elementen**

Het aantal elementen in het model die onderworpen zijn aan veranderen in de toekomst. Dit heeft ook betrekking op de wijziging of verandering van business rules.

- **MoMetriek 8: Geschatte kosten van veranderingen**

De geschatte kosten van de veranderingen. Voor elke mogelijk verandering dient er een overzicht gemaakt te worden wat de geschatte kosten zijn als er een zich wijziging voordoet.

- **MoMetriek 9: Strategisch belang van wijzigingen**

Strategisch belang van de wijzigingen. De gevolgen van elke mogelijke wijziging dient vastgesteld te worden. Deze wijzigingen dienen ook uitgedrukt te worden in een beoordeling door stakeholders met de noodzaak snel te kunnen reageren op de wijzigingen.

6.1.4 Begrijpbaarheid

Stakeholders moeten het conceptueel model kunnen begrijpen om na te gaan of aan hun eisen zijn voldaan. De ontwikkelaars moeten ook in staat zijn om het model te kunnen begrijpen. Als de ontwikkelaars het conceptueel model niet begrijpen maken ze wellicht een heel ander informatiesysteem dan de stakeholders voor ogen hebben. Het is dus belangrijk dat ook de ontwikkelaars begrijpen wat ze moeten gaan maken.

Begrijpbaarheid is ook belangrijk voor eindgebruikers van het informatiesysteem. Als de eindgebruikers moeite hebben met het doorgronden van het conceptueel model dan zullen ze ook met het informatiesysteem moeite hebben. Empirische studies tonen aan dat in de praktijk conceptuele modellen slecht worden begrepen door gebruikers. In de meeste gevallen worden de informatiesystemen niet ontwikkeld met directe betrokkenheid van de gebruikers [12]. Hoewel modelleren een zeer effectieve techniek is voor database ontwerpen, is het veel minder effectief voor de communicatie met stakeholders [25]. Vaak interpreteren de stakeholders het conceptueel model niet of ze begrijpen de conceptuele modellen gewoon niet.

Volgens Moody kan begrijpbaarheid ook alleen maar geëvalueerd worden met nauwe samenwerking met de stakeholders. Begrijpelijkheid kan alleen worden gecontroleerd, door te controleren of elk model-element dat is opgenomen in het conceptueel model begrijpelijk is. Het praktische probleem met het conceptueel model zijn de gevolgen als een model verkeerd geïnterpreteerd wordt. Dit kan ervoor zorgen dat de stakeholders niet precies weten waarmee ze te maken hebben als er wijzigingen door worden gevoerd, resulterend in het feit dat informatiesysteem bijvoorbeeld onnodig wordt aangepast, of verkeerd wordt aangepast zodat het informatiesysteem helemaal niet meer bruikbaar is.

Voor begrijpbaarheid heeft Moody een drietal metrieken ontwikkeld. De drie metrieken zullen hieronder aanbod komen.

- **MoMetriek 10: Beoordeling door stakeholders**

Gebruikers geven een beoordeling van het model. De gebruiker beoordeelt het conceptueel model grotendeels gebaseerd op de begrippen, namen en definities die worden gebruikt, alsmede de wijze waarop het conceptueel model wordt gepresenteerd.

Een gevaar van deze statistieken zijn de zakelijke termen die in het conceptueel model aanwezig zijn. Doordat er zakelijke termen worden gekoppeld aan het conceptueel model kunnen stakeholders denken dat ze het conceptueel model hebben doorgrond. Terwijl het door de stakeholders niet begrepen is. Met als gevaar dat ze een goed cijfer geven voor een conceptueel model wat wellicht niet aan hun eisen voldoet.

- **MoMetriek 11: Interpretatie stakeholder**

De manier waarop stakeholders het conceptueel model interpreteren. Dit kan gemeten worden door business scenario's op het model los te laten. Door een stakeholders een populatie laten maken voor een business scenario kan er aan de hand van de fouten die ze maken worden gekeken hoe goed ze het model begrijpen.

Moody zelf prefereert deze metriek boven metriek "Beoordeling door stakeholders" want dit geeft een beter beeld of een stakeholder het conceptueel model begrijpt of niet. Moody beschrijft dat ook via het model van Lindland et al [17] beter gemeten kan worden.

- **MoMetriek 12: Oordeel ontwikkelaars**

Het oordeel van de ontwikkelaar is ook een manier om het conceptueel model te meten. De ontwikkelaar zal het uiteindelijke informatiesysteem gaan maken. Het is daarom belangrijk dat de ontwikkelaar precies weet wat het conceptueel model precies inhoudt, om er zo voor te zorgen dat het informatiesysteem goed word geïmplementeerd. Voor de stakeholders die direct betrokken zijn bij het ontwikkelen van het conceptueel model is alles duidelijk, voor een ontwikkelaar die het model voor het eerst ziet kan het wellicht vraagtekens opwerpen.

6.1.5 Correctheid

Correctheid beschrijft of een conceptueel voldoet aan de voorschriften van de regelgeving. Conventies voor de correctheid bevatten diagrammen, naamgeving, definitie regels en voorschriften voor de samenstelling (bijvoorbeeld elk entiteit moet een primaire sleutel bevatten in een ER diagram).

Correctheid heeft alleen maar betrekking op de vraag of de gegevens syntax en grammaticaal correct worden gebruikt. Aan de hand van correctheid kan de vraag "Is een model geldig?"beantwoord worden.

Moody vindt dat correctheid het eenvoudigst te controleren is van alle kwaliteitsfactoren. Hij geeft aan dat dit te maken heeft met de objectiviteit. Geldigheid van een model is geheel objectief. Bijvoorbeeld de naamgeving van de modelelementen kan makkelijk worden nagelopen. Tevens kan het conceptueel model in afzondering (lees zonder samenwerking met stakeholders) doorgenomen worden. Uiteindelijk zal het conceptueel model voldoen aan de regels of niet. Mocht het conceptueel niet

voldoen, dan zal dit resulteren in een lijst met gebreken. Deze lijst bestaat dan voornamelijk uit syntax en grammatica fouten die het conceptueel model bevat.

De volgende metrieken voor het evalueren van correctheid zijn voornamelijk op basis van syntactische correctheid gecreëerd. De correctheidsfouten worden verdeeld in verschillende categorieën typenfouten, om zo de fouten makkelijk te kunnen identificeren. Na het identificeren kan je de typefouten elimineren.

- **MoMetriek 13: Conventionele overtredingen**

Het aantal conventie overtredingen van een model. Deze overtredingen kunnen verder worden verdeeld onder:

- Diagram normen overtredingen (bijv. de relaties zijn niet benoemd)
- Benamingen (bijv. verkeerd gebruik van bij namen van entiteitstypen)
- Ongeldige primaire sleutels (bijv. niet unieke sleutels)
- Ongeldig gebruik van constructies (bijv. overlappende subtypes)

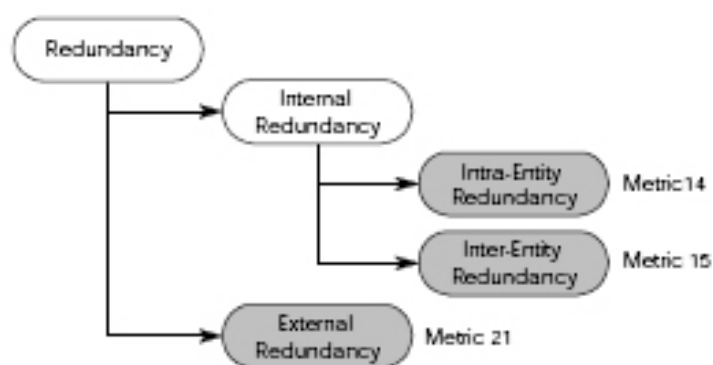
- **MoMetriek 14: Normaalvorm fouten**

Het aantal normaalvorm overtredingen. Deze overtredingen kunnen gecategoriseerd worden onder:

- Eerst normaalvorm (1NF)
- Tweede normaalvorm (2NF)
- Derde normaalvorm (3NF)
- Hogere normaalvorm (4NF +)

- **MoMetriek 15: Redundantie**

Het aantal redundante gevallen in het conceptueel model. Bijvoorbeeld als entiteitstypen elkaar overlappen. Of wanneer overbodige relaties zijn gecreëerd. Moody heeft de redundantie opgedeeld in 2 hoofdgebieden 'Internal Redundancy' en 'External Redundancy'. Dit is te zien in het onderstaande figuur 6.3.



Figuur 6.3: Moodys [23] Classification of Redundancy

Intra-entity is feitelijk de normaalvormen beschrijven in MoMetriek 14 "Normaalvorm fouten". Inter-entity is hierboven in MoMetriek 15 "Redundantie" beschreven. External redundancy word uitgelegd in MoMetriek 21 "Het aantal dubbele data elementen".

6.1.6 Eenvoud

Eenvoud betekent dat er zo min mogelijk entiteiten worden gebruikt om de eisen te verwerken in een ER-model. Eenvoudige modellen zijn volgens Moody [22] flexibeler, uiteindelijk makkelijker te implementeren en makkelijker te begrijpen. Moody heeft de keuze voor eenvoud gebaseerd op Ockham's Razor. Moody heeft dit vertaald naar het conceptueel modelleren. Als er de keuze is tussen twee modellen die hetzelfde betekenen maar met een ongelijk aantal entiteitstypen dan zal de keuze moeten gaan voor het model met de minste entiteitstypen. Want het conceptueel model met de minste entiteitstypen zal ook het makkelijkst te begrijpen zijn volgens Moody.

Het evalueren van eenvoud van een model is erg makkelijk om te doen volgens Moody [22]. Het enige wat je hoeft te doen is de entiteitstypen tellen. Men hoeft geen expert te zijn om dit te kunnen, als je kunt tellen dan zit je al goed. Moody geeft drie metrieken om de eenvoud van een conceptueel model te evalueren. De drie metrieken om de eenvoud te meten worden hieronder beschreven.

- **MoMetriek 16: Hoeveelheid entiteitstypen**

Het aantal entiteitstypen van een conceptueel model. Het aantal entiteitstypen staat in verhouding tot de complexiteit van het conceptueel model. Hoe groter het aantal entiteitstypen des te complexer het conceptueel model wordt. Volgens Symons [40] is het aantal entiteitstypen gelijk aan de grote van de te ontwikkelen applicatie.

- **MoMetriek 17: Entiteiten en hun relaties**

Het aantal entiteitstypen en de relaties tussen de entiteitstypen. Deze metriek geeft een beter beeld van de complexiteit van een conceptueel model, omdat hier niet alleen de entiteitstypen worden meegenomen maar ook hun relaties. Door ook de relaties mee te nemen wordt er een beter beeld verkregen tussen de onderlinge verbanden van entiteitstypen. Complexiteit van een conceptueel model heeft als basis het aantal entiteitstypen en de relaties tussen deze entiteitstypen.

- **MoMetriek 18: Het aantal constructen**

Het aantal constructen. Met constructen bedoelt Moody entiteitstypen (E), relaties (R) en attributen (A). Deze metriek is het specifiekst van de drie door Moody ontwikkelde metrieken voor eenvoud. Deze metriek kan berekend worden door $E+R+A$. Moody geeft zelf al aan dat deze metriek niet meer informatie geeft als MoMetriek 17 "Entiteiten en hun relaties".

6.1.7 Integratie

Moody definieert integratie als een consistentie check met de organisatie data. In de praktijk komt het vaak voor dat informatiesystemen los van elkaar worden ontwikkeld. Doordat de informatiesystemen los van elkaar zijn ontwikkeld, kan het zijn dat dezelfde bedrijfsdata in twee systemen wordt ingevoerd. Met als resultaat dat de organisatiedata dubbel aanwezig is en hierdoor kan het problemen geven voor management rapporten. Bijvoorbeeld welke informatie uit welk informatiesysteem moet je gaan gebruiken de voor management rapporten?

Het evalueren van integratie is volgens Moody een vergelijking tussen het conceptueel model, met het organisatiemodel. Dit resulteert zich in een opsomming van conflicten. Om de integratie te evalueren heeft Moody vier verschillende metrieken ontwikkeld die hieronder behandeld worden:

- **MoMetriek 19: Aantal conflicten**

Het aantal conflicten met het organisatiemodel. Deze conflicten kunnen onderverdeeld worden in verschillende categorieën.

- Entiteit conflicten: het aantal entiteiten waarvan de definities inconsistent zijn met de definities in het organisatiemodel.
- Data elementconflicten: het aantal attributen die een andere definitie of domein hebben vergeleken met het organisatiemodel.
- Benamingsconflicten: het aantal entiteitstypen of attributen met dezelfde betekenis maar met een verschillende naam vergeleken met het organisatiemodel.

- **MoMetriek 20: Het aantal data conflicten**

Het aantal data conflicten met het bestaande systeem. Deze data conflicten kunnen onderverdeeld worden in verschillende categorieën:

- Het aantal data elementen wiens definities in conflict zijn met het bestaande systeem. Volgens Moody leiden inconsistente definities tot interface problemen.
- Het aantal sleutel conflicten met het bestaande systeem. Sleutel conflicten ontstaan als er verschillende verwijzingen bestaan naar bijvoorbeeld een klant vanuit verschillende systemen. Dit leidt tot fragmentatie van data en tot het onvermogen om gegevens met elkaar te verbinden.
- Het aantal benamingsconflicten met andere systemen. Benamingsconflicten kunnen tot gevolg hebben dat het onderhoud langer duurt of dat de gegevens verkeerd geïnterpreteerd worden. Omdat er bijvoorbeeld langer gezocht moet worden naar een bepaalde entiteit in het informatiesysteem terwijl deze een andere naam heeft.

- **MoMetriek 21: Het aantal dubbele data elementen**

Het aantal dubbele data elementen in verhouding tot het conceptueel model en het bestaande systeem. Dit wordt *externe redundantie* genoemd. Externe redundantie is de redundatie die niet binnen het model aanwezig is (zie MoMetriek 14 "Normaalkvorm fouten" en MoMetriek 15 "Redundantie" of afbeelding 6.3).

- **MoMetriek 22: Waardering van vertegenwoordigers van andere systemen**

Waardering van het conceptueel model door vertegenwoordigers van andere informatiesystemen. Door middel van deze metriek wilt Moody bekijken of de vertegenwoordigers een waardering kunnen geven of hun systeem kan integreren met het conceptueel model.

6.1.8 Implementeerbaarheid

Implementeerbaarheid is door Moody gedefinieerd als de mogelijkheid waarop het systeem ontwikkeld kan worden. Rekening houdend met de tijd, het budget en de technologische constraints. Terwijl het belangrijk is dat het conceptueel model geen aannemens mag bevatten over de manier hoe het geïmplementeerd moet worden.

Volgens Moody gebeurt het evalueren van implementeerbaarheid door een applicatie ontwikkelaar. De reden dat het evalueren door een applicatie ontwikkelaar wordt gedaan, is dat een applicatie ontwikkelaar het beste kan zien of het een realistisch en haalbaar project is. Doordat de applicatie ontwikkelaar het conceptueel model

moet beoordelen, krijgt de applicatie ontwikkelaar al een idee hoe het uiteindelijke informatiesysteem eruit moet komen te zien. Om de implementeerbaarheid te kunnen evalueren heeft Moody drie metrieken ontwikkeld, die hieronder aanbod komen.

- **MoMetriek 23: Schatting technisch risico**

Schatting technisch risico is de waarschijnlijkheid dat het informatiesysteem aan de requirementseisen kan voldoen. Gebaseerd op het conceptueel model.

- **MoMetriek 24: Schatting tijdsduur**

De schatting van tijd om het informatiesysteem te bouwen.

- **MoMetriek 25: Schatting ontwikkelkosten**

De schatting van de kosten om het informatiesysteem te bouwen. Mocht het informatiesysteem teveel kosten dan kan de reikwijdte van het systeem verminderd worden, het budget vergroot worden of het conceptueel model moet eenvoudiger.

6.2 Metrieken door Ann Maes

Maes [21] beschrijft dat er geen evaluatie is op basis van gebruikers voor conceptuele modellen. Maes heeft daarom dit onderzoek gedaan naar conceptual modelling scripts vanuit het standpunt van gebruikers. Een conceptual modelling script is de taal waarmee het conceptueel model is opgezet. Om ervoor te zorgen dat alle vereiste kennis over het probleemdomein (of UoD) met daarbij de functies die uitgevoerd moeten worden allemaal het conceptueel model terug komen. Hiervoor gebruikt Maes het conceptual modelling script. Het script beschrijft dus de eisen en de manier van interpreteren van systeemgebruikers, systeemanalisten en systeemontwikkelaars over een bepaald probleem domein (of UoD). Uiteindelijk zorgt deze informatie voor de input van het te ontwikkelen systeem. Maes denkt dat het evalueren en gebruiken van scripts leidt tot een vroegtijdige detectie van requirementsfouten en tot de verbetering hiervan. Resultierend in het feit dat de kwaliteit van het te ontwikkelen systeem groter wordt.

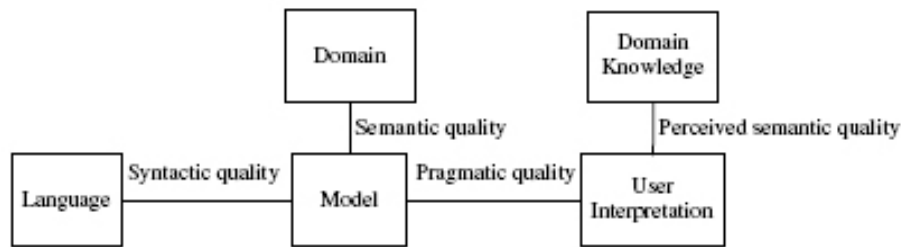
Het ontwikkelen van informatiesystemen gebeurt meestal voor een klant. Maar word er gekeken met de klant naar het conceptueel model? Maes zegt dat er tot nog toe geen onderzoek is gedaan naar hoe tevreden een klant is met een conceptueel model. Maes heeft dus metrieken ontwikkeld om de kwaliteit van het conceptueel model te meten in samenwerking met de klant. Dus vanuit het standpunt van de uiteindelijke gebruiker van het informatiesysteem.

Maes heeft verschillende frameworks onderzocht maar het framework dat initieel door Lindland et al. [17] is ontwikkeld en later uitgebreid door Krogstie [15] was de enige optie voor Maes, want het framework was als enige uitvoerig gevalideerd. In afbeelding 6.4 (zie volgende baldzijde) zijn de volgende entiteiten te vinden:

- Language: alle uitdrukkingen die volgens de syntax en volgens de grammatica toegestaan zijn.
- Domain: alle mogelijke verklaringen die relevant en correct zijn voor de beschrijving van het domein (of UoD).
- Model: een schematische beschrijving van de werkelijk.
- User Interpretation: alle uitspraken die stakeholders denken dat het conceptueel model bevat.
- Domain Knowledge: alle mogelijke beschrijvingen die het probleem domein omschrijft.

De bovenstaande entiteitstypen worden met elkaar gekoppeld door de volgende kwaliteitseigenschappen:

- Syntactic quality: beschrijving of het conceptueel model aan de grammatica voldoet.
- Semantic quality: beschrijft hoe goed het conceptueel model voldoet aan de realiteit.
- Pragmatic quality: beschrijft hoe goed het conceptueel model wordt begrepen door gebruikers.
- Perceived semantic quality: beschrijft wat gebruikers denken wat in het conceptueel model thuishoort en wat het conceptueel model precies moet inhouden.



Figuur 6.4: Krogstie [15] extended framework for quality of conceptual modelling scripts

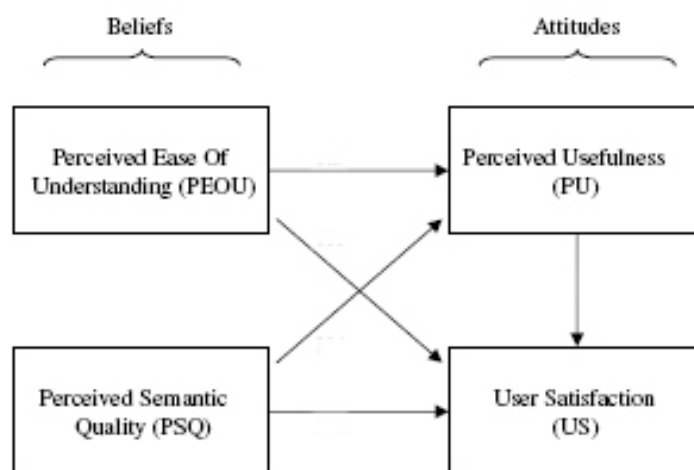
Syntactic Quality:

Volgens Maes is deze kwaliteitseigenschap van een conceptual modelling script het makkelijkst te evalueren. Simpelweg door je gemakkelijk kunt kijken of een script voldoet aan bepaalde grammatica conventies. Voor het evalueren van syntactic quality zijn al enige programma's / tools die dit objectief voor je meten. Hier heeft Maes ook geen metriecken voor ontwikkeld.

Semantic Quality:

Semantic quality kan je botweg omschrijven als de beschrijving van de realiteit. Maar enige nuance is wel nodig, want door welke bril kijk je naar de realiteit? Dit houdt in dat iedereen op een andere manier naar de realiteit kijkt. Semantic Quality is daarom moeilijk te evalueren, tenzij je in iemands hoofd kan kijken. Vandaar dat Krogstie het Lindland model heeft uitgebreid met een extra manier om Semantic Quality te evalueren.

De overige twee kwaliteitseigenschappen zullen aanbod komen aan de hand van de onderstaande afbeelding. Waarbij Pragmatic quality word verbonden met Perceived Ease of Understanding (PEOU) en Perceived Semantic Quality (PSQ) zal in het gelijknamige kopje worden behandeld.



Figuur 6.5: User Evaluation Based Quality Model for Conceptual Modelling [21]

Afbeelding 6.5 geeft de onderlinge relaties aan tussen de kwaliteitseigenschappen

die Maes heeft gebruikt voor de ontwikkeling van haar metriecken. Zoals u kunt zien gaan de pijlen naar de kwaliteiteigenschappen Perceived Usefulness (PU) en User Satisfaction (US). Dit houdt in de evaluatie vooral van de gebruiker centraal staan.

Aan de hand van de beschrijving die Moody [23] in zijn onderzoek gebruikt om zijn metriecken onder een bepaalde definitie te plaatsen, zal ik de metriecken van Maes op basis van Moodys definitie benoemen. Dit wordt gedaan om vast een stap te zetten naar de verschillen en overeenkomsten tussen de ontwikkelde metriecken. Deze verschillen en overeenkomsten zullen in hoofdstuk 8 aanbod komen.

6.2.1 Perceived Ease of Understanding

Perceived ease of understanding is afhankelijk van de systeemkwaliteit, met name betrekking tot het gebruik van het systeem. Als het systeem gemakkelijk is om te gebruiken geldt dit volgens Maes ook voor conceptual modelling scripts. Wanneer gebruikers gebruik maken van een script dan kunnen de gebruikers achteraf evalueren of het script de juiste informatie bevat. Gebruikers kunnen ook evalueren hoe gemakkelijk het is om de informatie te lezen en/of ze deze informatie ook goed is te interpreteren. Perceived ease of understanding is een indirecte manier om pragmatic quality te evalueren.

Om de Perceived Ease of Understanding te kunnen evalueren heeft Maes de volgende metriecken bedacht:

- **MaMetriek 1: Makkelijk te begrijpen (Begrijpbaarheid)**
Het is makkelijk te begrijpen wat het conceptueel model moest modelleren.
- **MaMetriek 2: Frustratie met gebruik (Begrijpbaarheid)**
Het gebruik van het conceptueel model is vaak frustrerend.
- **MaMetriek 3: Makkelijk in gebruik (Eenvoud)**
In het algemeen is het conceptueel model makkelijk te gebruiken.
- **MaMetriek 4: Lezen is makkelijk (Begrijpbaarheid)**
Het leren lezen van het conceptueel model is makkelijk.

6.2.2 Perceived Usefulness

DeLone en McLean zeggen dat de Perceived usefulness een van de belangrijkste factoren is voor een succes van een informatiesysteem. Volgens Maes heeft Seddon gelijk [36] als het gaat over het meten van usefulness, want het is niet de manier van gebruiken van het informatiesysteem. Want het systeem kan wel gebruikt worden maar mogelijk op de verkeerde manier dan oorspronkelijk bedoeld is. Met andere woorden, het werk dat een gebruiker moet uitvoeren, moet dus worden verrijkt door het gebruik van het informatiesysteem. Maes denkt dat dit ook geldt voor conceptual modelling scripts. Als een gebruiker 'communiceert' met het script dan kan de gebruiker als snel oordelen of het script een verrijking is voor zijn functie of niet. Dus gebruikers zullen de bruikbaarheid van het script beoordelen.

Om de Perceived Usefulness te kunnen evalueren heeft Maes de volgende metriecken bedacht:

- **MaMetriek 5: Verbetering ten opzichte van tekst (Implementeerbaarheid)**
Het conceptueel model vind ik een verbetering ten opzichte van de tekstuele beschrijving van het bedrijfsproces.

- **MaMetriek 6: Helpt bij het begrijpen (Begrijpbaarheid)**
Het conceptueel hielp mij bij het begrijpen van het gemodelleerde proces.
- **MaMetriek 7: Verbetert de prestatie om te begrijpen (Begrijpbaarheid)**
Het conceptueel model verbetert de prestatie om het gemodelleerde proces beter te begrijpen.

6.2.3 Perceived Semantic Quality

Informatiekwaliteit refereert naar de kwaliteit van informatie die het informatiesysteem produceert. Dit wordt gedaan aan de hand van nauwkeurigheid, relevantie, compleetheid en precisie van de informatie die geproduceerd wordt. Volgens DeLone en McLean [6] is dit de manier om semantic quality te evalueren. Maes zegt dat dit het zelfde geldt voor conceptual modelling scripts. Dus de informatiekwaliteit wat hierboven is beschreven correspondeert volgens Maes met Perceived Semantic Quality. Want de gebruikers kijken aan de hand van hun taken welke informatie ze nodig hebben en welke informatie ze voor handen krijgen vanuit het informatiesysteem. Met als resultaat dat gebruiker kan kijken of de informatie die ze krijgen van het informatiesysteem ook past bij de realiteit.

Om de Perceived Semantic Quality te kunnen evalueren heeft Maes de volgende metrieken bedacht:

- **MaMetriek 8: Bedrijfsproces komt terug (Correctheid)**
Het bedrijfsproces is correct weergegeven in het conceptueel model.
- **MaMetriek 9: Realistische spiegeling van het bedrijfsproces (Compleetheid)**
Het conceptueel model is een realistische spiegeling van het bedrijfsproces.
- **MaMetriek 10: Tegenstrijdige elementen (Integriteit)**
Het conceptueel model bevat tegenstrijdige elementen.
- **MaMetriek 11: Zijn alle elementen relevant (Compleetheid)**
Alle elementen in het conceptueel model zijn relevant voor de representatie van het bedrijfsproces.
- **MaMetriek 12: Het bedrijfsproces past in conceptueel model (Correctheid)**
Het bedrijfsproces is in zijn totaliteit gepresenteerd in het conceptueel model.

6.2.4 User Satisfaction

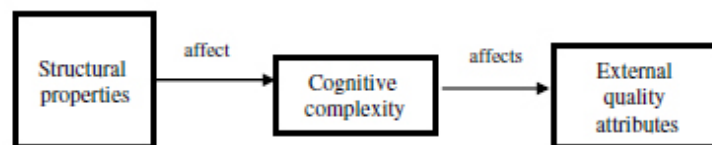
User satisfaction is een van de belangrijkste en de meeste gebruikte succes factor om informatiesystemen te evalueren. Dit komt omdat precies kunt meten of het informatiesysteem voldoet aan de eisen van de stakeholders. Seddon [36] beschrijft dat de user satisfaction een goed beeld geeft van de gehele tevredenheid. Maes gebruikt de definitie die Seddon heeft beschreven in verhouding tot het conceptual modelling script.

Om de User satisfaction te kunnen evalueren heeft Maes de volgende metrieken bedacht:

- **MaMetriek 13: Informatiebehoeften (Compleetheid)**
Het conceptueel model voldoet aan de informatiebehoeften.
- **MaMetriek 14: Niet efficiënt verstrekken van informatie (Intergriteit)**
Het conceptueel model was niet efficiënt in het verstrekken van de informatie die ik nodig had.
- **MaMetriek 15: Efficiënt verstrekken van informatie (Integriteit)**
Het conceptueel model is efficiënt in het verstrekken van de informatie die ik nodig heb.
- **MaMetriek 16: Tevreden over informatie verstrekking (Correctheid)**
Over het algemeen ben ik tevreden van de informatie verstrekking van het conceptueel model.

6.3 Metrieken door Geert Poels

Alle metrieken die Poels [32] heeft voorgesteld zijn eigenlijk gebaseerd op de onderhoudbaarheid van ER modellen. Poels beschrijft dat onderhoudbaarheid een externe kwaliteitsfactor is. Dit houdt in dat deze kwaliteitsfactor alleen geëvalueerd kan worden door de gebruikersomschrijving van het ER diagram. Hierdoor kan er volgens Poels pas aan het einde van het ontwikkelproces geëvalueerd worden. Terwijl het wenselijk is dat tijdens het ontwikkelen al geëvalueerd kan worden, om zo mogelijke fouten te voorkomen.



Figuur 6.6: Relationship between structural properties, cognitive complexity, and external quality attributes [3]

Poels heeft het Briand [3] model (zie afbeelding 6.6) overgenomen en op basis hiervan metrieken ontwikkeld voor ER diagrammen. Briand heeft een onderzoek gedaan naar objectgeoriënteerde software waarbij naar voren is gekomen dat de complexiteit vooral afhangt van het aantal relaties dat een software product kan hebben. Poels denkt dat dit hetzelfde geldt voor ER diagrammen. Complexe software of modellen (Structural properties) beïnvloedt de cognitieve complexiteit (Cognitive complexity). Doordat een model veel relaties kan hebben zorgt dit voor een hoge cognitieve complexiteit, waardoor het moeilijk is om het ER model te doorgronden. Met een hoge cognitieve complexiteit kan dit de attributen van externe kwaliteit (External quality attributes) beïnvloeden. Bijvoorbeeld door een complex model met veel onderlinge relaties kan de onderhoudbaarheid in het geding komen, waardoor het voor mensen moeilijk wordt om het ER model te onderhouden.

De metrieken maken gebruik van bepaalde constructen die Poels in tabel 6.1 (op bladzijde 43) heeft gedefinieerd. Alle metrieken zijn op basis van tellen hoeveel entiteiten aan deze metriek voldoen. De metrieken die volgen zullen in natuurlijke taal zijn opgesteld, dit wil zeggen dat ze makkelijker te begrijpen zijn voor mensen die niet dagelijks met metrieken werken.

- **PoMetriek 1: Het aantal entiteiten (Compleetheid)**

Het aantal entiteiten dat gedefinieerd is binnen een ER diagram waarbij zowel zwakke als sterke entiteiten worden opgenomen.

- **PoMetriek 2: Het aantal attributen (Eenvoud)**

Het aantal attributen is gedefinieerd als het totaal aantal attributen binnen het ER diagram. Dit geldt zowel voor entiteit attributen als relatie attributen.

- **PoMetriek 3: Het aantal afgeleide attributen (Eenvoud)**

Het aantal attributen dat een afgeleide van de afgeleide attributen in een ER diagram. Het aantal afgeleide attributen is altijd minder dan de PoMetriek 2 genaamd "Het aantal attributen".

- **PoMetriek 4: Het aantal samengestelde attributen (Eenvoud)**
Het aantal attributen dat samengesteld van alle samengestelde attributen in een ER diagram. Het aantal samengestelde attributen is altijd minder of gelijk aan de PoMetriek 2 genaamd "Het aantal attributen".
- **PoMetriek 5: Het aantal meerwaardige attributen (Eenvoud)**
Het aantal attributen dat meer waarden kan bevatten een ER diagram. Het aantal meerwaardige attributen is altijd minder of gelijk aan de PoMetriek 2 genaamd "Het aantal attributen".
- **PoMetriek 6: Het aantal relaties (Begrijpbaarheid)**
Het aantal relaties dat aanwezig is binnen een ER diagram.
- **PoMetriek 7: Het aantal M:N relaties (Begrijpbaarheid)**
Het aantal M:N relaties die aanwezig zijn binnen een ER diagram. Dit aantal is altijd minder of gelijk aan de PoMetriek 6 genaamd "Het aantal relaties".
- **PoMetriek 8: Het aantal 1:N relaties (Begrijpbaarheid)**
Het aantal 1:N en 1:1 relaties die aanwezig zijn binnen een ER diagram. Dit aantal is altijd minder of gelijk aan de PoMetriek 6 genaamd "Het aantal relaties".
- **PoMetriek 9: Het aantal N-Ary Relaties (Begrijpbaarheid)**
Het aantal N-Ary relaties die aanwezig zijn binnen een ER diagram. Dit aantal is altijd minder of gelijk aan de PoMetriek 6 genaamd "Het aantal relaties".
- **PoMetriek 10: Het aantal binaire relaties (Eenvoud)**
Het aantal binaire relaties die aanwezig zijn binnen een ER diagram. Dit aantal is altijd minder of gelijk aan de PoMetriek 6 genaamd "Het aantal relaties".
- **PoMetriek 11: Het aantal reflexieve relaties (Eenvoud)**
Het aantal reflexieve relaties die aanwezig zijn binnen een ER diagram. Dit aantal is altijd minder of gelijk aan de PoMetriek 6 genaamd "Het aantal relaties".
- **PoMetriek 12: Het aantal afgeleide entiteitstypen (Compleetheid)**
Het aantal afgeleide entiteitstypen die aanwezig zijn binnen een ER diagram. Elke relatie tussen een super-type/sub-type dient opgenomen te worden in deze metriek. Deze metriek heeft betrekking op de definitie generalisatie en specialisatie [8].

De genoemde metrieken door Poels ontwikkeld, hebben ook een formele definitie. Deze definitie is gebaseerd op de DISTANCE methode [33]. Het basis idee van de DISTANCE methode is om eigenschappen te definiëren voor objecten die als referentiepunten dienen van andere objecten. Hoe groter de afstand, hoe groter de mate waarin de objecten worden gekenmerkt door de eigenschappen. Door deze definitie kunnen deze objecteigenschappen worden gemeten door metrieken. Voor de metrieken die Poels heeft ontwikkeld is het voornamelijk de kardinaliteit die gemeten wordt. Bijvoorbeeld hoe groter het aantal entiteitstypen in een ER diagram ten opzichte van een lege verzameling entiteitstypen is het model in verhouding complexer.

Tabel 6.1: ER Modelling construct [31]

ER Modelling Construct		Remarks	
Entities Attributes	Strong Simple Non-driven Single-valued	Weak Composite Derived Multi-valued	common relationships, sometimes also called associations, are the ones considered in Chen's original ER model; IS-A relationships are generalization/specialization relationships as considered in most extensions of the ER model [8].
	Relationships common	IS_A	
	reflective	Binary	n-ary
	1:1	1:N	M:N
Kinds of common relationships according to the number of entity types involved. Kinds of common reflexive or binary relationships according to the connectivities (maximum multiplicities) involved.			

Deel III

Casus

Hoofdstuk 7

Inleiding

U heeft in de vorige pagina's uitgebreid kunnen lezen over de ontwikkelde modelmetrieke en hoe deze werken volgens de onderzoekers. In dit deel van de scriptie worden de genoemde modelmetrieke toegepast op twee eigen modellen. De twee modellen die u zo dadelijk krijgt te zien, zijn gebaseerd op een casus. De twee modellen zullen bestaan uit een ER diagram en een procesmodel. Later in dit deel van de scriptie zal er een UML-model geïntroduceerd worden. De reden dat hetzelfde model omgezet is naar UML is om te kijken of de ontwikkelde ER modelmetrieke ook toepasbaar zijn op het UML model. Met de korte inleiding die nu volgt word er geprobeerd om een beeld van de reisorganisatie te scheppen.

7.1 De reisorganisatie

Elke stakeholder binnen een project heeft bepaalde eisen. Welke eisen dit zijn ligt aan de rol die je als stakeholder speelt in het project. Aan de hand van deze eisen word er een conceptueel model gemaakt. De eisen van de stakeholders dienen terug te komen in het conceptueel model. Uiteindelijk zal dus het informatiesysteem alle eisen omvatten die de stakeholders hebben beschreven. Kortom de basis van een conceptueel model begint bij het opstellen van de eisen, die je achteraf kunt controleren aan de hand van het conceptueel model [23].

Als reiziger ben je een grote stakeholder van de reisorganisatie. Het is belangrijk dat het informatiesysteem van de reisorganisatie inspeelt op de eisen van de reiziger. De onderstaande eisen van de belanghebbenden hebben betrekking op een fictief reisbureau. De eerste eisen die aanbod komen zijn die van de reiziger een van de belangrijkste redenen dat een reisbureau kan bestaan. De eisen zijn persoonlijk van aard, want het zijn de eisen die ik zelf aan een reisbureau stel. Deze eisen zijn gebaseerd op persoonlijke ervaring tijdens het boeken van reizen.

De reden dat een vliegmaatschappij en touroperator ook een rol als belanghebbende spelen, is omdat een reisbureau de reis dient te reserveren en te verifiëren bij de touroperator en/of vliegmaatschappij voor vliegtickets.

Eisen reiziger:

- Beschikbare reizen aan de hand van vertrek- en aankomstdatum kunnen zien;
- Reisverzekering is mogelijk om tijdens het boeken af te sluiten;
- Keuze uit appartement of hotelkamer;
- Vervoer is op de plaats van bestemming te huur op via de touroperator;
- Bij individuele reis dien je ook een auto te kunnen huren;
- Vlieg- of autovakantie is mogelijk;

Eisen reisbureau:

- Aanbieden van verschillende reizen;
- Reis bij een touroperator te boeken;
- Tickets zijn beschikbaar voor geboekte periode;
- Vervoer is op de plaats van bestemming te huur op via de touroperator;
- Bij individuele reis dien je ook een auto te kunnen huren;
- Vlieg- of autovakantie is mogelijk;

Eisen vliegmaatschappij:

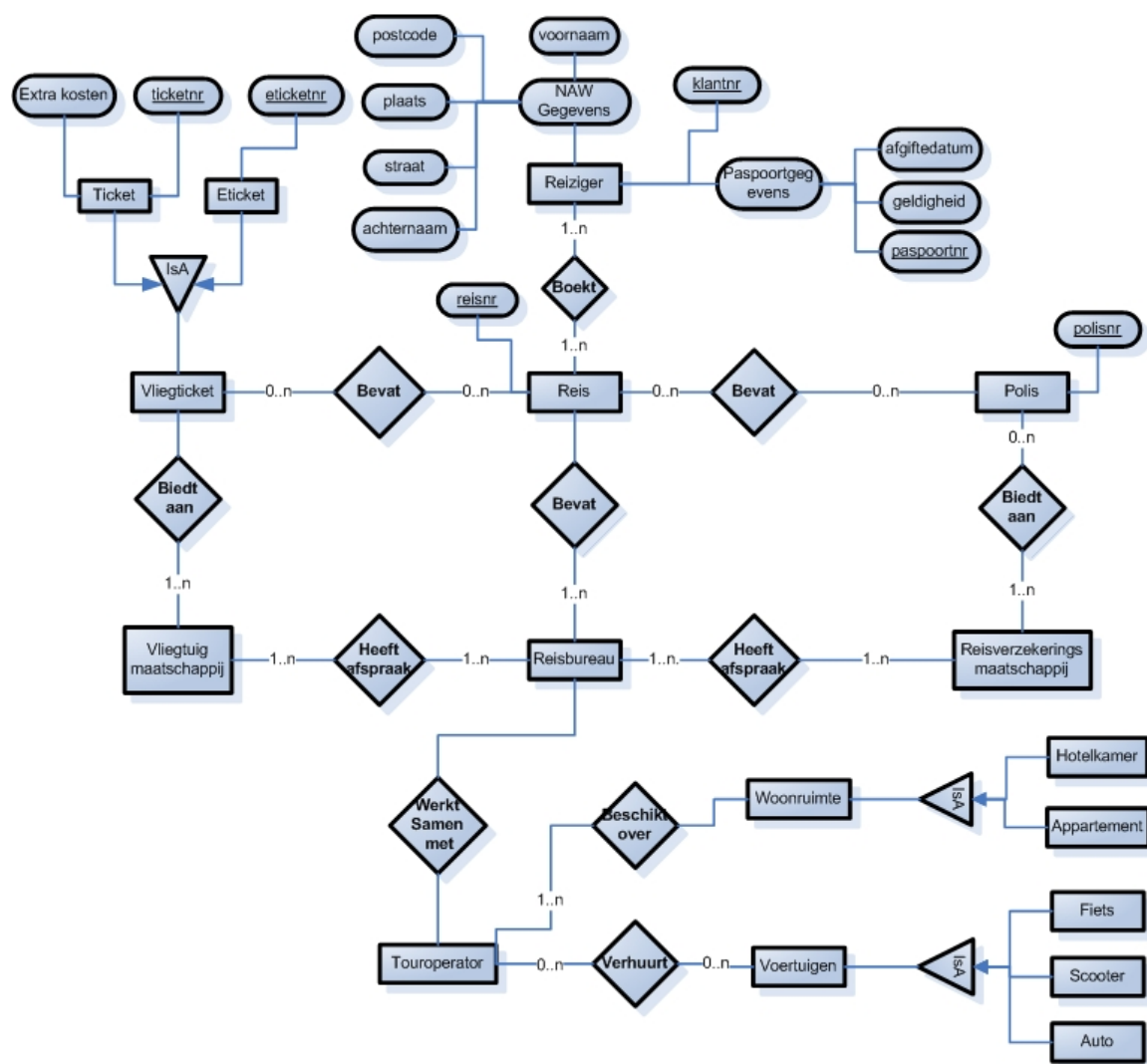
- Aanbieden van tickets via een bepaald reisbureau;

Eisen touroperator:

- Via een reisbureau woonruimte aanbieden;
- Het aan kunnen bieden van te huren vervoer;

Nu we de eisen op een rijtje hebben staan kan het conceptueel model geïntroduceerd worden. Het ER model (zie afbeelding 7.1) is gebaseerd op de bovenstaande eisen. Om de werking van de metrieken goed te kunnen evalueren heeft Marc Pieterse voor mij aan de hand van de bovenstaande eisen ook een ER model ontwikkeld (zie appendix B). Doordat er nu twee ER modellen zijn gemaakt is het makkelijker om deze uitkomsten later met elkaar te vergelijken. Dit is handig omdat er zo een goed beeld ontstaat hoe de metrieken precies werken.

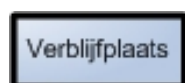
De metrieken die per doel zijn beschreven zullen ook als doel toegepast worden op het conceptueel model van het reisbureau. Dit houdt in dat alle metrieken die een bepaald doel hebben allemaal onafhankelijk van auteur onder dit doel worden geclusterd. Zodat in een later stadium makkelijker de overeenkomsten en verschillen per doel per metriek te identificeren zijn. Ook worden de onderlinge doelen vergeleken op toepasbaarheid en op relevantie. De volgorde van doelen die wordt aangehouden is op basis van de beschreven metrieken van Moody. Compleetheid zal als eerst aanbod komen, dan integriteit, flexibiliteit, etc.



Figuur 7.1: Entity Relationship model van het reisbureau

Het ER model

We beginnen eerst met een korte uitleg hoe het ER model precies is opgebouwd en wat alle modellementen betekenen. Het ER model is opgebouwd uit entiteitstypen (zie afbeelding 7.2), dit zijn de rechthoekige vakken die je kunt terug vinden in het model. Deze entiteitentypen kunnen gebruikt worden in een database, dan representeren ze de tabellen die nodig zijn om alle informatie op te slaan. Tussen de entiteitstypen bevinden zich relaties, deze relaties zorgen voor de connecties tussen de tabellen een database. De tabellen worden op basis van Primary keys (zie afbeelding 7.3) verbonden.



Figuur 7.2: Een entiteitstype



Figuur 7.3: Een attribuut als Primary Key

De relatie heeft een naam, die kan je terug vinden in het ER model als wybertje met daarin een werkwoord. Er is ook nog een speciaal soort relatie in het ER model te vinden. Namelijk het driehoekje (zie afbeelding 7.4) met de *IsA* beschrijving erin. Deze IsA is een generalisatie symbool. Dit houdt dat entiteitstypen geschaard kunnen worden onder een verzamelnaam. Wat bijvoorbeeld gedaan is bij voertuigen. Het entiteitstype voertuig is een verzamelnaam voor fiets, auto en scooter. Deze laatste drie entiteitstypen zijn specialisatie entiteitstypen.



Figuur 7.4: Een IsA Relatie

De reden dat erop de relaties verschillende getallen en letters staan, is om de kardinaliteit aan te geven. Kardinaliteit (zie afbeelding 7.5) houdt in hoe vaak een relatie kan voorkomen. Om een voorbeeld te geven van een vliegreis. Daar kan de reis bestaan uit één of meerdere (1..N) tickets. In het ER model kunt u zien dat een reis nul of meerdere tickets kan bevatten. Dit komt doordat het mogelijk moet zijn dat er ook een autovakantie geboekt kan worden, daar heb je natuurlijk geen vliegtickets voor nodig.



Figuur 7.5: Een 1..N Relatie

Om terug te komen op de Primary key. Deze kun je terug vinden in het ER model als een attribuut die onderstreept is. Een attribuut (zie afbeelding 7.3) zijn de ovals die rondom de entiteitstypen zijn te vinden. De attributen representeren de velden in een database. De velden in een database bevatten de informatie die nodig is om een bepaald proces af te wikkelen. Het is daarom belangrijk dat er van te voren goed wordt nagedacht welke informatie nodig is en waar, zodat deze opgenomen kunnen worden in het ER model.

Het procesmodel

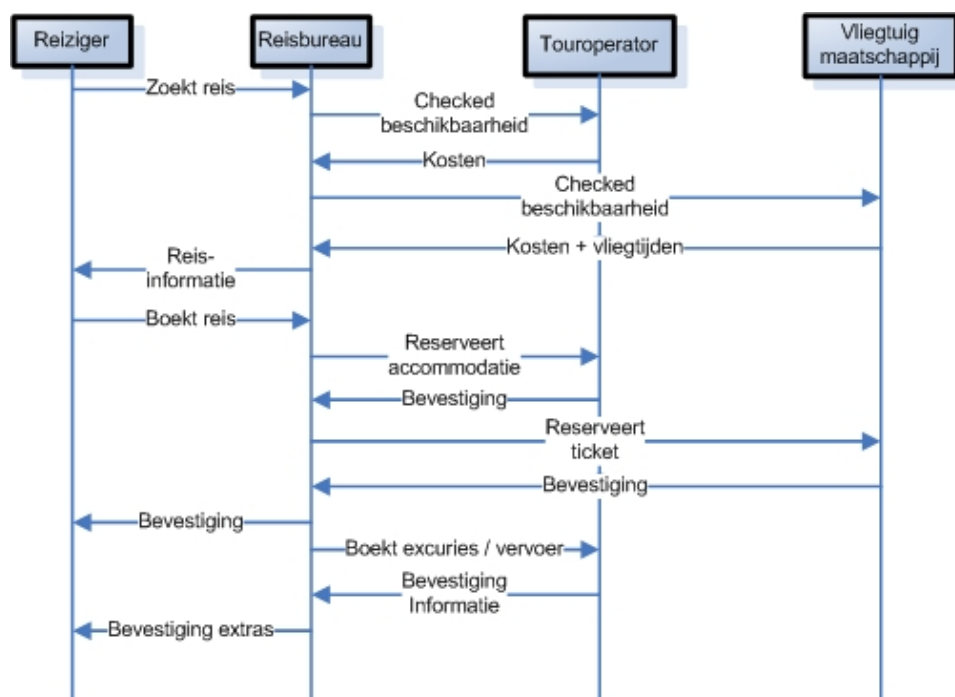
Feitelijk is het procesmodel niet anders dan de weergave van alle activiteiten van een proces in modelvorm. Ook hier zijn de entiteitentypen terug te vinden alleen staan deze entiteitstypen bovenaan. Tussen deze entiteitstypen *stroomt* er informatie. De lijnen onder de entiteitstypen representeren de manier van ontvangst en verzending van informatie.

Dit procesmodel is gebaseerd op het boeken van een reis. Je leest het procesmodel van boven naar beneden en je volgt alle pijlen (zie afbeelding 7.6). De pijlen representeren de stroom van informatie tussen de entiteitstypen. Ik kom bijvoorbeeld binnen bij een reisbureau en ik zoek een reis (eerste pijl). Dit is informatie voor het reisbureau die gaat dus zoeken in het aanbod van reizen bij de touroperators en mogelijk bij de vliegtuigmaatschappijen voor een ticket. Het over een weer van pijlen is eigenlijk de informatie die beide nodig hebben om het proces 'het boeken van een reis' af te ronden. De beschrijving op de pijlen is de informatie die men elkaar doorgeeft.



Figuur 7.6: Een informatiepijl

Het procesmodel wordt altijd opgesteld aan de hand van het bestaande proces. Mensen kunnen precies vertellen welke informatie wanneer nodig is en wat er mee gedaan moet worden. Het is belangrijk dat een procesmodel word opgesteld als er gekeken moet worden naar welke informatie gebruikt wordt.



Figuur 7.7: Procesmodel van het boeken van een reis

Hoofdstuk 8

Bevindingen na toepassing modelmetriecken

U heeft reeds kunnen lezen welke modelmetriecken zijn ontwikkeld door de verschillende onderzoekers. In dit hoofdstuk zullen de bevindingen worden beschreven na het toepassen van de modelmetriecken op de beschreven casus. Aan het eind van dit deel zal er een tabel word weergegeven waarin de overeenkomsten en verschillen overzichtelijk worden weergegeven.

Moody [23] is een van de belangrijkste grondleggers geweest van modelmetriecken. Moody heeft voor zijn modelmetriecken verschillende definities beschreven in zijn onderzoek. De metriecken worden geplaatst onder Moodys definities, om zo onderlinge verbanden tussen de ontwikkelde metriecken te kunnen ontdekken.

De deelvraag die in dit deel wordt beantwoord is:

1. Wat zijn de overeenkomsten en verschillen tussen de verschillende metriecken?

Ook hier zullen de metriecken zijn net zoals in de beschrijving van de metriecken genummerd en genaamd worden per onderzoeker. Door de benaming MoMetriek, MaMetriek en PoMetriek kan er makkelijk worden terug gekeken in de beschrijving (zie hoofdstuk 6) van een modelmetriek.

8.1 Compleetheid

MoMetriek 1: Niet overeenkomende modelelementen

Deze metriek is redelijk langzaam in gebruik, want je moet goed op een rijtje hebben wat nu precies de behoeften zijn. Het kan voorkomen dat een stakeholder afwezig is of niet met de volle aandacht aanwezig is bij het opstellen van de requirements. Hoe kan je dan controleren aan de hand van modelelementen of je dan alle eisen / behoeften hebt?

MoMetriek 2: Ontbrekende eisen

Ontbrekende eisen zijn ook (on)mogelijk om na te gaan, het kan namelijk ook hier voorkomen dat een stakeholder afwezig is of niet met de volle aandacht aanwezig is bij het opstellen van de requirements. Hoe weet je dan dat je alle echt eisen dan hebt? Hoe weet je überhaupt dat je alle eisen tijdens het ontwikkelen van een

conceptueel model? Deze metriek zal alleen effect hebben als tijdens het ontwikkelen deze metriek verschillende keren word toegepast. Of je zou als ontwikkelaar ook jezelf kunnen verplaatsen in een andere rol, bijvoorbeeld die van eindgebruiker. Maar dan riskeer je een valkuil dat je eisen gaat opnemen die de eindgebruiker niet wilt hebben, maar die jij als persoon logisch zou vinden.

MoMetriek 3: Foutief gedefinieerde modelementen

Het is moeilijk te achterhalen wanneer een modelement foutief is. Mocht een modelement foutief zijn gemodelleerd kan het ook om een reden zijn geweest. Zie model 2, daar is het modelement verblijfplaats opgenomen. Volgens deze metriek is verblijfplaats foutief gedefinieerd, terwijl het woonruimte of leefruimte moet zijn van hotelkamers tot appartementen. Dan is verblijfplaats een algemenere zin een betere benaming dan woonruimte of leefruimte.



Figuur 8.1: Entiteittype verblijfplaats

Om de bevinding te verduidelijk is hierboven uit model 2 het entiteittype verblijfplaats afgebeeld. Zodat u weet over welke entiteittype het gaat om misverstanden te voorkomen.

MoMetriek 4: Inconsistenties ten opzichte van het procesmodel

Deze metriek vond ik belangrijk, doordat je op twee manieren naar het probleem-domein kijkt kom je eerder tot een goed model. Als je een procesmodel maakt denk je na over de informatie die je nodig hebt om het proces te doorlopen. Mocht je informatie ontdekken die je nodig hebt in het proces, zal deze informatie ook in het conceptueel model opgenomen moeten worden. De inconsistenties zorgen ervoor dat het conceptueel model goed aansluit aan het proces dat het informatiesysteem moet gaan ondersteunen.

MaMetriek 9: Realistische spiegeling van het bedrijfsproces

Bij deze metriek wordt er gekeken of het procesmodel realistisch is ten opzichte van het de werkelijkheid. In feite doe je bij deze metriek precies hetzelfde als bij metriek 4.

MaMetriek 11: Zijn alle elementen relevant

Bij deze metriek is het moeilijk te achterhalen of de elementen relevant zijn. Ik denk dat je aan de hand van de volgende vragen kan achterhalen of de elementen relevant zijn:

- Waarom is het element in eerste instantie opgenomen? Als het antwoord geen duidelijke argumentatie is dan zal het element wellicht overbodig zijn.
- Wat is het doel van het element? Als het doel niet duidelijk beschreven kan worden dan kan het zijn dat het element niet relevant is.
- Het kan ook helpen door iemand naar je model te laten kijken die niet bij de ontwikkeling betrokken is. Als deze persoon snapt wat er gemodelleerd is, dan is het mogelijk dat de elementen relevant zijn. Anders zou hij of zij er vraagtakens bij zitten en er naar vragen waarom dat zo gemodelleerd is.

MaMetriek 13: Informatiebehoeften

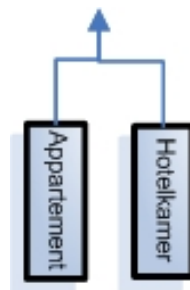
Bij deze metriek is het vooral belangrijk dat je afvraagt welke informatie heb ik nodig en welke informatie kan ik uit het model halen. Als de antwoorden van deze twee vragen overeenkomen dan zul je waarschijnlijk alle informatie behoeften hebben gemodelleerd.

PoMetriek 1: Het aantal entiteiten

Deze metriek is veruit het makkelijkste in gebruik van alle compleetheidsmetrieken. Het is gewoon een kwestie van tellen en er komt een getal uit, simpeler kan niet. Doordat je weet dat er een getal uit moet komen is het ook makkelijker te controleren.

PoMetriek 12: Het aantal afgeleide entiteitstypen

Deze metriek is net zoals PoMetriek 1 makkelijk in uitvoering, alleen hier moet er goed opgelet worden dat men de juiste entiteitstypen telt. Het gaat dus alleen om specialisatie entiteitstypen. Deze uitkomst kan ook nooit meer zijn dan PoMetriek 1.



Figuur 8.2: Afgeleide (specialisatie) entiteitstypen

Hierboven word eruit model 1 de specialisatie typen weergegeven. Specialisatie typen zijn in de ER-modellen te herkennen aan de pijlen, zoals afgebeeld.

Overeenkomsten en verschillen:

Ten opzichte van alle andere metrieken die compleetheid bevat, zijn PoMetriek 1 en 12 het opvallendst. Het grote verschil zit hem in de uitkomst van deze metrieken. Waar de andere metrieken een beschrijving of een lijst met fouten genereren krijg je hier als uitkomst een getal. Ook qua toepasbaarheid is PoMetriek 1 anders dan de andere metrieken. Waar de andere metrieken je aan het denken zitten waarom je precies iets hebt gemodelleerd, moet je bij deze metriek de entiteiten tellen. Terwijl je bij de andere metrieken maar moet afwachten wat de uitkomst zal zijn, weet je van te voren dat hier een getal uit moet komen, dit is een geruststellend gevoel omdat je weet dat het antwoord meestal goed te begrijpen is. PoMetriek 12 is ongeveer hetzelfde als PoMetriek 1 alleen bij deze metriek draait het om de afgeleide entiteitstypen (specialisatie typen). Zie afbeelding 8.2 voor afgeleide entiteitstypen.

MoMetriek 4, MaMetriek 9 en 13 overlappen elkaar doordat ze allemaal kijken naar de informatie binnen het conceptueel model en het procesmodel. Als er verschillen tussen deze modellen optreden dan dient één van de twee modellen aangepast te worden of wellicht beide modellen. Het grote verschil met de andere metrieken is de manier van toepassen, de metrieken gebruik je als een vergelijking tussen het procesmodel en het conceptueel model. Het voordeel van deze metrieken is het controleren aan de hand van het proces dat het informatiesysteem uiteindelijk moet ondersteunen. Het proces doorlopen brengt precies naar boven welke informatie en eisen je nodig hebt. Hoewel het een erg tijdrovend zal zijn om alle processen binnen

de organisatie te modelleren en te controleren.

Op basis van de controle van de modelementen binnen het conceptueel model zijn MoMetriek 3 en MaMetriek 11 overeenkomend. MoMetriek 1 is ook een controle van de modelementen maar MoMetriek 1 bekijkt de modelementen van een conceptueel model vanuit de eisen die de stakeholders hebben opgegeven. MoMetriek 3 en MaMetriek 11 kijken los van de eisen van het conceptueel model naar het modelement. Dit doe je door jezelf vragen te stellen over deze modelementen. Zoals beschreven bij MaMetriek 11.

Alle stakeholders hebben eisen opgesteld en aan de hand van MoMetriek 1 en 2 worden deze eisen gecontroleerd. Hoewel er een klein verschil zit in de benadering, want bij MoMetriek 1 wordt er uitgegaan vanuit de eisen en bij MoMetriek 2 wordt uitgegaan vanuit het conceptueel model. Ondanks dit verschil kijken beide metrieken of de opgestelde eisen binnen het conceptueel terug te vinden zijn in het conceptueel model. Wat beide metrieken ook in gemeen hebben is dat de metrieken bijna tot niet uit te voeren zijn. En als ze uit te voeren zijn dan zijn de metrieken ontzettend traag en vergen veel tijd.

8.2 Integriteit

Note: In deze scriptie zijn geen business rules gedefinieerd. Toch zal ik bij MoMetriek 5 en 6 mijn mening geven over hoe ik de metriek zou hebben toegepast op het conceptueel model. De definitie die Moody [23] heeft opgesteld "Business rules definiëren wat wel en niet dient te gebeuren met de gegevens" zal ik gebruiken voor de MoMetrieken 5 en 6.

MoMetriek 5: Niet afgedwongen business rules

Door de gegevens die je nodig hebt voor het boeken van een reis op een rijtje te zetten en te bedenken wat er mee dient te gebeuren, kun je nagaan of de business rules die zijn opgesteld worden afgedwongen.

MoMetriek 6: Niet overeenkomende business rules

De niet overeenkomende business rules zijn volgens mij makkelijk te controleren. Net zoals met de eisen kan er via een lijst gewerkt worden met daarop de business rules. Maar het zal moeilijk zijn om na te gaan of de business rules echt afgedwongen worden zonder dat het informatiesysteem actief is.

MaMetriek 10: Tegenstrijdige elementen

Tegenstrijdige elementen, wat is dat precies vraag ik me af. Zijn dat twee of meerdere elementen die zijn opgenomen met dezelfde definitie maar met een andere benaming? Of zijn dit elementen die elkaar als het ware opheffen in het model? Bijvoorbeeld als er een eis van een stakeholder beschrijft dat het absoluut niet mogelijk mag zijn om mobiele nummers toe te voegen aan een reiziger. Terwijl de reis alleen geboekt kan worden door verplicht te stellen van het opgeven van een mobiel nummer. Maes mag bij deze metriek duidelijker zijn met haar definitie wat nu precies met tegenstrijdig bedoeld wordt.

MaMetriek 14: Niet efficiënt verstrekken van informatie

Zonder het programma in gebruik te nemen is het volgens moeilijk na te gaan wat de mogelijke efficiëntie van de informatieverstrekking is. Tenzij de gebruiker al gewerkt heeft met een soort gelijk informatiesysteem kan er uit ervaring worden gekeken

welke informatie nodig is. Wat ook mogelijk is door zelf na te gaan welke informatie nodig is van een reiziger door middel van een procesmodel. Door een lijst te maken van informatie die nodig is bij een reisboeking kan je nagaan of deze informatie ook is terug te vinden in het conceptueel model.

MaMetriek 15: Efficiënt verstrekken van informatie

Of het model efficiënt is in het verstrekken van informatie is een andere vraag, omdat je niet weet of de informatie snel en overzichtelijk wordt opgehaald of weergegeven. Deze metriek is volgens mij niet na te gaan, want efficiëntie is als op basis van bepaalde factoren (snelheid, overzicht, etc.) deze factoren niet worden meegenomen in deze metriek. Hierdoor is het volgens mij onmogelijk om de efficiëntie van informatie verstrekking van een conceptueel model na te gaan.

Overeenkomsten en verschillen:

MoMetriek 5 en 6 maken gebruik van de business rules binnen een organisatie. Door het gebruik van business rules kan wellicht worden gezien welke informatie er nodig is. Verder is het volgens mij niet echt belangrijk of de business rules echt worden afgedwongen.

De toepassing van MaMetriek 10 is ten opzichte van de andere metrieken totaal verschillend. Waar de andere metrieken kijken naar de business rules en de informatie binnen het conceptueel model, word er met deze metriek gekeken naar de tegenstrijdige elementen. Maar waar deze metriek heeft wel een overeenkomst met MoMetriek 5 en 6, dat is de onduidelijkheid hoe je deze metriek moet gaan toepassen.

Hoewel de MaMetrieken 14 en 15 het tegenovergestelde willen bereiken zijn ze qua benadering hetzelfde. Met beide metrieken ga je kijken naar de informatieverstrekking van het model. De onderlinge uitkomsten zullen wellicht van elkaar verschillen.

8.3 Flexibiliteit

MoMetriek 7: Het aantal veranderende elementen

Met de blik op de toekomst weet je eigenlijk nooit helemaal zeker wat er gaat veranderen in je informatiesysteem. Met deze metriek kun je alleen een schatting maken van welke elementen binnen het conceptueel model *kunnen* veranderen. Het is wel goed om na te denken over de toekomst maar het is lastig om te bedenken welke elementen veranderen gaan.

MoMetriek 8: Geschatte kosten van veranderingen

Deze metriek heb ik niet toegepast aangezien er geen kosten zijn verbonden aan deze casus.

MoMetriek 9: Strategisch belang van wijzigingen

Met deze metriek was het even worstelen, het was moeilijk om een beeld te vormen wat het strategisch belang is van de wijzigingen. Want bij strategisch denk ik aan welke wijzigingen van invloed zijn op het proces waarbij het informatiesysteem moet gaan ondersteunen. Als er tijd mee gewonnen kan worden dan zouden de wijzigingen strategisch van belang zijn. Mocht de doorlooptijd van het proces verkort worden maar het conceptueel model moet hiervoor aangepast worden met als resultaat dat de doorlooptijd van de andere processen worden verlengt, wat ga je dan doen?

Overeenkomsten en verschillen:

De veranderingen van modelementen bij MoMetriek 7 en het strategische belang van deze veranderingen bij MoMetriek 9 worden op dezelfde manier op het conceptueel model toegepast. Bij beide metrieken kijk je naar in de toekomst veranderende modelementen. Alleen de aanpak van MoMetriek 9 is alleen verschillend ten opzichte van MoMetriek 7, door af te vragen wat het strategische belang is van deze veranderende modelementen. Wat beide metrieken ook als overeenkomst hebben is de het moeilijk na te lopen is welke modelementen kunnen veranderen. Wat Moody precies als uitkomst bij deze metrieken had willen hebben is onduidelijk.

De aanpak van MoMetriek 8 is weliswaar hetzelfde in verhouding tot MoMetriek 6 en 9, want er wordt weer gekeken naar de modelementen, alleen de uitkomst zijn de mogelijke kosten verbonden aan deze modelementen.

8.4 Begrijpbaarheid

MoMetriek 10: Beoordeling door stakeholders

De beoordeling van een model door een stakeholder is volgens mij niet te doen, tenzij de stakeholder verstand heeft van conceptuele modellen. Er zijn denk ik weinig stakeholders die weten hoe ze een conceptueel model moeten lezen, laat staan beoordelen. Als ik bijvoorbeeld dit model aan mijn moeder geef en zij is een stakeholder dan zal ze geen flauw idee hebben wat het allemaal precies inhoudt. Dit is daarom een onbetrouwbare metriek. Waarschijnlijk zal de maker van het conceptueel model het model gaan uitleggen en op basis van deze uitleg en de interpretatie van de maker zal de stakeholder een cijfer geven. Als maker kan je dus onbedoeld de zaak te “manipuleren”, door alleen maar dingen op te noemen die relevant zijn voor de stakeholder.

MoMetriek 11: Interpretatie stakeholder

Moody beschrijft dat deze metriek gemeten kan worden op basis van business scenario's of door een populatie van het model te laten maken. Doordat de stakeholder een fout maakt kan zo gemeten worden hoe goed de stakeholder het model begrijpt. Hierbij grijp ik terug op de werking van metriek 10. Hoe kan een stakeholder überhaupt bevatten wat het conceptueel model precies beslaat.

MoMetriek 12: Oordeel ontwikkelaars

Deze metriek is zo subjectief dat de metriek bijna tot niet te gebruiken is. Mocht een van de ontwikkelaars tegen het project zijn van een nieuw informatiesysteem zal ontwikkelaar een negatief beeld scheppen. Zodat men verder gaat met het oude informatiesysteem (wellicht door hem/haar ontwikkeld). Zelf zou ik het conceptueel model in samenwerking met de ontwikkelaar ontwikkelen. Hierdoor voorkom je eventuele problemen dat de ontwikkelaar het conceptueel model niet goed snapt. Mocht het geval zijn dat de ontwikkelaar tegen het project is dan kan het zijn door de ontwikkelaar bij de ontwikkeling te betrekken te overtuigen dat een nieuw informatiesysteem nodig is.

MaMetriek 1: Makkelijk te begrijpen

De metriek kan alleen uitgevoerd worden door mensen die bekend zijn met het probleemdomen en met het ontwikkelen van conceptuele modellen. Als bijvoorbeeld iemand zonder domein- en modelkennis het conceptueel model moet begrijpen, dan zal deze persoon geen idee hebben wat er nu precies bedoeld wordt met het model. Daarom is deze metriek moeilijk toe te passen.

MaMetriek 4: Lezen is makkelijk

Het leren lezen van het conceptueel model is volgens mij bedoeld als een ontwikkelaar of een expert het model moet beoordelen. Het toepassen van deze metriek is voornamelijk op basis van het model ontworpen door Marc Pieterse, hierdoor wist ik niet goed wat ik nu moest antwoorden. Is het gewoon een ja of nee? Of moet ik een cijfer koppelen aan de leesbaarheid. Deze metriek vond ik ook niet echt bijzonder nuttig, vooral door die onbetrouwbare uitkomst.

MaMetriek 6: Helpt bij het begrijpen

Dit is weer een van de metrieken waarvan ik van mening ben dat het handig is om te kijken of het proces overeenkomt met het conceptueel model. Om te achterhalen welke informatie belangrijk is binnen het proces zodat het conceptueel model hierop afgestemd kan worden.

MaMetriek 7: Verbeterd de prestatie om te begrijpen

Het verbeteren van de prestatie om het conceptueel model beter te begrijpen. Deze metriek vond ik een beetje vaag omschreven, hoewel ik wel snap wat Maes er precies mee bedoelt, vind ik deze metriek eigenlijk overbodig als je MaMetriek 6 al hebt gebruikt. Want feitelijk doe je twee keer hetzelfde, bij beide metrieken word er gekeken of het proces beter te begrijpen is aan de hand van het conceptueel model.

PoMetriek 6: Het aantal relaties

Deze metriek was vrij simpel om na te lopen. De relaties in een conceptueel model zijn de zwarte lijnen die entiteitentypen met elkaar verbinden. In het eerste model waren er 11 relaties te vinden en in het tweede model 5. De werking was helder, snel en gemakkelijk. Alleen nu hebben we getallen als uitkomst, maar er wordt niet precies gedefinieerd wat er uiteindelijk mee moet gebeuren.



Figuur 8.3: ER Relatie

De lijnen zoals hierboven afgebeeld vormen de relaties binnen de ER-modellen. Normaal word er ook de kardinaliteit aangegeven zoals te zien in afbeelding 8.4.

PoMetriek 7: Het aantal M:N relaties

Wat er precies met deze metriek wordt bedoeld is onduidelijk. Waarschijnlijk zal Poels een N op N relaties bedoelen. Deze metriek heb ik wel uitgevoerd maar volgens mij definitie (N op N) is er geen relatie gevonden binnen de twee ER modellen. Doordat er niet precies duidelijk is wat er bedoeld word, bestempel ik deze metriek als onbruikbaar.

PoMetriek 8: Het aantal 1:N relaties

Het aantal 1:N relaties zijn naar mijn weten veel voorkomend binnen ER modellen. Deze metriek is gemakkelijk, helder, snel toe te passen en duidelijk geformuleerd. Een kantekening is wel nodig want het aantal 1:N relaties kan nooit meer zijn dan het totaal aantal relaties gedefinieerd in PoMetriek 6.



Figuur 8.4: 1:N Relatie

De lijn hierboven afgebeeld vormt een 1:N relatie. Bij deze relatie is de kardinaliteit

wel aangegeven, dit kunt u herkennen aan de $1..n$ op de lijn.

PoMetriek 9: Het aantal N-Ary relaties

Bij deze metriek geldt eigenlijk hetzelfde als PoMetriek 7. Hier is het ook niet goed gedefinieerd wat er precies de bedoeling is. Welke relaties nu binnen deze categorie vallen en welke niet. Hierdoor is deze metriek ook niet te gebruiken in dit onderzoek.

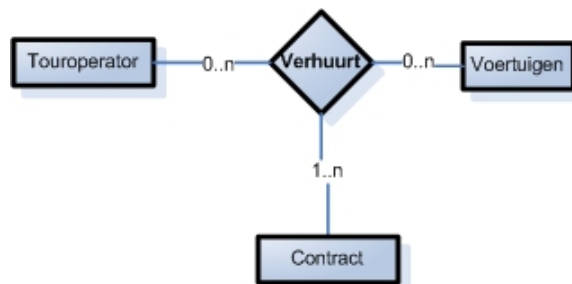
PoMetriek 10: Het aantal binaire relaties

Het aantal binaire relaties is vrijwel gelijk aan het aantal relaties. In de ER modellen zijn het aantal relaties en aantal binaire relaties gelijk. Deze metriek werkt ook hetzelfde als PoMetriek 1, snel, makkelijk en duidelijk gedefinieerd wat er precies de bedoeling is.



Figuur 8.5: Binaire relatie

De afbeelding hierboven is een voorbeeld van een binaire relatie. Binair wilt zeggen dat er een relatie tussen maximaal twee entiteitstypen is.



Figuur 8.6: Ternaire relatie

De afbeelding hierboven is een voorbeeld van een ternaire relatie. In tegenstelling tot binair bestaat ternair maximaal uit drie entiteitstypen. In tegenstelling tot model 1, komt er in model 2 een ternaire relatie voor.

PoMetriek 11: Het aantal reflexieve relaties

Ook hier weer de onduidelijke definitie van reflexieve relaties. Doordat er weer een onduidelijke definitie is gegeven van deze metriek, is deze metriek ook niet te gebruiken.

Overeenkomsten en verschillen:

MoMetriek 10 en 12 zijn qua aanpak hetzelfde, er wordt een oordeel gegeven over het conceptueel model. Beide metrieken hebben ook als overeenkomst dat ze subjectief zijn, dit komt doordat de ontwikkelaar en stakeholder een cijfer geven aan het conceptueel model. Hierdoor zijn beide metrieken ook onbetrouwbaar. Er is ook een verband te ontdekken met MoMetriek 11 in verhouding tot de MoMetriek 10 en 12, want het oordeel wordt gegeven op basis van de interpretatie van de stakeholder en

ontwikkelaar. Maar toch is de aanpak van MoMetriek 11 anders, want bij MoMetriek 11 word er gemeten aan de hand van business scenario's.

De basis van MaMetriek 1, 4, 6 en 7 is het begrijpen van het conceptueel model of het helpen om het conceptueel model te begrijpen. MoMetriek 11 die aan de hand van business scenario's gemeten word past ook tussen deze metrieken, want met behulp van de business scenario's geeft de metriek aan of de stakeholder het model begrijpt. MaMetrieken 1 en 4 gaan voornamelijk over het begrijpen van het conceptueel model, terwijl MaMetriek 6 en 7 beschrijven dat het conceptueel model heeft geholpen om het bedrijfsproces beter te kunnen begrijpen. Ondanks dit kleine verschil dat draait om de begrijpbaarheid, is de uitkomst van alle vier de metrieken onduidelijk. Is het ook hier de bedoeling van een simpele ja of nee? Of dient er een uitgebreider antwoord te komen.

De PoMetriek van Poels hebben alleen betrekking op de begrijpbaarheid. Door het aantal relaties binnen conceptueel model kan het zijn dat het model niet begrijpbaar is. In verhouding kan je dus zien dat het ER model gepresenteerd in de casus 11 relaties heeft een stuk onbegrijpelijker is ten opzichte van het ER model gepresenteerd in appendix B. Dus de overeenkomst tussen deze metrieken is de controle van het aantal relaties. Alleen de benadering is per PoMetriek verschillend. Bij PoMetriek 6 gaat het totaal aantal relaties tussen de entiteitstypen, bij PoMetriek 8 betreft het alleen om de 1:N relaties. PoMetriek 7, 9 en 11 zijn alle drie in benadering vaag en onduidelijk door dat je niet weet welke relaties je moet bekijken zijn deze metrieken niet uit te voeren. PoMetriek 10 heeft nog het meest weg van PoMetriek 6, want in deze ER modellen zijn de uitkomsten gelijk.

8.5 Correctheid

MoMetriek 13: Conventionele overtredingen

Bij deze metriek is het gebruik van een checklist niet overbodig. Door op te sommen welke conventies er binnen bepaalde ontwikkelvormen van conceptuele modellen aanwezig zijn, kan er gemakkelijker worden bekeken welke conventies worden overtreden. Mocht er automatisch code gegenereerd worden op basis van het conceptueel model, dan is het wenselijk dat er precies beschreven staat wat je wilt modelleren. Ook dat het conceptueel model voldoet aan de conventies, anders zou het voor kunnen komen dat er foutieve code gegenereerd kan worden.

MoMetriek 14: Normaalvorm fouten

Er is niet gekeken naar normaalvorm fouten in het conceptueel model. Omdat deze metriek weinig toevoegd aan het verbeteren van conceptueel model aan de hand van het proces. Het enige wat ik zou kunnen voorstellen waar deze metriek voor nut heeft is de snelheid waarmee de informatie uit de database word opgevraagd te verbeteren.

MoMetriek 15: Redundantie

Tijdens het toepassen van deze metriek viel het me op dat je niet weet wat nu precies redundantie is. Zijn dit entiteiten die dubbel zijn opgenomen in het conceptueel model het zei met een andere benaming? Of zijn dit redundante eisen die zijn opgenomen in het conceptueel model? Ik heb deze metriek toegepast op basis van de eerste definitie. Hieruit is naarvoren gekomen naar mijn mening de entiteiten, vliegtuigmaatschappij met de tickets (ticket en eticket) en de verzekeringsmaatschappij met de polis. Want eigenlijk kan je bij het ticket een attribuut opnemen waarin staat

vliegtuigmaatschappij en dit geldt hetzelfde voor de polis, daar kan je de verzekeringsmaatschappij opnemen als attribuut.

MaMetriek 8: Bedrijfsproces komt terug

Door het procesmodel en het conceptueel model langs elkaar te leggen, kan je zien of de gedefinieerde entiteiten van het procesmodel die overeenkomen met de entiteiten in het conceptueel model. Het toepassen van deze metriek was makkelijk, het is een kwestie van vergelijken en de bevindingen beschrijven.

MaMetriek 16: Tevreden over informatie verstrekking

Aan de hand van de informatie die tussen de entiteiten over en weer gaan, kan je kijken of deze informatie ook terug te vinden is in het conceptueel model. Maar om er zeker van te zijn dat alle informatie in het conceptueel model aanwezig is, kan men het beste van alle processen die de reisorganisatie bevat een procesmodel maken. Dit is een tijdrovende klus en vergelijkbaar met conceptuele modellen weet je niet zeker of de procesmodellen goed zijn. Met andere woorden, of alle informatie die nodig is in het model is beschreven.

Overeenkomsten en verschillen:

De metrieken die ontwikkeld zijn om de correctheid te evalueren hebben onderling vrijwel geen overeenkomsten. MoMetriek 13 heeft het over conventie overtredingen. Conventies zijn een soort standaard waarmee je een conceptueel model mee tekent of ontwikkelt. Als deze worden overtreden door bijvoorbeeld verkeerd gebruik van de syntax, kan dit met een checklist worden nagegaan. Terwijl MoMetriek 13 via een checklist is na te kijken, dit zou door een tool gedaan kunnen worden. Zodat je een subjectieve evaluatie krijgt.

De rest van de metrieken hebben de enige overeenkomst dat ze allemaal objectief zijn. Normaalvorm fouten zijn gebaseerd op een database dus deze metriek is niet van toepassing op elk conceptueel model. Maar deze normaalvorm fouten moeten door een persoon worden ontdekt en worden verwijderd. Het ligt dus aan die persoon of het een normaalvorm fout is. MoMetriek 14 is dus niet altijd toepasbaar en onbetrouwbaar.

MoMetriek 15 is de evaluatie van redundantie binnen het conceptueel model. Deze metriek werkt op basis van evaluatie op entiteit niveau, dit is zo verschillend vergeleken met de andere correctheidsmetrieken. Want de andere metrieken evalueren het conceptueel model als een geheel. Als je kijkt naar MaMetriek 8 word er gekeken vanuit het bedrijfsproces of dat terug komt in het conceptueel model. Terwijl er bij MaMetriek 16 word gekeken of de informatie die wordt verstrekt ook daadwerkelijk nuttig is voor het te ondersteunen proces.

8.6 Eenvoud

MoMetriek 16: Hoeveelheid entiteitstypen

De hoeveelheid van entiteitstypen is makkelijk uit te tellen. De metriek was makkelijk, snel, foutloos en helder toe te passen. Aan de hand van afbeelding hieronder afgebeeld is het makkelijk te zien hoeveel entiteitstypen het ER-model heeft.



Figuur 8.7: Entiteittype

MoMetriek 17: Entiteiten en hun relaties

Omdat het tellen van de entiteiten bij MoMetriek 16 al aan de orde is geweest, vond ik het een beetje dubbelzinning om bij deze metriek nog een keer de entiteiten te noemen. Daarom heb ik alleen de relaties geteld. Voor de rest was deze metriek ook makkelijk te begrijpen en toe te passen.

MoMetriek 18: Het aantal constructen

Het oordeel over deze metriek is hetzelfde als MoMetriek 16 en 17. De metriek was makkelijk, snel en eenvoudig toe te passen. Dit komt volgens omdat je weet dat er een getal uit moet komen.

MaMetriek 2: Frustratie met gebruik

Tijdens het toepassen van deze metriek was het voor mij onduidelijk wat nu met frustratie bedoeld werd. Frustrerend als ontwikkelaar doordat het conceptueel model te complex is? Als stakeholder, dat er geen goed overzicht in het conceptueel model is gecreëerd van de eisen? Of frustrerend als gebruiker dat de goede informatie niet terug te vinden is in het conceptueel model. Hierdoor is het onduidelijk hoe ik deze metriek nu moet toe gaan passen.

MaMetriek 28: Makkelijk in gebruik

Bij deze metriek vroeg ik me af, hoe kan ik het conceptueel model gebruiken? Het is niet terug te halen uit de literatuur hoe Maes deze metriek toepast op een conceptueel model. Op basis van metriek 18 heb ik deze metriek toegepast. Ik heb namelijk gekeken naar het aantal constructen en hieraan een oordeel gegeven of het model makkelijk in gebruik is.

PoMetriek 2: Het aantal attributen

Het totaal aantal attributen is gemakkelijk te zien en te tellen binnen een ER model. Dit zijn alle ovalen die voorkomen in het ER model.



Figuur 8.8: Attribuut

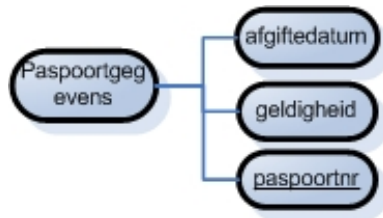
In een ER-model zijn de attributen te herkennen aan de ovalen, zoals hierboven in de afbeelding te zien is.

PoMetriek 3: Het aantal afgeleide attributen

Het aantal afgeleide attributen is een vreemde definitie. Want eigenlijk is elk attribuut binnen de ER modellen afgeleid van een entiteittype. De definitie die Poels gebruikt kon ik niet terug plaatsen in de ER modellen. Doordat het onduidelijk is wat de bedoeling van deze metriek is, reken ik deze metriek als onbruikbaar.

PoMetriek 4: Het aantal samengestelde attributen

Het aantal samengestelde metrieken spreekt eigenlijk voor zich. Bijvoorbeeld het attribuut NAW is een attribuut die kan voorkomen bij klanten van een reisbureau. Het attribuut NAW is opgesteld uit Naam, Adres, Woonplaats, Postcode etc. Deze metriek was helder gedefinieerd en gemakkelijk toe te passen op de ER modellen.



Figuur 8.9: Samengesteld attribuut

Een samengesteld attribuut is opgedeeld in verschillende attributen. Zoals paspoortgegevens in afbeelding 8.9, deze kan je op delen in overige drie andere attributen.

PoMetriek 5: Het aantal meerwaardige attributen

De definitie die ik aan deze metriek geef is, meerwaardige attributen betekenen dat het attribuut van een bepaald entiteitstype meer de waardes kan bevatten. Verder is het onduidelijk wat er nu bedoeld wordt. Het kan dus zijn dat deze metriek de plank mis slaat. Want hoe vertaald zich de meedere waardes in een database?

Overeenkomsten en verschillen:

De basis van MoMetriek 16, 17, en 18 is voornamelijk het tellen van de entiteiten, relaties en attributen. De uitkomsten van deze metrieken zijn dan ook getallen. In tegenstelling tot de andere metrieken die Moody heeft beschreven zijn deze metrieken erg gemakkelijk en snel om toe te passen.

Het verschil tussen de MoMetriek 16, 17 en 18 en MaMetriek 2 en 3 is de manier waarop je het conceptueel model evalueert. Bij MaMetriek 2 en 3 bekijk je of het conceptueel model gemakkelijk of frustrerend is in gebruik. Waarbij één van de metrieken waar hoort te zijn, hoewel ik denk dat het soms ook een combinatie is van beide metrieken. Terwijl je het conceptueel model bij MoMetriek 16, 17 en 18 eigenlijk niet evalueert alleen tel je de modelelementen. Verder zijn aan de uitkomsten van alle metrieken geen consequenties gebonden. Of oplossingen om bijvoorbeeld het conceptueel model minder frustrerend in gebruik te maken.

Alle PoMetrieken zijn van invloed op de eenvoud van het ER model. Het aantal attributen binnen het ER model zorgen ervoor dat het ER model moeilijk te implementeren is, of te integreren en nog niet over de begrijpbaarheid niet te spreken. PoMetriek 3, 4 en 5 zijn ongeveer hetzelfde doordat er wordt gekeken naar een speciaal soort attribuut binnen de ER modellen. Maar PoMetriek 3 en 5 hebben geen goede definitie en daarom zijn deze ook moeilijk om toe te passen. PoMetriek 2 en 4 zijn echter wel goed geformuleerd en naar mijn mening vaker voorkomend binnen een ER model.

8.7 Integratie

Van de reisorganisatie is er geen bestaand systeem ontwikkeld, of zijn er modellen ontwikkeld in deze scriptie waarmee het informatiesysteem moet gaan integreren.

Hierdoor komen de metrieken van deze kwaliteitseigenschap te vervallen. Het vervallen van deze metrieken zal geen groot effect hebben op de uitkomst van deze scriptie, omdat ik denk dat de integratie met andere informatiesystemen moeilijk te meten valt met de voorgeschreven metrieken.

8.8 Implementeerbaarheid

MoMetriek 23: Schatting technisch risico

De schatting van technisch risico wordt gemeten aan de hand van de overeenstemming met de eisen die in het begin van deze casus zijn opgesteld. Tijdens het toepassen van deze metriek kwam ik tot de ontdekking dat het schatten niet mogelijk is. Ondanks dat je alle informatie (eisen, budget, resources, etc) voor handen hebt. Je kan inschatten dat het een risico is als de eisen niet zijn opgenomen in het conceptueel model. Stel dat alle eisen opgenomen zijn het in conceptueel model, maar het conceptueel model is hierdoor een groot complex geheel geworden waar ontwikkelaars niet uit kunnen komen. Wat voor zin heeft het dan nog om een schatting te maken van een technischrisico op basis van de eisen.

MoMetriek 24: Schatting tijdsduur

Op basis van de constructen beschreven in MoMetriek 18 (onder de kop eenvoud in Appendix A) heb ik gekeken of ik een schatting kon maken. Hieruit is gekomen dat model 1 meer tijd zal kosten tijdens het ontwikkelen dan model 2. Ik denk dat dit komt omdat het aantal constructen van model 1 op 44 ligt en van model 2 op 27. Door alleen het verwerken van de constructen in model 1 in het informatiesysteem zal al meer tijd kosten, maar ook het interpreteren van model 1 zal door het aantal constructen meer tijd vergen. Om nu een echte schatting te geven in maanden of dagen is volgens mij alleen mogelijk als je een ontwikkelaar bent en verstand hebt van ontwikkelen op basis van een conceptueel model. Dus deze metriek is volgens mij niet goed bruikbaar in de praktijk, tenzij je aan de genoemde eisen voldoet.

MoMetriek 25: Schatting ontwikkelkosten

Het oordeel over deze metriek is grotendeels hetzelfde als MoMetriek 24. De ontwikkelkosten staan in verband met de tijdsduur van het conceptueel model. Mocht het zijn dat het conceptueel model als complex word beschouwd zal dit invloed hebben op de tijdsduur van het ontwikkeling en de kosten van het project. Hoe dan ook, MoMetriek 24 en 25 hebben een onderling verband. Ik kan niet zeggen of het echt een toevoeging is voor de modelmetrieken, het zal altijd een schatting blijven. Wat ook opviel is de oplossing mocht er over het budget worden gegaan. Dan wordt of het budget vergroot of het model versimpeld. Het vergroten van het budget kan niet altijd, dus kan het zijn dat sommige eisen niet worden meegenomen of dat de klant een product krijgt waar hij niet op zit te wachten. Dit geldt hetzelfde voor het versimpelen van het conceptueel model, dit kan je doen door eisen weg te laten. Maar het herzien van het conceptueel model kost ook weer tijd en geld. De metriek is daarom niet goed gedefinieerd vooral met betrekking tot de oplossing en hierdoor deels (on)bruikbaar.

MaMetriek 5: Verbetering ten opzichte van tekst

Ik ben van mening dat alle conceptuele modellen altijd een verbetering zijn ten opzichte van tekst. Het enige nadeel van een conceptueel model is dat iedereen het op een andere manier interpreteerd. Maar dat geldt volgens mij ook voor teksten. Het procesmodel van beide ER modellen zijn makkelijker te lezen dan een tekst waarbij de activiteiten worden beschreven. Deze metriek vond ik tijdens het

toepassen eigenlijk nietszeggend. Of je deze metriek nu toepast of niet het zal geen effect hebben op het conceptueel model of procesmodel. Ik snap eigenlijk ook niet wat eruit deze metriek moet voortkomen. Gewoon een simpele *ja*? Kortom overbodig en niet bruikbaar.

Overeenkomsten en verschillen:

MoMetriek 24 en 25 hebben een onderling verband, namelijk als de tijdsduur van het project groter wordt dan zullen de ontwikkelkosten toenemen. Als de ontwikkelkosten toenemen dan zal het project hoogst waarschijnlijk meer tijd kosten om te ontwikkelen. Verder is het verschil tussen MoMetriek 24 waarbij het over de doorlooptijd van het project gaat en bij MoMetriek 25 wat de mogelijke kosten zijn. Beide metrieken hebben ook een overeenkomst, dat het moeilijk is om te schatten hoeveel tijd en geld een project kost. Tenzij je een ervaren ontwikkelaar bent die werkt op basis van conceptuele modellen, anders kom je bijna tot niet op een reële schatting van het project.

Het schatten bij MoMetriek 23 is een kleine overeenkomst vergeleken met MoMetriek 24 en 25. Maar het verschil zit hem in benadering, waar MoMetriek 24 en 25 kijken naar het hele project. Kijkt MoMetriek 23 alleen naar de eisen van het conceptueel model en evalueert op basis hiervan het conceptueel model.

MaMetriek 5 is een wat vreemde metriek wat betreft tot de kwaliteitseigenschap implementeerbaarheid. Op basis van het procesmodel wordt er gekeken of het conceptueel model een verbetering is ten opzichte van tekst. Maar ik denk dat op basis van een conceptueel model het informatiesysteem makkelijker binnen de organisatie te implementeren is. MaMetriek 5 is verder ook niet te vergelijken met de andere metrieken.

Tabel 8.1: Modelmetrieken: Verschillen en Overeenkomsten

Auteurs Focus	Verschillen		Overeenkomsten	Subjectief/Objectief?
Poels [31] Onderhoud voor modellen.	ER-	Ten opzichte van de andere auteurs heeft Poels formele metrieken ontwikkeld.	MoMetrieken (17,18 en 19) van Moody tellen ook het aantal entiteiten, relaties en attributen. Dit is de enige overeenkomst die Poels heeft in vergelijking met Maes en Moody.	Objectief
Moody [23] Compleetheid, Integerheid, Flexibiliteit, Begrijpbaarheid, Correctheid, Eenvoud, Intergratie, Implementeerbaarheid		Moody heeft weinig verschillen met Maes. In vergelijking met Poels heeft Moody voornamelijk informele metrieken.	Moody heeft de grootste overeenkomsten met Maes die informele metrieken heeft ontwikkeld. Hoewel Moody ook drie metrieken heeft die ook terug te vinden zijn in de metrieken van Poels.	Subjectief en Objectief
Maes [21] Perceived Ease of Understanding, Perceived Usefulness, Perceived Semantic Quality, User satisfaction	Under-	Maes heeft een informele metrieken ontwikkeld, ten opzichte van Poels die formele metrieken heeft ontwikkeld, is dit een groot verschil.	Maes heeft in een verkapte vorm dezelfde kwaliteitseigenschappen als Moody gebruikt. Tevens zijn de metrieken van Maes ook informeel net zoals die van Moody.	Subjectief

Hoofdstuk 9

Toepassing metrieken op een UML model

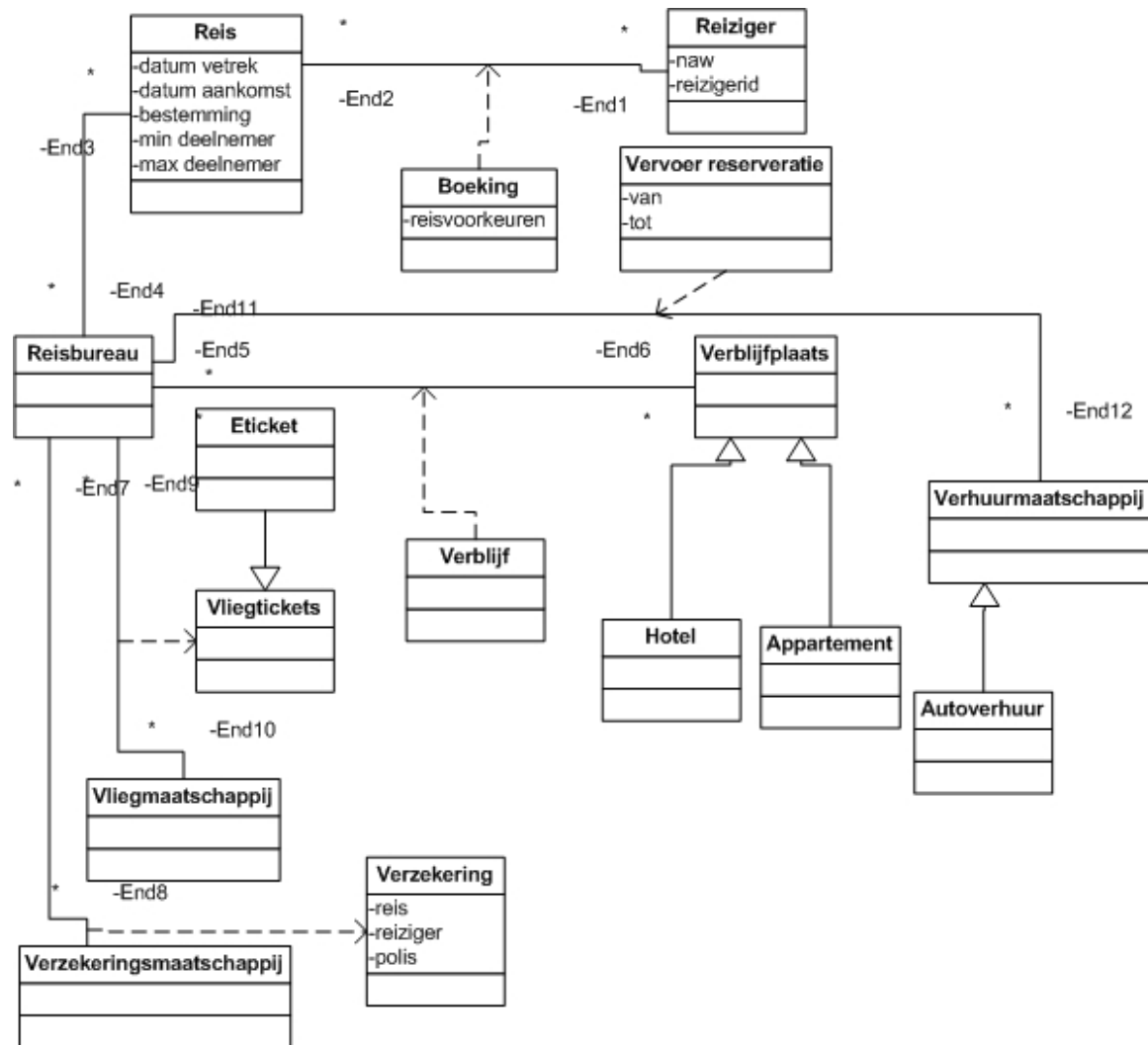
U heeft kunnen zien hoe de metrieken werken en wat de bevindingen zijn op basis van de ER- en procesmodellen. In dit deel van zullen de metrieken nog een keer worden toegepast, alleen nu op een UML model. Dit model waarvoor ik erg dankbaar ben is ontwikkeld is door Marc Pieterse. De beantwoording wordt gebaseerd op de resultaten die terug te vinden zijn in appendix A. Er is gekozen voor het UML model van Marc Pieterse om te voorkomen dat ik het UML model tijdens het ontwikkelen veranderde op basis van de voorkennis die ik heb van de metrieken. Doordat Marc geen metrieken kent of heeft toegepast heb ik de resultaten zo objectief mogelijk kunnen verzamelen. Ik heb ook gekozen voor het UML model van Marc omdat het ER model van Marc (zie afbeelding 13.1, appendix B) beter naar voren kwam na het toepassen van de metrieken. Dit ten opzichte van mijn eigen ER ontwerp (zie afbeelding 7.1). Door het toepassen van de metrieken op het UML model kunnen de volgende twee deelvragen worden beantwoord:

1. Zijn de ER metrieken toepasbaar op een UML-model?
2. Welke problemen kunnen optreden tijdens het toepassen van ER metrieken op UML-modellen?

Na het beantwoorden van deze deelvragen kan de hoofdvraag beantwoord worden. Maar eerst zal het UML model worden geïntroduceerd dat door Marc Pieterse is ontworpen. Op basis van dezelfde eisen die aan het begin van dit deel van de scriptie aanbod kwamen is het volgende UML model (zie afbeelding 9.1 op de volgende bladzijde) gemaakt. Het is noodzakelijk om dezelfde eisen te gebruiken, om zo te zorgen dat aan de hand van het procesmodel (zie afbeelding 13.2 in appendix B) de metrieken kunnen worden toegepast. Als deze eisen niet werden gebruikt en er werd een UML model geïntroduceerd van een ander probleem domein. Dan zouden verschillende metrieken niet meer toepasbaar zijn en zouden wellicht hele andere antwoorden krijgen.

De entiteitstypen in dit model zijn de rechthoeken waar naar zwarte lijnen lopen. Deze entiteitstypen worden ook wel klassen genoemd. UML modellen zijn conceptuele modellen voornamelijk bedoeld voor automatische generatie van softwarecode. De entiteiten hebben attributen die in deze rechthoek zijn beschreven. Afbeelding 9.2 heeft als attributen:

- NAW (afkorting voor naam, adres en woonplaats)



Figuur 9.1: UML model van het reisbureau door Marc Pieterse

- reizigID

Deze attributen zijn vergelijkbaar met de ovalen die in het ER model worden gebruikt. De zwarte lijnen tussen de entiteiten staan voor de onderlinge relaties van de entiteiten. Dit is ook vergelijkbaar met de relaties van de ER modellen. De stippellijn die verwijst naar een relatie tussen twee entiteiten is noodzakelijk voor deze relatie. Als de relatie tot stand komt tussen entiteiten dan zal de entiteit ook altijd aanwezig zijn. De grote witte pijlen in het UML model zijn de specialisatie pijlen. Hiermee wordt bedoeld dat een verblijfplaats zowel een appartement of hotelkamer kan zijn.

Zijn de ER metriekeken toepasbaar op een UML-model?

Het antwoord op deze vraag kan zowel ja als nee zijn. Ja, als het gaat om de beschrijvende metriekeken die door Moody [23] en Maes [21] zijn ontwikkeld. Dit zijn dus de informele metriekeken die algemeen toepasbaar blijken te zijn. Terwijl er maar een fractie van de elf informele metriekeken die Poels [31] heeft ontwikkeld toepasbaar zijn op het UML model. Van de elf informele metriekeken zijn er maar zes toe te passen op het UML model. Ten opzichte van de informele metriekeken waar er 41 van zijn waren

er maar tien niet toe te passen. Doordat er bijvoorbeeld geen business scenario's in de casus aanwezig waren, er is geen bestaand systeem om mee te vergelijken of er is geen budget opgesteld. Bij de toepassing van de formele metrieke door Poels, is wel enige nuance nodig want die metrieke waren voornamelijk op de ER conventies gebaseerd. Deze conventies zijn dan ook niet terug te vinden in een UML model. Met als resultaat dat ze niet toe te passen waren op het UML model.

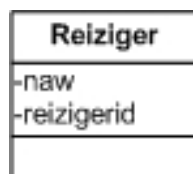
Ondanks dat de metrieke voor een groot deel toepasbaar zijn op het UML model. Is er niet precies naar voren gekomen hoe ik het model nu moet veranderen of aanpassen zodat er een betere kwaliteit ontstaat. Vaak kan er een uitkomst worden gevonden, maar negen van de tien keer is deze uitkomst subjectief (op metrieke van Poels na). Subjectiviteit is ook bij het UML model van toepassing het is maar hoe het UML model wordt geïnterpreteerd. Doordat Marc het UML model heeft ontwikkeld interpreteer ik het model via mijn eigen gedachtegang. Het kan zijn dat ik de metrieke toe heb gepast op het UML model, op een manier waarbij de uitkomst foutief is. Terwijl Marc eigenlijk hetzelfde heeft bedoeld maar anders in het model heeft opgeschreven.

Welke problemen kunnen optreden tijdens het toepassen van ER metrieke op UML-modellen?

Tijdens het toepassen van de metrieke op het UML model zijn ook een aantal problemen naar voren gekomen. Voornamelijk die te maken hebben met de ER conventies. Dit probleem was van te voren verwacht en heeft ook geen effect op de uitkomst van dit onderzoek. Dit komt omdat het over een klein gedeelte van alle metrieke gaat. Want van de 53 metrieke die in dit onderzoek zijn gebruikt werden er in totaal 16 niet toegepast. Vooral de metrieke van Poels zijn op basis van deze ER conventies gebaseerd en daarom ook niet mogelijk om toe te passen.

De metrieke van Moody die ze bezig houden met het tellen van modelementen konden toegepast worden op het UML model. Neem bijvoorbeeld metriek 18: Het aantal constructen van Moody. Daarbij moet er gekeken worden naar de entiteiten, relaties en attributen van het UML model. De attributen zijn niet los te zien in een UML model, niet zoals bij een ER model. Maar feitelijk gaat het in beide modellen om attributen, dus ook telbaar. Als er gekeken wordt naar de relaties van een ER model zijn deze makkelijk te ontdekken door de zwarte lijnen, hetzelfde geldt ook voor UML modellen. Waar ook de zwarte lijnen worden gebruikt om relaties tussen modelementen aan te geven. Alleen de kardinaliteit waarbij Poels een onderscheid maakt is een klein verschil te ontdekken in het UML model. In een ER model is de kardinaliteit aangegeven voor $1..n$ (één of meerdere) bij UML is dit $1..*$.

Tijdens het toepassen van de metrieke kunnen ook de modelementen voor problemen zorgen. Want bij UML modellen worden minder modelementen gebruikt ten opzichte van een ER model. Als een persoon die niet veel verstand heeft van UML modellen op zoek is naar bepaalde modelementen, kan het zijn dat er onnodig tijd



Figuur 9.2: Entiteit / Klasse

word gespendeerd. Door het zoeken van modelementen die in de metriek worden beschreven maar er niet zijn. Dit kan ook voor verwarring zorgen. Er moet van tevoren worden bekeken met een expert welke metrieken wel en niet toepasbaar zijn voordat men de ER metrieken gaat toepassen op een UML model.

Deel IV

Conclusie

U heeft in deze scriptie kunnen lezen over metrieken die ontwikkeld zijn door Moody [23], Maes [21] en Poels [31]. Met de vraag of deze metrieken toepasbaar zijn op een UML model. In het eerste deel van deze conclusie zal aanbod komen welke metrieken het beste of het minst bruikbaar zijn tijdens het evalueren van conceptuele modellen. Deze algemene conclusie zal beginnen met de metrieken van Moody aangezien hij de eerste was die concrete metrieken heeft beschreven in zijn onderzoek. Daarna zullen de metrieken van Maes worden behandeld om te eindigen met die van Poels.

In het tweede deel van deze conclusie zal ik de hoofdvraag van deze scriptie beantwoorden, namelijk de vraag *"Kunnen de modelmetrieken die zijn ontwikkeld voor ER-modellen worden toegepast op UML-modellen?"*.

Algemene conclusie

In de algemene conclusie komen de metrieken nogmaals aan bod. Maar vanuit het standpunt welke metrieken het beste te gebruiken zijn en welke het minst bruikbaar.

Om de metrieken van Moody te gebruiken is er veel informatie nodig. Deze informatie bestaat bijvoorbeeld uit procesmodellen, business rules, normaalvormen van een database en ontwikkelbudget. Hoewel je bij Maes ook procesmodellen nodig hebt, ben je al verder in het ontwikkelproces voordat je alle informatie bij elkaar hebt verzameld. Terwijl je juist snel een evaluatie wilt hebben van een conceptueelmodel tijdens het ontwikkelen. Ik denk daarom dat de metrieken die Moody heeft ontwikkeld pas achteraf nadat het conceptueelmodel ontwikkeld is worden toegepast.

Ook zijn de metrieken van Moody (bijna) allemaal beschrijvend van aard. Hierdoor word het onduidelijk wat de metrieken voor een uitkomst hebben en hoe je de metrieken precies moet gaan toepassen.

De metrieken van Maes zijn vergelijkbaar met die van Moody. Maes heeft in tegenstelling tot Moody geprobeerd om metrieken te ontwikkelen voor conceptuele modellen op zich. De metrieken van Maes zijn zo algemeen geworden omdat ze alle conceptuele modellen moeten beslaan, dat de metrieken niet meer bruikbaar zijn. Metrieken horen volgens mij concreet te zijn met een eenduidig antwoord. Bij bepaalde metrieken van Maes kan je simpel weg een *ja* of *nee* antwoorden. Maar met een simpele ja of nee als antwoord kan het conceptueelmodel niet in kwaliteit verhoogd worden, die je juist in eerste instantie wilt gaan verbeteren met het gebruiken van metrieken.

Maes heeft een goed punt meegenomen in haar metrieken en dat is User satisfaction. Hiermee grijpt ze terug naar de evaluatie van software-engineering projecten. Door de gebruikers erbij te betrekken krijg je een beter beeld of het uiteindelijke systeem wel voldoet aan het te ondersteunen proces. Mochten gebruikers niet met het conceptueelmodel kunnen werken, dan kunnen ze dat ook zeker niet met het uiteindelijke informatiesysteem. Het enige minpuntje aan deze manier van evalueren is dat de gebruikers wel moeten weten hoe ze conceptuele modellen moeten lezen en / of toepassen.

De metrieken die Poels heeft ontwikkeld zijn veruit het makkelijkst toe te passen, het is namelijk een kwestie van goed kunnen tellen. Deze metrieken zijn ook uiterst geschikt als basis om programma's en tools te ontwikkelen die automatisch ER modellen controleren. Hoewel het vrij gemakkelijk is om deze metrieken toe te passen zul je toch kennis moeten hebben van ER modellen. Zonder deze kennis zal je bijna tot niet kunnen achterhalen welke elementen Poels bedoelt, ondanks dat er in de

literatuur een legenda bij staat.

Wat ook een groot voordeel is van de metrieken die Poels heeft ontwikkeld, is de uitkomst die vast staat. Het is namelijk altijd een getal, je kunt dit (bijna) niet fout doen. Daarom denk ik dat deze metrieken uiterst geschikt zijn om te gebruiken. Maar er moet wel gekeken worden naar welke metrieken je nodig hebt voor welke klus.

Ondanks dat de metrieken van Moody en Maes ten opzichte van Poels langzaam en veel informatie vergen. Denk ik dat het het beste is om de metrieken van Moody, Maes en Poels te combineren. Door te combineren van de metrieken kijk je op twee verschillende manieren naar een conceptueelmodel. Een voorbeeld van een mogelijke combinatie zou zijn de metriek die Poels heeft beschreven over het aantal entiteitentypen in een conceptueelmodel en combinatie met het oordeel van een ontwikkelaar van Moody. De ontwikkelaar kan zien of het aantal entiteitentypen lastig is om te verwerken in een programma, te integreren met een bestaand systeem of veel tijd gaat kosten om uiteindelijk het informatiesysteem te ontwikkelen.

Als de metrieken los van elkaar worden gebruikt ga je je bijvoorbeeld afvragen waarom je bepaalde entiteitentypen hebt gemodelleerd. Terwijl een simpel getal geen informatie bevat over hoe je moet gaan nadenken over bepaalde entiteitentypen.

Hoofdstuk 10

Beantwoording hoofdvraag

In dit hoofdstuk word er antwoord gegeven op de onderzoeksvraag:

Kunnen de modelmetriecken die zijn ontwikkeld voor ER-modellen worden toegepast op UML-modellen?.

Na het beantwoorden van de vijf deelvragen, zie hoofdstuk 6 voor deelvraag één en twee, hoofdstuk 8 voor deelvraag drie en hoofdstuk 9 voor deelvraag vier en vijf. Het antwoord is gebaseerd op bevindingen van modelmetriecken die toegepast zijn op ER- en UML modellen. Daarbij zijn de kwaliteitsaspecten van Moody de leidraad van deze scriptie geweest. Het antwoord dat nog volgt op de hoofdvraag is verworven uit de antwoorden van de deelvragen. Deze antwoorden zullen hier kort toegelicht worden om een duidelijk beeld te geven hoe het antwoord op de hoofdvraag tot stand is gekomen.

Deelvraag 1: Welke metriecken zijn er door verschillende auteurs ontwikkeld?

In deze scriptie zijn de metriecken van drie onderzoekers behandeld. Ik zal ze in het kort samenvatten en opsommen:

- De metriecken van Moody [23] zijn op basis van acht kwaliteitseigenschappen. Moody heeft aan de hand van deze kwaliteitseigenschappen de informele metriecken ontwikkeld. Het idee wat bij Moody word gecreëerd is dat je nog een keer gaat nadenken over wat je gemodelleerd hebt.
- De metriecken van Maes [21] zijn net zoals die van Moody informeel. Alleen de manier waarop Maes ze heeft ontwikkeld, voor conceptuele modellen in het algemeen. Is het idee misschien te ver door geschoten, want de metriecken zijn hierdoor algemeen geworden en veel al onbruikbaar. Dit komt door te algemene definities waar alle conceptuele modellen onder moeten vallen.
- Poels [31] is de vreemde eend in de bijt tussen deze auteurs, hij is de enige die informele metriecken heeft ontwikkeld. Door metriecken te ontwikkelen waarvan de uitkomst een getal is lijkt het erg makkelijk om toe te passen. Maar later wordt weinig gezegd over deze getallen, laat staan dat je het model kunt verbeteren aan de hand van deze getallen. Hierbij heeft Poels volgens mij de plank misgeslagen.

In het hoofdstuk onderzoek tot nu toe is er een tabel te vinden van nog meer onderzoekers die metriecken hebben ontwikkeld. Hier zal ik later in deze conclusie nog op terug komen.

Deelvraag 2: Voor welke doeleinden zijn de metrieken ontwikkeld?

De metrieken van Moody zijn voor de volgende kwaliteitseigenschappen ontwikkeld:

- **Compleetheid**
Alle eisen dienen in het conceptueelmodel terug te komen.
- **Integriteit**
De mate waarin de business rules van toepassing zijn op het conceptueelmodel.
- **Flexibiliteit**
Het gemak waarmee een conceptueelmodel aangepast kan worden op de wensen van de markt en gebruikers.
- **Begrijpbaarheid**
De mate waarin stakeholders het conceptueelmodel kunnen begrijpen.
- **Correctheid**
Het conceptueelmodel moet voldoen aan alle voorschriften en conventies.
- **Eenvoud**
In het conceptueelmodel dienen er zo min mogelijk entiteitstypen te worden gebruikt om alle eisen van de stakeholders op te nemen.
- **Integratie**
De mate waarin het conceptueelmodel voldoet aan de organisatie data.
- **Implementeerbaarheid**
De mate waarop het systeem ontwikkeld kan worden. Rekening houdend met tijd, kosten en technologische constraints.

De aspecten die Moody in zijn onderzoek gebruikt zijn naar mijn mening precies aspecten die een conceptueelmodel heeft. Je wilt dat een model compleet en correct is voordat je het gaat implementeren en integreren, het dient eenvoudig en begrijpbaar te zijn voor de ontwikkelaars. Daarbij dient het model integer te zijn door juist om te gaan met gegevens en het model dient flexibel te zijn in het te ondersteunen proces.

Maes kijkt naar hele andere aspecten binnen het conceptueelmodel:

- **Perceived Ease of Understanding**
De mate van makkelijkheid waarin het conceptueelmodel te gebruiken is.
- **Perceived Usefulness**
De mate waarin het conceptueelmodel het proces ondersteunt.
- **Perceived Semantic Quality**
De mate waarin het conceptueelmodel de informatie aanbied die nodig is voor het te ondersteunen proces.
- **User Satisfaction**
De manier waarop de eisen van een stakeholder terug komen in het conceptueelmodel.

Bij deze metrieken had ik het idee dat ze hetzelfde beschrijven als de aspecten die Moody gebruikt. Perceived ease of understanding is een andere beschrijving voor begrijpbaarheid en eenvoud van een model. Perceived Usefulness is vergelijkbaar met compleetheid en implementeerbaarheid, omdat je hier gaat kijken hoe nuttig

het model daadwerkelijk is. Perceived Semantic Quality is verwoven met de correctheid van een conceptueelmodel. De semantiek van een conceptueelmodel wordt correct geacht. User satisfaction kan ook wel correctheid en compleetheid worden genoemd. Uiteindelijk dient het informatiesysteem de verwachtingen van een klant waar te maken.

Deelvraag 3: Wat zijn de overeenkomsten en verschillen tussen de verschillende metrieken?

Aan het einde van hoofdstuk 8 is een tabel te zien met de verschillen en overeenkomsten van de metrieken tussen de onderzoekers. Het antwoord op deze vraag wordt er gekeken naar de kwaliteitseigenschappen onderling. De specifieke verschillen en overeenkomsten tussen de metrieken zijn te vinden in hoofdstuk 8.

De kwaliteitseigenschap eenvoud maakt gebruik van de begrijpbaarheid. Moody stelt namelijk hoe eenvoudiger het conceptueelmodel is, hoe beter deze te begrijpen is. Ik kom tot deze conclusie op basis van bevindingen van de metrieken die toegepast zijn op de ER modellen. Als men kijkt naar het eerste ER model (zie afbeelding 7.1 hoofdstuk 7), bestaat deze uit 44 constructen. Ten opzichte van het ER model gemaakt door Marc Pieterse (zie afbeelding 13.1 appendix B) die heeft maar 27 constructen. Hieruit kan je opmaken dat eenvoud de begrijpbaarheid verhoogd. Relatief zijn dit de makkelijkste aspecten om te controleren binnen een conceptueelmodel. Doordat je een duidelijk idee hebt wat deze aspecten inhouden.

Begrijpbaarheid op zichzelf heeft een uitwerking op de flexibiliteit, integratie en implementeerbaarheid van het conceptueelmodel. Als het model verkeerd geïnterpreteerd wordt dan kan het model niet op juiste manier geïmplementeerd worden. Dit geldt hetzelfde voor de integratie met andere systemen. Als de begrijpbaarheid van een model (bijv. door teveel constructen) slecht is dan kan het conceptueelmodel ook moeilijker geïntegreerd worden omdat men niet weet aan welke modelelementen het bestaande systeem gekoppeld dient te worden. Begrijpbaarheid is dus de basis van integratie, die weer nauw is verbonden met de flexibiliteit. Tijdens het integreren van het bestaande systeem in het conceptueelmodel kan naar voren komen dat het model niet flexibel genoeg is. Bijvoorbeeld doordat het model niet mee kan veranderen zodat het bestaande systeem een goede aansluiting heeft bij het te ondersteunen proces. Deze aspecten zijn al wat lastiger om te controleren. Hoe goed het conceptueelmodel te integreren is binnen het bestaande systeem is een kwestie van uitvoeren, of het model dan flexibel is komt vanzelf naar voren. Implementeren is alleen te controleren om daadwerkelijk het systeem te ontwikkelen.

Compleetheid, correctheid en integriteit zijn vergeleken met de andere kwaliteitsaspecten een ander soort kwaliteitsaspect. Bij deze aspecten worden gekeken of de eisen terug in het model voorkomen, of er aan de conventies worden gehouden of het model aansluit bij de wensen van het bedrijf. Op correctheid na zijn deze overige twee aspecten moeilijk om na te gaan. Bij compleetheid weet je nooit of alle eisen aanwezig zijn in het conceptueelmodel. Integriteit is ook moeilijk omdat je dit pas na het implementeren pas kunt zeggen of het systeem aansluit bij het bedrijf. Correctheid is het makkelijkste aspect van allemaal, aangezien dit met een checklist nagelopen kan worden of bepaalde conventies niet worden nagekomen.

Deelvraag 4: Zijn de ER metrieken toepasbaar op een UML-model?

De ER metrieken zijn toepasbaar op een UML model. Alleen moet er goed gekeken worden naar welke metrieken er wordt gebruikt en/of deze geen ER conventies controleren. Vooral de informele metrieken zijn makkelijker toe te passen dan formele metrieken. Feitelijk omdat je opnieuw aan het denken wordt gezet over het con-

ceptueelmodel.

Deelvraag 5: Welke problemen kunnen optreden tijdens het toepassen van ER metriecken op UML-modellen?

Zoals hierboven al gezegd werd zijn in het bijzonder de ER conventies een probleem bij het toepassen van ER metriecken op een UML model. Modelelementen kunnen ook voor problemen zorgen doordat er andere modelelementen worden gebruikt.

Aan de hand van de bovenstaande antwoorden op de deelvragen kan het antwoord worden gegeven op de onderzoeksvraag van deze scriptie. Het antwoord op de onderzoeksvraag luidt als volgt:

ER metriecken kunnen worden toegepast op UML modellen in achtnemend dat het voornamelijk informele metriecken betreft. De basis van formele metriecken zijn ER conventies deze conventies komen niet terug komen in een UML model, daarom zijn deze formele metriecken minder geschikt om toegepast worden.

10.1 Aanbeveling

In de conclusie heeft u kunnen lezen dat de ER metriecken toegepast kunnen worden op UML modellen. De ER metriecken die ik zou gebruiken zijn voornamelijk de informele metriecken die ontwikkeld zijn door Maes [21] en Moody [23]. De aanbeveling voor informele metriecken baseer ik op de toepassing van de metriecken. De informele metriecken zijn volgens mij op elk conceptueelmodel toe te passen, zolang dit conceptueelmodel maar uit modelelementen bestaat. De informele metriecken verschaffen de ontwikkelaar van het conceptueelmodel een manier om over zijn of haar model na te denken. Doordat er op verschillende manieren naar het conceptueelmodel wordt gekeken kan dit wellicht zorgen voor een kwaliteit verbeteren van het conceptueelmodel.

Voor ER modellen zou ik een combinatie van zowel de informele als de formele metriecken aanraden. Formele metriecken om te zien hoeveel modelelementen er worden gebruikt in het ER model. Hierdoor zou je ervoor kunnen kiezen om het aantal modelelementen in het ER model terug te dringen. De informele metriecken zou ik voornamelijk de metriecken gebruiken die betrekking hebben op het procesmodel. In het procesmodel bevinden zich alle activiteiten die door het informatiesysteem ondersteund dienen te worden. Door een vergelijking te trekken tussen het procesmodel en het ER model kunnen er al "fouten" opgespoord worden.

10.2 Persoonlijke evaluatie

Het laten ontwikkelen van modellen door Marc Pieterse heeft goed uitgepakt. Omdat ik al een ruime tijd bezig was met het lezen van literatuur over de modelmetriecken was ik al beïnvloed voordat ik begon met modelleren. Doordat Marc geen modelmetriecken heeft toegepast en een ander beeld van het voorgestelde probleemdomain heeft, heb ik een beter idee gekregen hoe ik de metriecken moest vergelijken en moest toepassen. Met Marc's zijn ontwikkelde model werd ik ook niet beïnvloed door mijn manier van modelleren en kon ik het model objectief beoordelen. Uiteindelijk heb ik ook gekozen om zelf geen UML model te ontwerpen, maar het model van Marc te gebruiken. Ik heb hiervoor gekozen omdat ik de modelmetriecken al twee keer toegepast heb op de ER modellen. Ik zou geneigd zijn het UML model zo goed mogelijk te on-

twikkelen om de gewenste uitkomst te krijgen.

Op aanbeveling van Patrick heb ik een casus gemaakt om de metriecken op toe te passen. Door een eigen casus te ontwerpen heb ik een beter beeld gekregen hoe de metriecken daadwerkelijk toe te passen zijn. Ook ben in de metriecken als geheel binnen conceptual modelling beter gaan begrijpen.

De kwaliteitseigenschappen die Moody heeft gebruikt kon ik mij het meest in vinden. Doordat je er gelijk een beeld bij kan vormen wat welke kwaliteitseigenschap precies inhoud. Deze kwaliteitseigenschappen heb ik gebruikt om de metriecken van de andere onderzoekers onder te clusteren. Dit leverde een probleem op bij de metriecken van Poels, hierbij kon niet gelijk een link vinden tussen de kwaliteitseigenschappen van Moody en de formele metriecken van Poels. Doordat je gelijk een beeld creëert van de eigenschap laat je, je ook gemakkelijk leiden door een uitkomst te forceren die past bij de betreffende eigenschap. Achteraf had ik misschien zelf een uiteenzetting moeten maken tussen formele en informele metriecken. Dan had ik deze problemen wellicht kunnen vermijden en hierdoor me zelf niet laten beïnvloeden door de eigenschappen die Moody gebruikt.

Wat overigens ook geen goede zet was is om alle metriecken van drie onderzoekers te gebruiken. Het onderzoek zou makkelijker verlopen zijn als ik het belangrijkste deel van deze metriecken had gebruikt. Sommige metriecken kon ik niet uitvoeren en waren ook overbodig om te beschrijven. Dit komt denk ik doordat ik te laat ben begonnen met het maken van de casus. Als ik eerder was begonnen met de casus had ik een deel van de metriecken weg kunnen laten omdat ze op deze casus niet toe te passen waren. Daarbij was het ook moeilijk door de hoeveelheid metriecken, om verschillen en overeenkomsten te beschrijven.

Het moeilijkste van mijn scriptie is het schrijven van een conclusie. De vragen wat ga ik allemaal schrijven, hoe moet ik het beschrijven en wat zou de lezer allemaal verwachten hielden me veel bezig. Door de hoeveelheid informatie die verworven is zag ik hierdoor niet hoe ik dit het beste kon aanpakken. Met hulp van het Academisch Schrijfcentrum ben ik hier een heel eind uit gekomen.

Ondanks alle goede en verkeerde keuzes, heb ik deze periode als zeer leerzaam ervaren. Voornamelijk de manier en wat er allemaal bij komt kijken om onderzoek te doen op wetenschappelijk niveau. Ik kijk ook met een goed gevoel terug op mijn scriptie, die ik voor mijn gevoel in een goed tempo heb weten te voltooien.

Hoofdstuk 11

Vervolgonderzoek

Binnen dit onderzoek is er alleen gekeken naar de modelmetrieken van de Moody, Maes en Poels en de bevindingen hiervan. Het is mogelijk dat andere modelmetrieken die niet aanbod zijn gekomen in deze scriptie, wel oplossingen aandragen of makkelijker zijn om toe te passen op conceptuele modellen.

Naar aanleiding van dit onderzoek zou men kunnen kijken naar de oplossingen van de onderzochte metrieken. De metrieken zijn in dit onderzoek drie keer toegepast twee keer op twee verschillende ER modellen en één keer op een UML model. Aan de hand van de uitkomsten (zie appendix A) zijn er eigenlijk geen oplossingen aangedragen om het conceptueelmodel te verbeteren. Uit de uitkomst van de toegepaste metriek kan je wel aannames opmaken. Maar zijn deze aannames dan ook beter voor de kwaliteit van het conceptueelmodel, dat is nog maar af te vragen. Bij het toepassen van metriek 7: het aantal veranderende elementen op het UML model, geeft als uitkomst twee entiteiten die veranderd kunnen worden. Maar er word ergens gezegd dat deze entiteiten gelijk aangepast dienen te worden of dat het de bedoeling is dat deze entiteiten pas als ze echt veranderen aan te passen. Het lijkt me logisch om gelijk de mogelijke veranderingen mee te nemen tijdens de ontwikkeling, zodat er minder kosten hoeven gemaakt te worden mochten er veranderingen optreden.

Tevens mogen de metrieken zelf ook beter ontwikkeld worden. Zodat de metrieken eenduidig geïnterpreteerd kunnen worden. Dit is om te voorkomen dat elke keer als je een metriek toepast op hetzelfde conceptueelmodel de uitkomst veranderd. Het is ook wenselijk dat de uitkomst hetzelfde blijft ondanks dat iemand anders de metrieken toepast. De metrieken zijn nu nog te vaag om ook maar goed te kunnen gebruiken, laat staan dat ze de kwaliteit van het conceptueelmodel kunnen verbeteren.

Metrieken die de begripbaarheid of eenvoud kunnen berekenen op basis van paden binnen een conceptueelmodel zijn nog niet in de literatuur gevonden of ontwikkeld. Een pad in een conceptueelmodel kan een relatie zijn tussen entiteiten. Op basis van het bewandelen van deze paden zou er iets gezegd kunnen worden over het complexe geheel van conceptueelmodel. Bijvoorbeeld hoe meer paden binnen een conceptueelmodel één keer bewandeld kunnen worden des te groter en complexer het model. Poels heeft hier een soort gelijk onderzoek naar gedaan, ik zou dit onderzoek verder uitbreiden. Aangezien dit een goede manier is om de begripbaarheid van een conceptueelmodel te evalueren.

Formele metrieken op basis van predicaten logica net zoals Eick [7] heeft ontwikkeld, zouden meer in aanmerking komen voor de ontwikkeling van geautomatiseerde model checkers. Het voordeel van deze metrieken zijn volgens mij de eenduidige antwoor-

den die ze generen. Hoewel ik denk dat deze metrieken voor een groot deel van de modellerende mensen niet te gebruiken zijn.

Bovenal denk ik net zoals Moody [22] dat het belangrijk is dat er word gekeken naar een standaard framework voor conceptuele modellen. Zonder een standaard krijg je metrieken die nergens overgaan, moeilijk zijn toe te passen, niet eenduidig zijn en niet met oplossingen komen om de kwaliteit van conceptuele modellen te verbeteren. Het is daarom belangrijk dat er word gekeken naar een standaard voor zowel formele als informele metrieken. Anders zullen de onderzoekers maar metrieken blijven ontwikkelen wat uiteindelijk niet in de praktijk gebruikt gaat worden. Kijk maar naar het onderzoek van Poels [31] die heeft bekeken welke metrieken in de praktijk toegepast of gevalideerd zijn. Het blijkt dus dat van de zes auteurs die in het onderzoek van Poels navoren komen, vijf niet gevalideerd of toegepast zijn.

Deel V

Appendix

Hoofdstuk 12

Appendix - A - Resultaten

12.1 Resultaten: toepassing metrieken op de ER modellen

Compleetheid

MoMetriek 1: Niet overeenkomende modelelementen

Model 1 - geen modelelementen die niet overeenkwamen gevonden.

Model 2 - geen modelelementen die niet overeenkwamen gevonden.

MoMetriek 2: Ontbrekende eisen

Model 1 - Bij het reisbureau zijn er ontbrekende eisen:

- Opgeven van paspoortgegevens (plaats uitgifte, datum uitgifte, geldigheidsduur, etc);
- Opgeven rekeningnummer voor het incasseren van het verschuldigde bedrag;

Model 2 - Beide eisen van de touroperator ontbreken;

MoMetriek 3: Foutief gedefinieerde modelelementen

Model 1 - Niks is foutief definieert, aan de hand van de eisen komt alles terug.

Model 2 - Verblijfplaats is een modelelement die foutief gedefinieerd is.

MoMetriek 4: Inconsistenties ten opzichte van het procesmodel

Model 1 - In dit model zijn verschillende inconsistenties te ontdekken namelijk:

- Tickets zijn niet terug te vinden in het procesmodel, terwijl deze wel in het ER model aanwezig zijn;
- De reisverzekeringsspolis is ook niet terug te vinden in het model. Ook de reisverzekering is niet terug te vinden in het procesmodel;

Model 2 - In dit model is er één model inconsistentie te vinden namelijk reisbureau is wel opgenomen in het procesmodel maar niet in het ER model.

MaMetriek 9: Realistische spiegeling van het bedrijfsproces

Model 1 - Het boeken van een reis is realistisch verantwoord in het procesmodel. Alleen is de mogelijkheid tot een reisverzekering af te sluiten niet mogelijk. Terwijl dit in het ER model wel zou moeten kunnen.

Model 2 - Het boeken van een reis is realistischer beeld dan het procesmodel van

model 1. Maar er gaan alleen maar uitgaande pijlen vanuit het reisbureau naar de andere entiteiten. Terwijl je zou verwachten dat het reisbureau ook informatie terug krijgt om deze weer door te geven aan de reiziger.

MaMetriek 11: Zijn alle elementen relevant

Model 1 - Niet alle elementen zijn relevant namelijk, vliegtuigmaatschappij, reisbureau en reisverzekeringsmaatschappij. Deze elementen voegen weinig toe aan het ER model, want dit zullen bijvoorbeeld geen velden worden in een database. Vliegmaatschappij zal een attribuut zijn van een ticket. Hetzelfde geldt voor de reisverzekeringsmaatschappij.

Model 2 - Ook bij dit model is naar mijn mening het element vliegmaatschappij overbodig en kan als attribuut worden opgenomen bij vliegticket.

MaMetriek 13: Informatiebehoeften

Model 1 - Dit model voldoet niet aan de informatiebehoeften. Bij de toepassing van metriek 2 zijn er eisen niet opgenomen die wel nodig zijn voor de informatie van het reisbureau of vliegmaatschappij.

Model 2 - Dit model voldoet ook aan de informatiebehoeften aan de hand van de opgestelde eisen.

PoMetriek 1: Het aantal entiteiten

Model 1 - 17 entiteiten zijn terug te vinden in het model.

Entiteiten = [Ticket, Eticket, Tickets, Reis, Vliegtuigmaatschappij, Reisbureau, Tour-operator, Woonruimte, Hotelkamer, Appartement, Voertuigen, Fiets, Auto, Scooter, Reiziger, Polis, Reisverzekeringsmaatschappij]

Model 2 - 8 entiteiten zijn terug te vinden in het model.

Entiteiten = [Verblijfplaats, Reis, Vliegticket, Vliegmaatschappij, Vervoer, Verhuurmaatschappij, Reiziger, Reisverzekering]

PoMetriek 12: Het aantal afgeleide entiteitstypen

Model 1 - 3 IsA relaties

IsA relaties = [IsA (ticket, eticket), IsA (Hotelkamer, appartement), IsA (fiets, scooter, auto)]

Model 2 - Geen.

Integeriteit**MoMetriek 5: Niet afgedwongen business rules**

Model 1 - Er zijn geen business rules gedefinieerd.

Model 2 - Er zijn geen business rules gedefinieerd.

MoMetriek 6: Niet overeenkomende business rules

Model 1 - Er zijn geen business rules gedefinieerd.

Model 2 - Er zijn geen business rules gedefinieerd.

MaMetriek 10: Tegenstrijdige elementen

Model 1 - Bevat geen tegenstrijdige elementen.

Model 2 - Bevat geen tegenstrijdige elementen.

MaMetriek 14: Niet efficiënt verstrekken van informatie

Model 1 - Dit model is niet efficiënt in het verstrekken van informatie. Doordat vliegtuigmaatschappij en verzekeringsmaatschappij in het model staan die eigenlijk geen

betrekking hebben op de informatie binnen het reisbureau is het niet efficiënt. Daarbij is het alleen mogelijk via de touroperator een woonruimte of een voertuig te boeken.

Model 2 - Dit model is efficiënt in het verstrekken van informatie.

MaMetriek 15: Efficiënt verstrekken van informatie

Model 1 - Informatie over de woonruimtes en voertuigen die een touroperator aanbied is efficiënt te zien. Daarbij de informatie die een reisbureau nodig heeft van een reiziger is allemaal terug te vinden in de attributen van de reiziger.

Model 2 - Dit hele model beschrijft precies de informatie die je nodig hebt van een reiziger bij het boeken van een reis. Want een reis bevat een verblijfplaats, soms een ticket. Daarbij kan de reiziger kiezen om een reisverzekering af te sluiten en vervoer thuis te regelen of via een verhuurmaatschappij.

Flexibiliteit

MoMetriek 7: Het aantal veranderende elementen

Model 1 - Entiteit de reiziger zal in de toekomst kunnen veranderen. Door bijvoorbeeld het toevoegen van een fotoboek op de website, of een reisblog of een emailadres. Dit kan resulteren in extra velden in de database. Wellicht zal het element reisbureau, vliegtuigmaatschappij en reisverzekeringsmaatschappij verdwijnen omdat ze niks toevoegen aan het model.

Model 2 - De entiteit reis zal het meest onderhevig zijn aan verandering. Omdat in dit model de entiteit reis centraal staat zijn alle wijzigingen ook van invloed op deze entiteit. Wijzigingen die kunnen optreden zijn het toevoegen van verblijfplaats als een attribuut.

MoMetriek 8: Geschatte kosten van veranderingen

Model 1 - Niet van toepassing.

Model 2 - Niet van toepassing.

MoMetriek 9: Strategisch belang van wijzigingen

Model 1 - Het belang van het reisbureau, vliegtuigmaatschappij en reisverzekeringsmaatschappij is van strategisch belang omdat het opnemen van deze entiteiten niet noodzakelijk van toepassing zijn voor het boeken van een reis.

Model 2 - De entiteit reis is het belangrijkste binnen dit model, door bijvoorbeeld verblijfplaats als een attribuut op te nemen kan het ervoor zorgen dat er een entiteit verdwijnt in het model. Verblijfplaats zal dan een afgeleid attribuut worden. Hierdoor zou het model makkelijker te begrijpen en te ontwikkelen zijn.

Begrijpbaarheid

MoMetriek 10: Beoordeling door stakeholders

Model 1 - Als ik een stakeholder zou zijn van het reisbureau zou ik dit model een 5 geven. Het is eigenlijk een onoverzichtelijk model om goed te kunnen begrijpen wat nu precies de bedoeling is. Welke informatie het model bevat, wat de bedoeling is van de opgegeven entiteiten en hoe het dit model het boekingsproces gaat ondersteunen.

Model 2 - Ondanks dat niet alle eisen zijn opgenomen (zie resultaten metriek 2) in het conceptueelmodel zou ik dit model een 7 geven. Het model is makkelijker te lezen omdat er minder entiteiten en relaties zijn.

MoMetriek 11: Interpretatie stakeholder

Model 1 - Niet te gedaan omdat er geen business scenario's aanwezig zijn en/of mo-

gelijke populaties.

Model 2 - Niet te gedaan omdat er geen business scenario's aanwezig zijn en/of mogelijke populaties.

MoMetriek 12: Oordeel ontwikkelaars

Model 1 - Als ik ontwikkelaar zou zijn, dan zou dit model moeilijk te ontwikkelen zijn. Doordat het niet duidelijk is wat er nu gebeuren moet met de entiteiten vliegtuigmaatschappij en reisverzekeringsmaatschappij. Dient nu een tabel te worden in de database of word dit een attribuut bij het vliegticket of polis.

Model 2 - Dit model zou meer in aanmerking komen om ontwikkeld te worden. Omdat de opgegeven entiteiten al meer corresponderen met tabellen in een database. Hoewel het model nog simplistischer kan worden gemaakt.

MaMetriek 1: Makkelijk te begrijpen

Model 1 - Wat precies de bedoeling is van de gemodelleerde entiteiten reisbureau, vliegtuigmaatschappij en reisverzekeringsmaatschappij, komt niet helemaal tot uitdrukking in dit model. Hierdoor word het model moeilijker te begrijpen doordat je als het ware vastloopt bij het lezen van dit model.

Model 2 - Dit model is makkelijker te begrijpen dan Model 1. Doordat er onderling minder relaties zijn.

MaMetriek 4: Lezen is makkelijk

Model 1 - Het lezen van dit model is vergeleken met Model 2 wat moeilijker. Vooral het stuk rondom het reisbureau is een onoverzichtelijk. Hierdoor kan de lezer verwilderd raken door alle relaties en attributen.

Model 2 - Het lezen van dit model is makkelijker vergeleken met Model 1. Doordat er onderling minder relaties tussen de entiteiten aanwezig is het geheel beter te begrijpen.

MaMetriek 6: Helpt bij het begrijpen

Model 1 - Dit model helpt niet bij het begrijpen van het boekingsproces van een reis. Hoewel de entiteiten vliegtuigmaatschappij en reisverzekeringsmaatschappij terug komen in het bedrijfsproces, is het niet duidelijk waarom ze terug moeten komen in het informatiesysteem van de reisbureau.

Model 2 - Het model helpt bij het begrijpen van de achterliggende structuur bij het boeken van een reis. Maar het is niet helemaal duidelijk hoe het proces nu precies afgewikkeld moet worden. Of welke informatie nu belangrijk is voor het boeken van een reis.

MaMetriek 7: Verbeterd de prestatie om te begrijpen

Model 1 - Deze uitkomst geldt hetzelfde als voor metriek 31. Het begrijpen word niet verbeterd, het roept zelfs meer vragen op waarom bepaalde entiteiten zijn opgenomen in dit model.

Model 2 - Door het gemodelleerde procesmodel is dit model beter te begrijpen. In ieder geval is model 2 beter begrijpen.

PoMetriek 6: Het aantal relaties

Model 1 - 11 relaties

Relaties = [Isa 3x, Biedt aan 2x, heeft afspraak, werkt samen met, beschikt over, verhuurt, heeft afspraak, biedt aan, bevat 2x, boekt]

Model 2 - 5 relaties

Relaties = [Heeft, boekt, bevat, biedt aan, verhuurt, voorkeur]

PoMetriek 7: Het aantal M:N relaties

Model 1 - Geen.

Model 2 - Geen.

PoMetriek 8: Het aantal 1:N relaties

Model 1 - 7 1:N relaties

1:N relaties = [Heeft afspraak 2x, Beschikt over, biedt aan 2x, Werkt samen met, Boekt]

Model 2 - 4 1:N relaties

1:N relaties = [Biedt aan, verhuurt, boekt, voorkeur]

PoMetriek 9: Het aantal N-ary relaties

Model 1 - Geen.

Model 2 - Geen.

PoMetriek 10: Het aantal binaire relaties

Model 1 - 11 Binaire relaties

Binaire relaties = [Isa 3x, Biedt aan 2x, heeft afspraak, werkt samen met, beschikt over, verhuurt, heeft afspraak, biedt aan, bevat 2x, boekt]

Model 2 - 5 Binaire relaties Binaire relaties = [Heeft, boekt, bevat, biedt aan, verhuurt, voorkeur]

PoMetriek 11: Het aantal reflexieve relaties

Model 1 - Geen.

Model 2 - Geen.

Correctheid**MoMetriek 13: Conventionele overtredingen**

Model 1 - Geen conventionele overtredingen gevonden.

Model 2 - Geen conventionele overtredingen gevonden.

MoMetriek 14: Normaalvorm fouten

Model 1 - Niet van toepassing.

Model 2 - Niet van toepassing.

MoMetriek 15: Redundantie

Model 1 - De redundantie die ik zou terug vinden in het dit model is vliegtuig-maatschappij met tickets en de entiteiten ticket en eticket. Dit geldt ook voor reisverzekering en polis.

Model 2 - Geen redundantie gevonden.

MaMetriek 8: Bedrijfsproces komt terug

Model 1 - Het procesmodel komt voor een deel komt het terug in het ER model. Er zijn entiteiten weggelaten hierdoor is het procesmodel niet helemaal volledig (zoals reisverzekering).

Model 2 - Dit procesmodel komt beter overeen vergeleken met het procesmodel van model 1. Hoewel de informatiestroom naar het reisbureau niet helemaal voldoet.

MaMetriek 16: Tevreden over informatie verstrekking

Model 1 - De informatie die in het model aanwezig is (voornamelijk alle attributen) komt overeen met de informatie die in het bedrijfsproces nodig is.

Model 2 - De informatie die in het procesmodel aanwezig is, is niet voldoende voor een boeking van een reis. De informatie die het ER model verstrekt is voldoende.

Eenvoud

MoMetriek 16: Hoeveelheid entiteitstypen

Model 1 - 17 entiteiten zijn terug te vinden in het model.

Entiteiten = [Ticket, Eticket, Tickets, Reis, Vliegtuigmaatschappij, Reisbureau, Touroperator, Woonruimte, Hotelkamer, Appartement, Voertuigen, Fiets, Auto, Scooter, Reiziger, Polis, Reisverzekeringsmaatschappij]

Model 2 - 8 entiteiten zijn terug te vinden in het model.

Entiteiten = [Verblijfplaats, Reis, Vliegticket, Vliegmaatschappij, Vervoer, Verhuurmaatschappij, Reiziger, Reisverzekering]

MoMetriek 17: Entiteiten en hun relaties

Model 1 - 11 relaties zijn terug te vinden in dit model.

Relaties = [Isa 3x, Biedt aan 2x, heeft afspraak, werkt samen met, beschikt over, verhuurt, heeft afspraak, biedt aan, bevat 2x, boekt]

Model 2 - 5 relaties zijn terug te vinden in dit model.

Relaties = [Heeft, boekt, bevat, biedt aan, verhuurt, voorkeur]

MoMetriek 18: Het aantal constructen

Model 1 - 44 constructen (17 entiteiten, 16 attributen en 11 relaties)

Entiteiten = [Ticket, Eticket, Tickets, Reis, Vliegtuigmaatschappij, Reisbureau, Touroperator, Woonruimte, Hotelkamer, Appartement, Voertuigen, Fiets, Auto, Scooter, Reiziger, Polis, Reisverzekeringsmaatschappij]

Attributen = [Extra kosten, Ticketnr, Eticketnr, reisnr, polisnr, nawgegevens, straat, plaats, postcode, voornaam, achternaam, paspoortgegevens, afgiftedatum, geldigheid, paspoortnr, klantrnr]

Relaties = [Isa 3x, Biedt aan 2x, heeft afspraak, werkt samen met, beschikt over, verhuurt, heeft afspraak, biedt aan, bevat 2x, boekt]

Model 2 - 27 constructen (8 entiteiten, 14 attributen en 5 relaties)

Entiteiten = [Verblijfplaats, Reis, Vliegticket, Vliegmaatschappij, Vervoer, Verhuurmaatschappij, Reiziger, Reisverzekering]

Attributen = [type, adres, contactgeg, vertrekdata, aankomstdata, type vakantie, reisid, bestemming, polis, nawgegevens, reizigerid, kenteken, verhuurdata, terugbrengdata]

Relaties = [Heeft, boekt, bevat, biedt aan, verhuurt, voorkeur]

MaMetriek 2: Frustratie met gebruik

Model 1 - Dit model kan frustrerend zijn in gebruik, doordat er zoveel entiteiten, relaties en attributen zijn. Als ontwikkelaar zou je niet precies weten waar je moet beginnen.

Model 2 - Dit model is makkelijker in gebruik dan model 1. Omdat dit model 17 constructen minder heeft is het geheel overzichtelijk.

MaMetriek 3: Makkelijk in gebruik

Model 1 - Dit model is niet gemakkelijk in gebruik, ik zou kiezen voor model 2 doordat deze overzichtelijker is doordat er minder constructen in het model aanwezig zijn.

Model 2 - Dit model is makkelijker in gebruik dan model 1.

PoMetriek 2: Aantal attributen

Model 1 - 16 attributen

Attributen = [Extra kosten, Ticketnr, Eticketnr, reisinr, polisinr, nawgegevens, straat, plaats, postcode, voornaam, achternaam, paspoortgegevens, afgiftedatum, geldigheid, paspoortnr, klantnr]

Model 2 - 14 attributen

Attributen = [type, adres, contactgeg, vertrekdata, aankomstdata, type vakantie, reisid, bestemming, polis, nawgegevens, reizigerid, kenteken, verhuurdata, terugbrengdata]

PoMetriek 3: Het aantal afgeleide attributen

Model 1 - Geen.

Model 2 - Geen.

PoMetriek 4: Het aantal samengestelde attributen

Model 1 - 2 samengestelde attributen

Samengestelde attributen = [Paspoortgeg, NAW gegevens]

Model 2 - 1 samengesteld attribuut Samengesteld attribuut = [NAW]

PoMetriek 5: Het aantal meerwaardige attributen

Model 1 - Geen.

Model 2 - Geen.

Integratie**MoMetriek 19: Aantal conflicten**

Model 1 - Niet van toepassing, er is geen organisatie-model.

Model 2 - Niet van toepassing, er is geen organisatie-model.

MoMetriek 20: Het aantal data conflicten

Model 1 - Niet van toepassing, er is geen bestaand systeem.

Model 2 - Niet van toepassing, er is geen bestaand systeem.

MoMetriek 21: Het aantal dubbele data elementen

Model 1 - Niet van toepassing, er is geen bestaand systeem.

Model 2 - Niet van toepassing, er is geen bestaand systeem.

MoMetriek 22: Waardering van vertegenwoordigers van andere systemen

Model 1 - Niet van toepassing, er is geen bestaand systeem.

Model 2 - Niet van toepassing, er is geen bestaand systeem.

Implementeerbaarheid**MoMetriek 23: Schatting technische risico**

Model 1 - Dit model zal wel kunnen voldoen aan de eisen die worden gesteld. Maar zal zo complex zijn om te ontwikkelen dat het als nog een risico zal zijn om dit model in ontwikkeling te nemen. Door de complexiteit kan het aanzienlijk langer zijn om te ontwikkelen dan model 2.

Model 2 - Dit model voldoet aan de eisen van de stakeholders. En zal in verhouding tot model 1 makkelijker te ontwikkelen zijn, doordat de complexiteit een stuk minder is.

MoMetriek 24: Schatting tijdsduur

Model 1 - Het is moeilijk om een schatting te doen tot het ontwikkelen van dit model, maar het ontwikkelen zal langer duren dan het ontwikkelen van model 2.

Model 2 - Het is moeilijk om een schatting te doen tot het ontwikkelen van dit model, maar het ontwikkelen zal korter duren dan het ontwikkelen van model 1.

MoMetriek 25: Schatting ontwikkelkosten

Model 1 - Doordat dit model complexer is dan model 2, zullen de kosten om het conceptueelmodel te kunnen lezen en begrijpen hoger zijn dan de ontwikkelkosten van model 2.

Model 2 - De kosten van dit model zullen lager zijn omdat het conceptueelmodel makkelijker te begrijpen is. Er zijn ook minder onderlinge relaties waardoor het eenvoudiger programmeren zal zijn.

MaMetriek 5: Verbetering ten opzichte van tekst

Model 1 - Het procesmodel is een verbetering van de tekst, door een procesmodel te ontwikkelen denk je meer na over de informatie die je nodig hebt. Uit een tekst is het moeilijker om te achterhalen welke informatie nu nuttig is voor je informatiesysteem.

Model 2 - De beschrijving van model 1 geldt hetzelfde voor model 2.

12.2 Resultaten: toepassing metrieken op het UML Model

Note: Als er in de resultaten word verwezen naar het proces- of bedrijfsproces dan wordt afbeelding 13.2 (zie appendix B) bedoeld.

MoMetriek 1: Niet overeenkomende modelelementen

Alle modelelementen die aanwezig zijn in het UML model komen overeen met de eisen die de stakeholders hebben opgegeven.

MoMetriek 2: Ontbrekende eisen

De eisen betrekking tot de touroperator zijn niet opgenomen:

- Reis bij een touroperator boeken is niet opgenomen in dit model.
- Via een reisbureau woonruimte aanbieden
- Het aan kunnen bieden van te huren vervoer

MoMetriek 3: Foutief gedefinieerde modelelementen

In dit model zijn geen foutief gedefinieerde modelelementen.

MoMetriek 4: Inconsistenties ten opzichte van het procesmodel

Er zijn geen inconsistenties gevonden ten opzichte van het procesmodel.

MoMetriek 5: Niet afgedwongen business rules

De business rules zijn niet aanwezig in deze casus.

MoMetriek 6: Niet overeenkomende business rules

De business rules zijn niet aanwezig in deze casus.

MoMetriek 7: Het aantal veranderende elementen

De elementen die in de toekomst kunnen veranderen:

- Reiziger, bij dit element zouden extra attributen aan toegevoegd kunnen worden. Bijv. Rekeningnummer, mobiele nummer, etc. Dit zou het proces kunnen beïnvloeden, door online de reis te betalen (ideaal) of het mobiele nummer voor bevestiging van de reis voor dat er betaald moet worden.
- Reis, aan dit element kunnen extra attributen worden toegevoegd zoals fotoverslagen (door het reisbureau verzocht) of video's, etc. Dit zal dan extra ontwikkelkosten kunnen geven.

MoMetriek 8: Geschatte kosten van veranderingen

Aangezien er geen budget is gemaakt voor deze casus kunnen de kosten niet worden geschat.

MoMetriek 9: Strategisch belang van wijzigingen

De wijzigingen zoals beschreven in Metriek 7, zou het makkelijker kunnen maken voor reizigers om reis te boeken. Ook om de reiziger een indruk te kunnen geven van het land waar hij/zij naartoe wilt.

MoMetriek 10: Beoordeling door stakeholders

Er zijn vier verschillende stakeholders in deze casus. Twee (touroperator en het reisbureau) van de vier stakeholders zullen niet tevreden zijn met het conceptueelmodel. Doordat de eisen niet goed zijn opgenomen in het conceptueelmodel.

MoMetriek 11: Interpretatie stakeholder

Het interpreteren van het conceptueelmodel is niet gedaan, doordat er geen business scenario's aanwezig zijn en er kan geen populatie worden gemaakt.

MoMetriek 12: Oordeel ontwikkelaars

Als ontwikkelaar zal ik dit model een 8 geven, voor de overzichtelijkheid en de eenvoud. Maar een 6 voor de eisen die opgenomen zijn in het UML model. Achteraf eisen in een informatiesysteem bouwen kost veel tijd en geld.

MoMetriek 13: Conventionele overtredingen

In dit model zijn geen conventionele overtredingen geconstateerd.

MoMetriek 14: Normaalvorm fouten

UML modellen zijn (meestal) geen database modellen, hier kan dus geen normaalvorm fouten in zitten.

MoMetriek 15: Redundantie

Het conceptueel bevat volgens de definitie van Moody geen redundantie.

MoMetriek 16: Hoeveelheid entiteitstypen

Dit model bevat 16 entiteiten

Entiteiten = [Reis, Boeking, Reiziger, Vervoer reservatie, verblijfplaats, Verhuurmaatschappij, Autoverhuur, appartement, hotel, verblijf, eticket, vliegtickets, reisbureau, vliegmaatschappij, verzekeringsmaatschappij, verzekering]

MoMetriek 17: Entiteiten en hun relaties

Dit model bevat 6 relaties

Relaties = [verzorgt, boekt 3x, huurt, sluit af]

MoMetriek 18: Het aantal constructen

Dit model bevat 35 constructen (16 entiteiten, 6 relaties en 13 attributen)

Entiteiten = [Reis, Boeking, Reiziger, Vervoer reservatie, verblijfplaats, Verhuurmaatschappij, Autoverhuur, appartement, hotel, verblijf, eticket, vliegtickets, reisbureau, vliegmaatschappij, verzekeringsmaatschappij, verzekering]

Relaties = [verzorgt, boekt 3x, huurt, sluit af]

Attributen = [datum vertrek, datum aankomst, bestemming, min deelnemer, max deelnemer, reisvoorkeuren, van, tot, naw, reizigerid, reis, reiziger, polis]

MoMetriek 19: Aantal conflicten

Deze metriek is niet toegepast omdat er geen ander systeem is.

MoMetriek 20: Het aantal data conflicten

Deze metriek is niet toegepast omdat er geen ander systeem is.

MoMetriek 21: Het aantal dubbele data elementen

Deze metriek is niet toegepast omdat er geen ander systeem is.

MoMetriek 22: Waardering van vertegenwoordigers van andere systemen

Deze metriek is niet toegepast omdat er geen ander systeem is.

MoMetriek 23: Schatting technisch risico

Het kan wel een vergroot technisch risico hebben doordat de eisen van de touroperator niet aanwezig zijn. Hierdoor kan het zijn dat het systeem niet ingevoerd wordt omdat het reisbureau deze eisen perse nodig heeft om het boeking proces af te ronden.

MoMetriek 24: Schatting tijdsduur

De tijdsduur kan verkort worden op basis van het conceptueelmodel, door automatisch code te laten generen. Dit kan gedaan worden in Rational Rose, voor een Java applicatie. Verder is de tijdsduur niet te schatten, maar aan de grote van het conceptueelmodel te zien zal het niet veel kosten.

MoMetriek 25: Schatting ontwikkelkosten

Doordat er geen budget is opgesteld in deze casus zal deze metriek niet van toepassing zijn.

MaMetriek 1: Makkelijk te begrijpen

Aan de hand van de eisen was het conceptueelmodel makkelijk te begrijpen.

MaMetriek 2: Frustratie met gebruik

Deze metriek is niet te controleren.

MaMetriek 3: Makkelijk in gebruik

Het conceptueelmodel, is gemakkelijk in gebruik bijv. om te vergelijken met het procesmodel. Verder is het moeilijk na te gaan of het model gemakkelijk in gebruik is.

MaMetriek 4: Lezen is makkelijk

Doordat het conceptueelmodel heeft weinig relaties en entiteiten daardoor is het conceptueelmodel gemakkelijk om te lezen.

MaMetriek 5: Verbetering ten opzichte van tekst

Het conceptueelmodel is een verbetering ten opzichte van de tekst. Alle eisen zijn terug te vinden in het conceptueelmodel.

MaMetriek 6: Helpt bij het begrijpen

Het conceptueelmodel heeft geholpen om een beter beeld te verkrijgen van het gemodelleerde bedrijfsproces.

MaMetriek 7: Verbeterd de prestatie om te begrijpen

Aan de hand van metriek 31, verbeterd het conceptueelmodel de prestatie om het proces te begrijpen.

MaMetriek 8: Bedrijfsproces komt terug

Het bedrijfsproces is terug te zien in het UML model. Alle informatie die nodig is om het proces te doorlopen is aanwezig in het UML model.

MaMetriek 9: Realistische spiegeling van het bedrijfsproces

Aan de hand van Metriek 33, kan men zeggen dat het conceptueelmodel een realistische spiegeling is van het bedrijfsproces.

MaMetriek 10: Tegenstrijdige elementen

Verblijf is ten opzichte van verblijfplaats een tegenstrijdig. Doordat verblijf bijna dezelfde benaming heeft als verblijfplaats kan het verwarrend zijn wat precies de bedoeling is van verblijf. Waarschijnlijk kan dit element beter veranderd worden naar verblijfsduur.

MaMetriek 11: Zijn alle elementen relevant

Alle elementen die in het model aanwezig zijn, zijn relevant voor de totstandkoming van een reisboeking.

MaMetriek 12: Het bedrijfsproces past in conceptueelmodel

Het bedrijfsproces past in het conceptueelmodel. De elementen die worden gebruikt tussen de informatiestromen komen overeen met die in het UML model.

MaMetriek 13: Informatiebehoeften

Aan de hand van het procesmodel dan komen de informatiebehoeften overeen.

MaMetriek 14: Niet efficiënt verstrekken van informatie

Bij dit UML model is het moeilijk na te gaan of het model (niet) efficiënt is in het verstrekken van informatie. De informatie die wordt verstrekt komt overeen met het procesmodel maar of deze efficiënt worden verstrekt is niet te zeggen. Omdat de definitie van efficiënt niet gedefinieerd is. Efficiënt in snelheid? Overzichtelijkheid? Belangrijkheid?

MaMetriek 15: Efficiënt verstrekken van informatie

Het is net zoals bij metriek 39 moeilijk na te gaan of het model efficiënt is in het verstrekken van de informatie.

MaMetriek 16: Tevreden over informatie verstrekking

Aan de hand van het procesmodel dat Marc Pieterse heeft gemaakt is de informatieverstrekking goed. De informatie die in de attributen zitten van het UML model komen ook terug in het proces waarmee een boeking van een reis afgewikkeld kan worden.

PoMetriek 1: Het aantal entiteiten

Dit model bevat 16 entiteiten

Entiteiten = [Reis, Boeking, Reiziger, Vervoer reservatie, verblijfplaats, Verhuurmaatschappij, Autoverhuur, appartement, hotel, verblijf, eticket, vliegtickets, reisbureau, vliegmaatschappij, verzekeringsmaatschappij, verzekering]

PoMetriek 2: Aantal attributen

Dit model bevat 13 attributen.

Attributen = [datum vetrek, datum aankomst, bestemming, min deelnemer, max deelnemer, reisvoorkeuren, van, tot, naw, reizigerid, reis, reiziger, polis]

PoMetriek 3: Het aantal afgeleide attributen

Dit model bevat geen afgeleide attributen.

PoMetriek 4: Het aantal samengestelde attributen

Dit model bevat geen samengestelde attributen.

PoMetriek 5: Het aantal meerwaardige attributen

Dit model bevat geen meerwaardige attributen.

PoMetriek 6: Het aantal relaties

Dit model bevat 6 relaties

Relaties = [verzorgt, boekt 3x, huurt, sluit af]

PoMetriek 7: Het aantal M:N relaties

Dit model bevat 1 M:N relatie:

M:N Relatie = [boekt (reiziger -> reis)]

PoMetriek 8: Het aantal 1:N relaties

Dit model bevat 5 1:N relaties:

1:N relaties = [verzorgt, boekt 2x, sluit af, huurt]

PoMetriek 9: Het aantal N-ary relaties

Dit model bevat geen N-ary relaties.

PoMetriek 10: Het aantal binaire relaties

Dit model heeft geen binaire relaties.

PoMetriek 11: Het aantal reflexieve relaties

Dit model heeft geen reflexieve relaties.

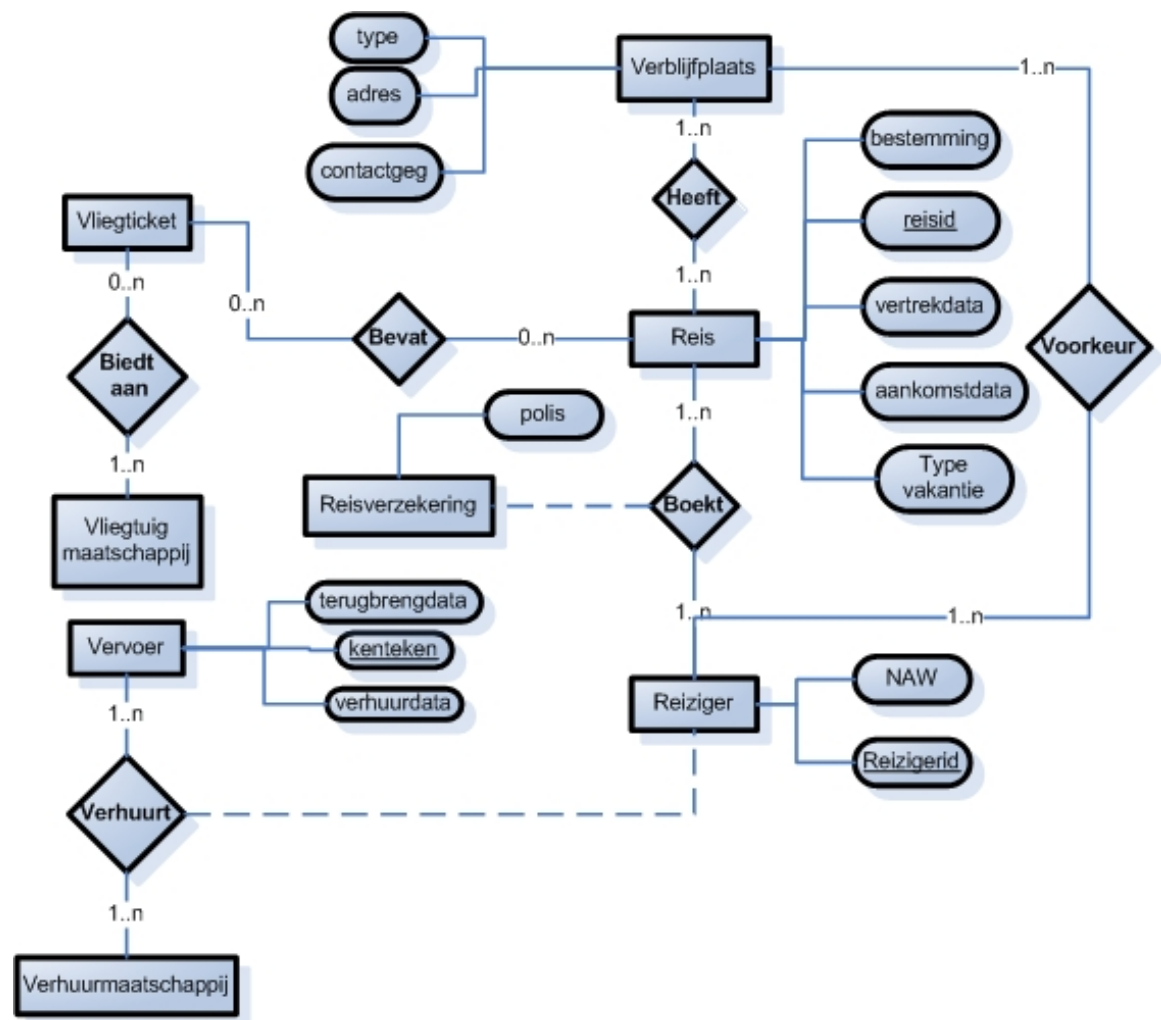
PoMetriek 12: Het aantal afgeleide entiteitstypen

Dit model heeft 4 afgeleide entiteitstypen.

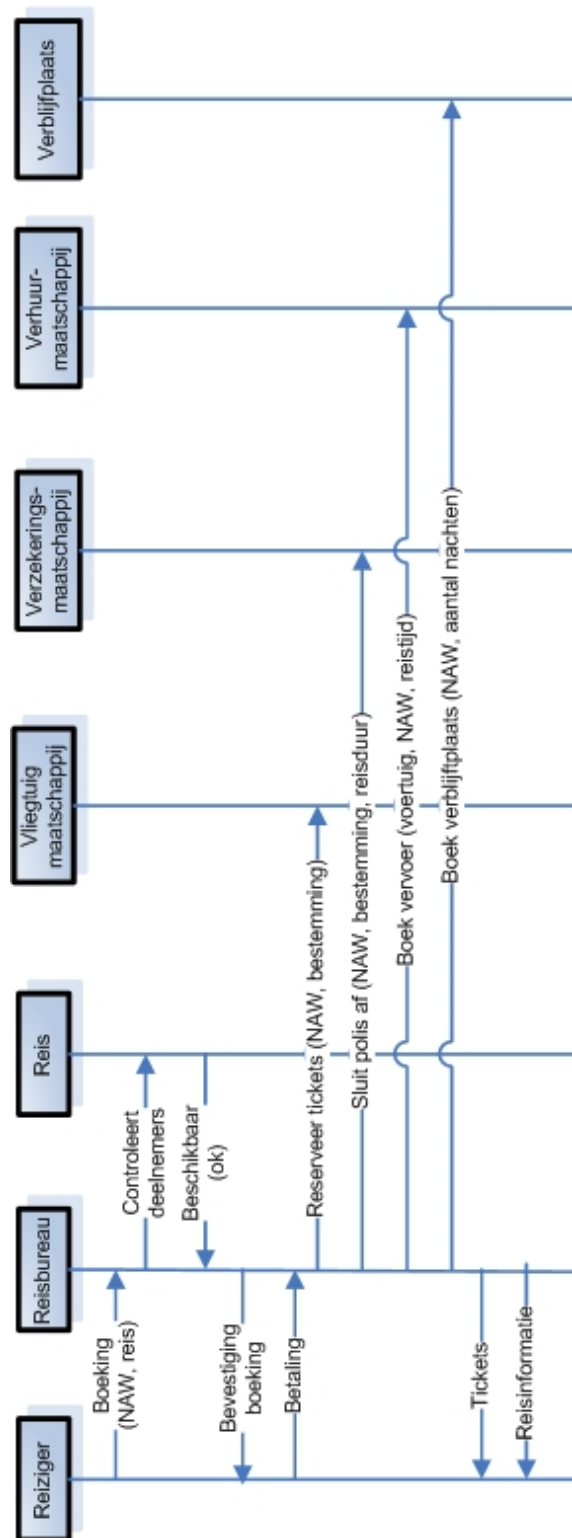
Afgeleide entiteitstypen = [Autoverhuur, Eticket, Hotel, Appartement]

Hoofdstuk 13

Appendix - B - Modellen



Figuur 13.1: Entity Relationship model van het reisbureau door Marc Pieterse



Figuur 13.2: Procesmodel van het boeken van een reis door Marc Pieterse

Bibliografie

- [1] Bansiya J., C. Davis, *A Hierarchical model for object-oriented design quality assessment*, IEEE Transactions on Software Engineering 28 (4-17) - 2002
- [2] Boehm B.W. , *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, USA (767) - 1981
- [3] L. Briand, J. Wüst, H. Lounis, *Replicated case studies for investigating quality factors in object-oriented designs*, Empirical Software Engineering 6 (volume 1) (11-58) - 2001
- [4] Cherfi S, Akoka J, Comyn-Wattiau I *Conceptual modeling quality - From EER to UML schemas evaluation*, Conceptual modeling - ER 2002, 2002
- [5] Diaz Oscar, Norman W. Paton and Jon Iturrioz *Formalizing and validating behavioral model through the event calculus*, Department of Computer Science, University of Manchester - Manchester, United Kingdom - 1998
- [6] DeLone W.H., E.R. McLean, *Information systems success: the quest for the dependent variable*, Information Systems Journal 3 (1) (60-95) - 1992
- [7] C. F. Eick *A Methodology for the design and Transformation of Conceptual Schemas* Department of Computer Science, Houston - 1991
- [8] R. Elmasri, S. Navathe, *Fundamentals of Database Systems, second ed.*, Addison-Wesley, Massachusetts - 1994
- [9] Enders A. , H.D. Rombach *A handbook of software and system engineering: Empirical Observations, Laws and Theories* Addison-Wesley 2003
- [10] Fenton, Neil *Software metrics: successes, failures and new directions*, City University, The Journal of Systems and Software 47 (149-157) - 1999
- [11] Genero M, Piattini M, Jimenez L *A metric-based approach for predicting conceptual data models maintainability*, International Journal of Software Engineering and Knowledge Engineering Volume: 11 Issue: 62 - 2001
- [12] Hitchman S. *Practitioner Perceptions On The Use Of Some Semantic Concepts In The Entity Relationship Model*, European Journal of Information Systems, 4, (31-40) - 1995

- [13] ISO/IEC *Standard 9126: Software Product Quality*, International Standards Organisation (ISO), International Electrotechnical Commission (IEC) - 2001.
- [14] S. Kesh, *Evaluating the quality of entity relationship models*, Information and Software Technology 37 (12) 681-689 - 1995
- [15] Krogstie, Lindland Sindre, *Defining quality aspects for conceptual models*, in: Proceedings of the 3rd IFIP 8.1 Working Conference on Information Systems, Marburg, Germany, (216-231) - 1995
- [16] Krogstie J., *Integrating the understanding of quality in requirements specification*, ACM SigSoft Software Engineering Notes 23 (86-91) - 1998)
- [17] Krogstie J., O. Lindland, G. Sindre, *Towards a deeper understanding of quality in requirements engineering*, in: Proc CAiSE 1995 (82-95) - 1995
- [18] Lindland, Sindre, Solveberg *Understanding Quality in Conceptual Modelling* IEEE Software, March - 1994
- [19] Lange Christian F.J., Martijn A.M Wijns, Michel R.V. Chaudron *Supporting task-oriented modeling using interactive UML views*, Technische Universiteit Eindhoven, Elsevier (399-419) - Juli 2007
- [20] Lauesen S. , Vinter O., *Preventing requirement defects*, in: Proceedings of the 6th International Workshop on Requirements Engineering: Foundation for Software Quality, Stockholm, Sweden - 2000
- [21] A. Maes, G. Poels, *Evaluating quality of conceptual modelling scripts based on user perceptions*, Data and knowledge Engineering - 2007
- [22] Moody D.L. *Theoretical and practical issues in evaluating the state of art* Data and Knowledge Engineering Volume 55 (243-276) - 2005
- [23] Moody Daniel L. *Metrics for Evaluating the Quality of Entity Relationship Models* Simsion Bowles and Associates, Melbourne
- [24] Moody D., G. Shanks, *Improving the quality of data models: Empirical validation of a quality management framework*, Inform. Syst. 28 (619-650) - 2003
- [25] Moody D. *Graphical Entity Relationship Models: Towards A More User Understandable Representation of Data* B. THALHEIM (ed.) Proceedings of the Fourteenth International Conference on the Entity Relationship Approach, Cottbus, Germany, October 7-9, (227-244) - 1996
- [26] Moody Daniel L. , Shanks G.G. *What Makes A Good Data Model? A Framework for Evaluating and Improving the Quality of Entity Relationship Models*, Australian Computer Journal - 1998
- [27] Olive* Antoni , Teniente Ernest *Derived types and taxonomic constraints in conceptual modeling*, Department de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Jordi Girona 1-3, 08034 Barcelona, Catalonia, Spain - 2002

- [28] Olive A. *Specific relationship types in conceptual modeling: the cases of generic and with common participants*, unpublished keynote lecture Fourth International Conference on Enterprise information systems, Ciudad Real, Spain 2002
- [29] Olive Antoni and Sancho Maria-Ribera *Validating conceptual specifications through model executions*, Universitat Politècnica de Catalunya, Facultat d'Informàtica de Barcelona, Pau Gargallo 5, 08028 Barcelona - Catalonia - 1996
- [30] Panagis Y, Sakkopoulos E, Sirmakessis S, et al. *Discovering reusable design solutions in web conceptual schemas: Metrics and methodology*, Web engineering, proceedings Volume: 3579 - 2005
- [31] Poels Geert, Genero Marcelo, Piattini Mario *Defining and validating metrics for assessing the understandability of entity-relationship diagrams*, Elsevier - Oktober 2007
- [32] Poels G., Genero M., Nelson J, Piattini M. *Quality in conceptual modeling: five examples of the state of the art* Data and Knowledge Engineering Volume 55 (237-242) - 2005
- [33] G. Poels, G. Dedene, *Distance-based software measurement: necessary and sufficient properties for software measures*, Information and Software Technology 42 (volume 1) (35-46) - 2000
- [34] Proper H.A. (Erik) *Grounded Enterprise Modelling* - 16 Mei 2007
- [35] Raed, Li *Software metrics: successes, failures and new directions*, The University of Alabama in Huntsville, The Journal of Systems and Software 81(1868-1882) - 2008
- [36] P. Seddon, *A respecification and extension of the DeLone and McLean model of IS success*, Information Systems Research 8 (3) (240-253) - 1997
- [37] Seffah A, Donyaee M, Kline RB, et al. *Usability measurement and metrics: A consolidated model* Software quality journal Volume: 14 Issue: 2 - 2006
- [38] Serrano M, Calero C, Trujillo J, et al. *Empirical validation of metrics for conceptual models of data warehouses*, Advanced Information systems engineering, proceedings Volume: 3084 - 2004
- [39] Simsion, G.C. *Data Modelling Essentials* Van Nostrand Reinhold, New York -1994
- [40] Symons C.R., *Function Point Analysis: Difficulties and Improvements*, IEEE Transactions on Software Engineering, 14(1), January. - 1988
- [41] Uthmann C. von, J. Becker, *Guidelines of modelling (GoM) for business process simulation*, ProcessModelling, Springer-Verlag, Berlin, Germany - 1999.
- [42] P. Verschuren, H. Doorewaard, *Het ontwerpen van een onderzoek*, Uitgeverij LEMMA, Den Haag - 2007

- [43] Warmer, Kleppe *Praktisch UML*, Pearson Education, 3e Editie - maart 2004
- [44] Xiaoqing, Gautam, Monu *An intelligent early warning system for software quality improvement and project management*, University of Missouri-Rolla, The Journal of Systems and Software 79 (1552-1564) - 2006