

## Abstract

The aim of this report is to define a model that outlines the production of geographical data. This model reveals how to ensure that the product fulfils all rules and demands that are stated in the specification, during the production. By means of the model, it is also studied, how to make the production more efficient, and how to improve the quality of the product.

Extensive research on relevant theory has been carried out. It is outlined how it is possible to model the real world, and how such models can be implemented in Geographical Information Systems and databases. It is also stated which types of errors that typical arises during the production of geographical data, and how the number of errors can be substantiated. When one wants to analyse the practical use of the product it is important that this substantiate is unambiguous. It is also important that the product is described on several levels. This is done by the use of a number of quality elements, which makes it possible to put a value on the quality of the product.

In this report the production of geographical data is stated as a process of four stages. The first stage is the preparation of the specification, which defines the production of the product. Second stage encloses an interpretation of data. The last to stages is respectively an internal and an external control.

The preparation of the specification consists in defining the rules that the product needs to fulfill. These rules differ widely, and are all useful at the last three stages of the production.

The interpretation of data is carried out by the use of the interpretation rules that are defined in the specification. This makes it possible to register an object from the given data. It is then, in the internal control, controlled whether this object e.g. observances the topological rules. These rules are also defined in the specification. When all objects have undergone the internal control, then the whole product must go through the external control. The external control compares the product with a known reference, and the result of this comparison, is assessed to the rules of quality, which are defined in the specification.

This report also outlines a method which automates part of the internal control. This improves the product and makes the production easier, which is the objective of this report.

## Resumé

Formålet med dette projekt er at opstille en model som beskriver produktionen af geografiske data. Modellen giver indsigt i hvordan det sikres, at produktet overholder de krav, der stilles i specifikationen, under produktionen. Ved hjælp af modellen undersøges det desuden, hvordan produktionen kan effektiviseres, og hvordan produktets kvalitet kan forbedres.

For at belyse problemstillingen er relevante teoretiske emner beskrevet. Her gives bl.a. en beskrivelse af, hvordan det er muligt at modellere den virkelige verden, og hvordan sådanne modeller kan håndteres i forbindelse med geografiske informations systemer og databaser. Derudover beskrives det, hvilke typer af fejl der kan opstå under produktionen af geografiske data, samt hvordan omfanget af fejl kan dokumenteres. Det er vigtigt, at denne dokumentation er entydig, når det skal vurderes, hvor brugbart et produkt er. Ligesom det er vigtigt, at produktets kvalitet beskrives på flere niveauer. Ved hjælp af en række kvalitetselementer, er det muligt at vurdere kvaliteten af det samlede produkt.

I rapporten beskrives produktionen af geografiske data som bestående af fire overordnede processer. Dette indbefatter en udarbejdelse af en specifikation, som produktet kan produceres efter, en fortolkning af et datagrundlag, samt en intern og ekstern kontrol.

Udarbejdelsen af specifikationen består i at definere de regler, der skal overholdes i forbindelse med produktionen. Disse regler er af forskellig karakter, og benyttes under fortolkningen samt de to kontroller.

Fortolkningen skal ske på baggrund af fortolkningsreglerne fra specifikationen. Herved er det muligt at registrere et objekt ud fra et givet datagrundlag. Dette objekt kan efterfølgende kontrolleres, i den interne kontrol, vha. blandt andet de topologiske regler, der ligeledes stammer fra specifikationen. Når alle objekter har gennemgået den interne kontrol, kontrolleres hele det producerede datasæt under den eksterne kontrol. Her sammenlignes datasættet med en reference, og resultatet af denne sammenligning, måles i forhold til kvalitetskravene fra specifikationen.

I rapporten er der ligeledes beskrevet en metode til at automatisere dele af den interne kontrol. Herved forbedres produktet og produktionen gøres mere effektiv, hvilket er målet med projektet.

## Forord

Denne rapport er udarbejdet i forbindelse med vores kandidatafhandling ved Danmarks Tekniske Universitet (DTU). Projektet er udført ved instituttet Informatik og Matematisk Modellering (IMM), faggruppen Geoinformatik, i perioden 3. februar 2003 - 7. juli 2003. Professor Ole Jacobi, DTU, har været vejleder på projektet og Jesper Vinther Christensen, Kort og Matrikelstyrelsen, har været medvejleder.

Arbejdets omfang svarer til 30 point i DTU's studiesystem.

Projektet er en fordybelse i et af problemfelterne, der blev opstillet i det forudgående arbejde i forprojektet. Hensigten er at give en indsigt i produktionen af geografiske data, samt at opnå en viden om nogle af de muligheder, der ligger i programmet ArcGIS i forbindelse med indlejring af regler.

I forbindelse med arbejdet vil vi gerne takke vores vejleder Professor Ole Jacobi, og medvejleder Jesper Vinther Christensen, for gode input og kritisk vejledning under hele processen. Vi vil også gerne sige tak til Kurt Andersen, Informi GIS, for stor hjælp til fremskaffelse af programmet ArcGIS.

7. juli 2003

c971474 - Jesper Kihlberg Christiansen

---

c971460 - Karsten Pihl Olsen

---





# Indholdsfortegnelse

<b>DEL 1 - INDLEDNING .....</b>	<b>1</b>
1 BAGGRUND .....	3
2 PROBLEMFOMULERING .....	5
3 AFGRÆNSNING .....	7
4 DEFINITIONER.....	8
4.1 GIS.....	8
4.2 Specifikation.....	8
4.3 Producent.....	8
4.4 Bruger.....	8
4.5 Entiteter.....	9
4.6 Objekter.....	9
4.7 Datasæt .....	9
4.8 Datamodel.....	9
4.9 Topologi.....	9
4.10 Kvalitet.....	9
4.11 Fitness for use.....	10
5 LÆSEVEJLEDNING .....	11
<b>DEL 2 - BAGGRUNDSTEORI.....</b>	<b>13</b>
6 MODELLERING AF VIRKELIGHEDEN.....	15
7 OBJEKTORIENTERET GIS .....	19
7.1 UML.....	19
7.2 Objektklasser og nedrivning.....	20
7.3 Relationer.....	21
7.4 Metoder.....	22
7.5 Afslutning.....	24
8 DATABASES.....	25
8.1 Introduktion til databaser.....	25
8.1.1 Databasestruktur .....	26
8.2 Databasetyper.....	28
8.2.1 Relationelle databaser .....	28
8.2.2 Objektorienterede databaser .....	30
8.2.3 Objekt relationelle databaser .....	31
8.2.4 Afslutning.....	33

9	MODELLER.....	34
9.1	<i>Feltbaserede modeller</i> .....	35
9.1.1	Operationer på felter .....	37
9.2	<i>Objektbaserede modeller</i> .....	40
9.2.1	Rumlige objekter .....	43
9.2.2	Operationer på rumlige objekter .....	45
9.2.3	Topologiske relationer.....	47
10	FEJL.....	50
10.1	<i>Fejl i produktionsprocessen</i> .....	50
10.1.1	Indsamling af rådata .....	50
10.1.2	Datagenerering.....	51
10.1.3	Datalagring .....	53
10.2	<i>Fejltyper</i> .....	53
10.2.1	Strukturelle fejl.....	54
10.2.2	Geometriske fejl.....	55
10.2.3	Topologiske fejl.....	57
11	DOKUMENTATION AF KVALITET .....	59
11.1	<i>Kvalitetslementer</i> .....	59
11.1.1	Kvantitative elementer .....	60
11.1.2	Kvalitative elementer .....	65
11.2	<i>Bestemmelse af kvalitetslementer</i> .....	67
<b>DEL 3 - ANALYSE.....</b>		<b>69</b>
12	PRODUKTIONSMODEL .....	71
13	SPECIFIKATION.....	74
13.1	<i>Opbygning</i> .....	74
13.1.1	Datamodel.....	75
13.1.2	Regler .....	77
13.1.3	Afslutning .....	80
13.2	<i>Udarbejdelse</i> .....	80
13.2.1	Metode.....	81
14	FORTOLKNING.....	83
14.1	<i>Datagrundlag</i> .....	83
14.2	<i>Fortolkningsregler</i> .....	84
14.2.1	Håndtering af fortolkningsregler .....	85
15	INTERN KONTROL .....	89
15.1	<i>Formatregler</i> .....	89
15.1.1	Attributregler.....	90
15.1.2	Geometriske regler .....	91

---

15.2	<i>Kvalitetsregler</i> .....	92
15.3	<i>Topologiske regler</i> .....	92
15.3.1	OpenGIS-funktioner .....	94
15.3.2	Alternative funktioner .....	100
15.3.3	Regler i Top10dk.....	109
15.4	<i>Implementering</i> .....	115
15.4.1	Topologi i ArcGIS .....	117
15.4.2	Topologiværktøj.....	121
16	<b>EKSTERN KONTROL</b> .....	127
16.1	<i>Bestemmelse af kvalitetsniveau</i> .....	128
16.1.1	Referencegrundlag.....	128
16.2	<i>Kontrol af kvalitetselementer</i> .....	130
16.2.1	Håndtering af kvalitetselementer .....	130
16.3	<i>Afslutning</i> .....	132
	<b>DEL 4 – AFSLUTNING</b> .....	<b>133</b>
17	<b>KONKLUSION</b> .....	135
18	<b>PERSPEKTIVERING</b> .....	137
19	<b>LITTERATURLISTE</b> .....	139
	<b>APPENDIKS</b> .....	<b>143</b>



# Del 1 - Indledning

I denne del præsenteres emnet for rapporten. Der gives en forklaring af baggrunden for projektet samt en beskrivelse af de problemfelter, der ønskes besvaret. Desuden beskrives nogle af begreber, der benyttes i rapporten.



# 1 Baggrund

Anvendelsen af geografiske data har de seneste år været stigende indenfor mange forskellige fagområder. Dette er gældende indenfor navigation og ruteoptimering, bymodeller og byplanlægning, miljøovervågning og -vurdering samt en lang række andre. Anvendelsen af geografiske data findes ligeledes i mange forskellige målestokke. Helt fra store systemer, der er gældende for hele universet, jorden, kontinenter og lande til mere lokale systemer, der dækker landsdele, amter, kommuner, byer og lokalsamfund. Et felt hvor geografiske informationer også kan være til gavn, er til lagerstyring af større eller mindre virksomheder samt til brug på hospitaler til styring af sengepladser og operationsstuer.

Stadig større mængder geografisk information indsamles og opbevares til forskellige formål. Disse informationer benyttes selvstændigt eller kombineret i forbindelse med en lang række projekter. Et eksempel på dette kan være til analyse af folks bevægelsesadfærd ved valg af kollektiv transport, hvor trafiktællinger benyttes samme med busruter og køreplaner. Det kan også være forskellige markedsanalyser, hvor det er muligt at benytte befolkningsfordelinger sammen med informationer omkring distributionssteder.

For at anvende enhver information, er det nødvendigt at have kendskab til baggrunden for de benyttede data. Dette gælder for geografiske data såvel som andre data, og det er derfor nødvendigt at der eksisterer information om kvaliteten af data. Ud fra kvalitetsoplysningerne er det muligt at tage beslutninger eller analysere ting, der har med disse data at gøre.

Kvaliteten er ikke kun bestemt af data i sig selv, men også af den sammenhæng de anvendes i. Der er forskellige aspekter inden for kvaliteten af geografiske informationer, der er vigtige for de forskellige anvendelsesområder. Det er derfor nødvendigt at opbygge en viden om denne kvalitet, herunder hvilke kvalitetselementer, der er nødvendige og hvilke, der er ønskværdige, for at være i stand til at beskrive kvaliteten.

Kvaliteten bestemmes bl.a. ud fra hvordan data er produceret, og den producent, der fremstiller data, registrerer disse kvalitetsinformationer. Under produktionen skal producenten, ud over kvaliteten, også tage hensyn til andre ting såsom markedsforhold og økonomi.

Producenten skal undersøge markedets behov for de pågældende data og finde ud af om det kan betale sig at producere disse. Der skal foregå en dialog mellem producenten og brugerne, så brugerne kan fortælle hvilke produkter de kunne tænke sig, og fortælle hvordan et eventuelt eksisterende produkt kan forbedres. Ud fra denne viden, er det muligt for producenten at producere et produkt bestående af de ønskede data.

For producenten af data, vil der generelt være to målsætninger for produktionen. Den første er at optimere produktet økonomisk, hvilket primært gøres ved at finde den billigste produktionsmetode. Den anden målsætning er at sikre fitness for use, dvs. at sørge for at produktet, i brugernes øjne, passer til det som det skal bruges til.

For en virksomhed, der producerer et produkt, er det ønskeligt at være i stand til at sikre, at kvaliteten af data lever op til de stillede krav, gøre produktionsprocessen hurtigere, og derved gøre det muligt at få opdateringer eller nye produkter på markedet hurtigere. For producenten er det også ønskværdigt at mindske udgifterne til produktionen, f.eks. hvis det ikke er nødvendigt at have højtuddannede medarbejdere som operatører, eller ved at automatisere processer, der tidligere krævede meget arbejdskraft.

Produktionsprocessen kan optimeres ved at operatøren, der registrerer data, ikke skal bruge så meget tid til at overholde regler, som computeren kan holde styr på, men i stedet kan koncentrere sig om det mennesket er bedre til, nemlig abstraktion og fortolkning. På den måde er det muligt at opnå en bedre udnyttelse af arbejdskraften og derved spare økonomiske ressourcer.

På baggrund af ovenstående, forsøges det i denne rapport, at besvare spørgsmålene der er opstillet i den følgende problemformulering.



## 2 Problemformulering

Formålet med dette projekt er at finde forskellige metoder til, dels at forbedre produkter, der består af geografisk data, og dels at forbedre den proces der foregår, for at tilvejebringe geografisk data. Arbejdet vil koncentrere sig om hvordan der, ud fra en specifikation, kan produceres et produkt, der lever op til denne specifikation, og hvorledes det dokumenteres at produktet gør dette.

På baggrund af dette skal følgende hovedspørgsmål besvares.

**Hvordan skal produktionen af geografiske data foregå, så data overholder de specificerede krav?**

Herunder

**Hvilken former for kontrol skal der foretages under produktionen?**

Et af formålene med arbejdet er, at finde en datastruktur, hvor specifikationen er indbygget i data. Det vil sige at et objekt har indbygget funktionalitet, der gør det lettere for operatøren at overholde kravene fra specifikationen.

Dette gøres ved bl.a. at opstille en række fortolkningsregler og topologiske regler i specifikationen for data, som data skal overholde. For at undersøge om dette er muligt, skal følgende spørgsmål besvares.

**Hvordan skal specifikationen struktureres, så samtlige krav kommer med?**

**Er det muligt at inddеле kravene fra specifikationen i grupper, og derved sikre et bedre overblik?**

**Er det muligt at automatisere reglerne fra nogle af disse grupper og derved simplificere produktionen?**

Ud over at forbedre arbejdsprocessen for operatøren, resulterer automatiseringen ligeledes i, at det sikres at kvaliteten dokumenteres. Dette undersøges ved at besvare følgende spørgsmål.

**Hvordan sikres det at kravene til kvaliteten overholdes og at alle kvalitetselementerne dokumenteres?**

Ved hjælp af ovenstående spørgsmål, søges det at undersøge hvordan produktionsprocessen kan optimeres, så den bliver mere økonomisk

og effektiv. Derudover sikrer fremgangsmåden, at det endelige produkt overholder de opstillede krav.

### 3 Afgrænsning

I dette projekt er produktionsprocessen forsøgt beskrevet ud fra en teoretisk tilgang, og der er derfor lagt mest vægt på denne teori. Af den grund er implementeringen kun sket i mindre omfang, og der er præsenteret enkelte eksempler, for at undersøge mulighederne ved eksisterende værktøjer, i dette tilfælde ArcGIS.

En del af projektet består i at opnå et kendskab til ArcGIS, og undersøge om nogle af de ønskede implementeringer kan foretages heri, og dette er derfor gjort i mindre omfang. Dette gør at andre eksisterende programmer ikke er afprøvet nærmere. Derimod gives der et eksempel på hvordan en implementering kan foretages vha. et tænkt eksempel.

Der tages udgangspunkt i et produkt som f.eks. Kort og Matrikelstyrelsens topografiske produkt, Top10dk, og de eksempler, der gives i denne rapport, vil på mange punkter ligne dette. Der er dog også dele af eksemplerne, hvor der benyttes andre objektklasser og regler. Dette gøres for at beskrive hvilke muligheder, der er for at benytte topologiske regler, inden for andre former for geografiske data.

## 4 Definitioner

De følgende definitioner beskriver hvordan de enkelte begreber opfattes og benyttes i denne rapport.

### 4.1 GIS

GIS er forkortelsen for geografisk informationssystem. Det er et værktøj, der kan benyttes til at indsamle, lagre, opdatere, manipulere, analysere og vise alle former for geografisk relateret information [ESRI, 2003a].

### 4.2 Specifikation

En specifikation er et dokument, der indeholder en detaljeret teknisk beskrivelse af et produkt og hvilke krav et produkt skal opfylde. Specifikationer kan opdeles i to typer:

- Produktionsspecifikation
- Produktspecifikation

Produktionsspecifikation angiver hvordan et produkt skal fremstilles og hvad det skal indeholde mens produktspecifikation beskriver hvordan produktet er blevet til og hvad det indeholder. Dette uddybes i Kapitel 13.

### 4.3 Producent

En producent er i denne rapport betegnelsen for den person, det firma eller den organisation, der står for produktionen af et produkt. Denne produktion er hele processen, lige fra udformning af specifikation, generering af data og til leveringen af det endelige produkt. Dele af produktionen kan producenten udlicitere til andre, f.eks. genereringen af data eller kvalitetskontrollen. Alle disse aktiviteter falder, i denne rapport, ind under betegnelsen producent.

### 4.4 Bruger

Brugeren er betegnelsen for alle, der benytter det producerede produkt, dvs. personer eller virksomheder, der køber det produkt eller har adgang til det. Producenten selv kan også betegnes som en bruger,

eksempelvis hvis produktet benyttes i forbindelse med en række andre aktiviteter eller produkter.

#### **4.5 Entiteter**

En entitet betegner enhver fysisk eller abstrakt genstand eller begivenhed i virkeligheden, og kan ikke opdeles yderligere i fænomener af samme slags [Petersen, 1992].

#### **4.6 Objekter**

Et objekt er et element i en model, der repræsenterer en entitet fra virkeligheden. Objekter kan deles op i objektklasser, der har en række fællestræk. En sådan objektklasse kan f.eks. være repræsentationer af bygninger i modellen, og et objekt fra denne klasse kan da være repræsentationen af en bestemt bygning, f.eks. Bygning 115 på DTU.

#### **4.7 Datasæt**

Et datasæt er en samling af data. I en objektbaseret model vil et datasæt sædvanligvis bestå af en række objekter fra forskellige objektklasser.

#### **4.8 Datamodel**

En datamodel er det medium hvor entiteternes egenskaber kan beskrives. Det er vigtigt at modellen er i størst mulig overensstemmelse med den virkelighed, der skal repræsenteres [Petersen, 1992].

#### **4.9 Topologi**

Topologi er den rumlige relation mellem to eller flere objekter, der er forbundne eller støder op til hinanden [ESRI, 2003a].

#### **4.10 Kvalitet**

Kvaliteten af geografisk data kan kvantificeres i et endeligt antal elementer, der i denne rapport kaldes kvalitetselementer. Disse elementer beskriver hver især forskellige karakteristika ved produktet, og kan tilsammen give en fyldestgørende syn på, om produktet kan tilfredsstille en brugers krav.

### **4.11 Fitness for use**

Begrebet fitness for use betyder direkte oversat brugbarhed, og er et udtryk for, hvor godt et produkt tilfredsstiller en potentiel brugers behov. Alle brugere har forskellige tilgange til et produkt, og derfor er det stort set umuligt at tilfredsstille alles behov, men det er et mål for en producent at maksimere fitness for use for flest mulige brugere. Fitness for use relaterer til kvaliteten af data, men er ikke udelukkende baseret på dette, da et produkt kan have en høj kvalitet, men ikke indeholde den information, brugerne har behov for.

## 5 Læsevejledning

Rapporten består af fire dele; en indledende del, en teoridel, en analysedel samt en afsluttende del. I det følgende gives en kort introduktion til de enkelte dele, så det er muligt at skabe sig et overblik over den samlede rapportstruktur.

Del 1 indeholder en introduktion til emnet for rapporten, hvilke hovedspørgsmål der ønskes besvaret i rapporten, samt hvilke begrænsninger der er taget som udgangspunkt. Derudover indeholder denne del definitioner af nogle af de begreber der benyttes, og præciserer vores anvendelse af begreberne i rapporten. Herved er det muligt, under læsningen af rapporten, at vende tilbage til dette kapitel for at få præciseret meningen af med enkelte begreber.

Del 2 beskriver de overordnede teoretiske emner for rapporten, som er relevante eller nødvendige for besvarelsen af de spørgsmål, der er opstillet i problemformuleringen under Del 1. Først beskrives det, hvorledes det er muligt at modellere den virkelige verden samt hvilke modeller der kan benyttes til at beskrive verden. Derudover beskrives en række fordele ved at arbejde med objektorienterede GIS og databaser. Desuden beskrives en række mulige geometriske og topologiske fejl samt hvordan kvaliteten af geografiske data kan dokumenteres.

Del 3 indeholder selve analysen, der beskriver en produktionsmodel for produktionen af geografiske data. Denne produktionsmodel indeholder bl.a. udarbejdelsen af specifikationen, og indholdet af denne specifikation beskrives derfor i denne del af rapporten. Derudover beskrives de enkelte dele, som en produktion består af, herunder genereringen af data samt kontrollen af disse.

Del 4 rummer en afslutning på rapporten, hvor der foretages en opsamling af de resultater, der er opnået. Dette består af en konklusion samt en perspektivering, der belyser, hvad resultatet kan bruges til og hvilke videre muligheder der er med emnet.

Rapporten indeholder ligeledes fire appendikser, hvor den praktiske fremgangsmåde for arbejdet med ArcGIS, samt et mere detaljeret teoriafsnit omkring topologiske relationer, beskrives.





# Del 2 - Baggrundsteori

For at besvare de spørgsmål, der er opstillet i Det 1, er der en række teoretiske emner, der er relevante at belyses. Disse emner beskrives i denne del, og består bl.a. af en gennemgang af hvordan virkeligheden kan modelleres og hvordan sådanne modeller kan benyttes i forbindelse med objektorienteret GIS og databaser. Derudover beskrives hvilke former for fejl, der kan opstå, når der benyttes objektbaserede modeller, herunder geometriske og topologiske fejl. For at dokumentere hvilke fejl datasættet indeholder, er det muligt, vha. en række kvalitetselementer, at vurdere den samlede kvalitet. Disse kvalitetselementer beskrives derfor til sidst i denne teoretiske del af rapporten.



## 6 Modellering af virkeligheden

Den virkelige verden består af en uendelig mængde objekter, der hver især indeholder uendelige mængder af information. For os mennesker, der lever i denne verden, er det ofte nødvendigt at kunne forklare og forstå den, og det kan vi f.eks. gøre ved at danne en model af den [Guptill & Morrison, 1995]. For at gøre det, er det nødvendigt at begrænse mængden af information i modellen, så der ses på en afgrænset del af verden, der er endelig og tællelig. Formålet med at modellere den virkelige verden, er at forstå den bedre. Ved hjælp af en modelleringsfunktion omsattes den virkelige verden til en model. Ved at bruge denne model, kan de erfaringer, der er opnået herved, føres tilbage til problemet i den virkelige verden. Dette gøres ved at bruge den inverse modelleringsfunktion. Et eksempel på dette kan være at en person skal finde vej fra Århus til København. Da han ikke er bekendt med den korteste vej, benytter han en model af virkeligheden til at overskue sit problem, i dette tilfælde et færdselskort. Kortet studeres og analyseres, og resultatet af analysen bruges da i den virkelige verden, til at finde de rigtige veje, der skal benyttes for at komme til København.

Modelleringsprocessen kan ses som en funktion  $m()$ . Et eksempel fra matematikkens verden kan være, at virkeligheden var:

$$x \rightarrow x^2$$

Det kan modelleres vha. modelleringsfunktionen:

$$m(x) = x^2$$

Den inverse funktion af dette, dvs. en funktion, der kan bruges til at føre modellen tilbage til virkeligheden ville være:

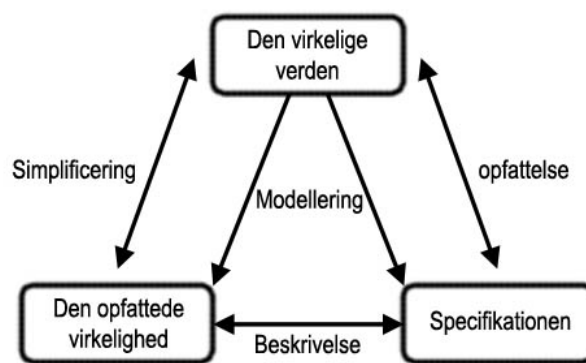
$$y \rightarrow \sqrt{y}, \text{ eller } m^{-1}(y) = \sqrt{y}$$

Hvor  $y$  er en værdi fra modellen, der ønskes ført tilbage til den virkelige verden. Det vil altså sige, at  $y = m(x)$ .

For at modellen kan have værdi for os, er det nødvendigt, at det er muligt, ved hjælp af en invers modelleringsfunktion, at omsætte indholdet i modellen til noget, der ligner den virkelige verden, således at:

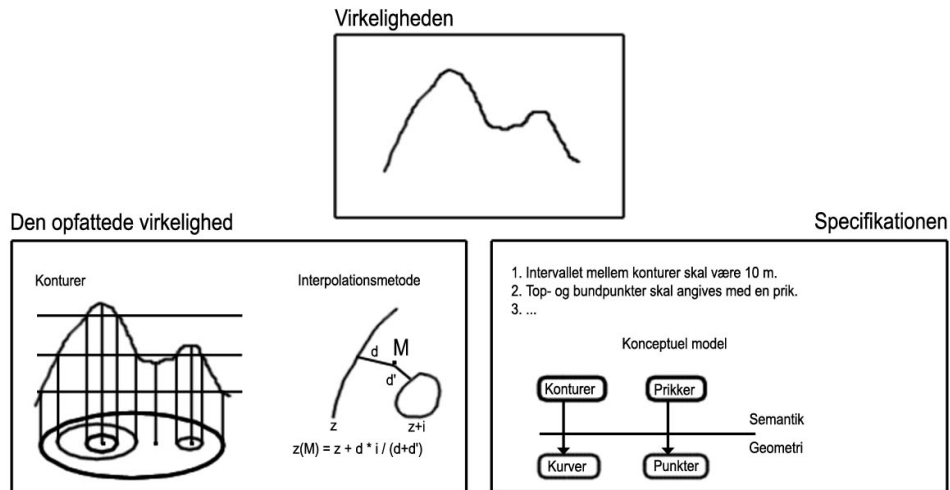
$$m^{-1}(m(x)) \approx x$$

Det er nødvendigt med nogle regler for, hvilke ting der skal med, og hvilke ting, der ikke skal med i modellen, samt hvordan de elementer, der indgår, skal hænge sammen. Dette beskrives i en specifikation. Det billede, der dannes af den virkelige verden, ud fra specifikationen, kaldes den opfattede virkelighed (Universe of discourse), og er en simplificering af den virkelige verden. Specifikationen er synet på verden, og beskriver den opfattede verden. På figur 6-1 kan sammenhængen mellem disse elementer ses.



Figur 6-1 Sammenhæng mellem den virkelige verden, den opfattede verden og specifikationen [Guptill & Morrison, 1995].

Eksempelvis kan den virkelige verdens overflade modelleres som konturlinier og interpolationsfunktioner, der er en matematisk approksimation af terrænets form. Den opfattede virkelighed er da det matematiske objekt (linier og interpolation), mens specifikationen definerer intervallet mellem konturlinierne og de interpolationsparametre, der skal benyttes. Data, der opfylder de krav, der er opstillet i specifikationen, repræsenterer den opfattede virkelighed, der er virkeligheden set igennem eller filtreret af modellen, se figur 6-2.



Figur 6-2. Et eksempel på hvordan den virkelige verden modelleres.

Specifikationen består af to dele. En beskrivelse af indholdet, der også kaldes for datamodellen, samt det konceptuelle dataskema, der består af de regler, som data i datasættet skal overholde. Det konceptuelle dataskema er en beskrivelse af hvilke relationer, data har indbyrdes. Hvis begreberne sammenlignes med opbygningen af et sprog, kan det konceptuelle dataskema sammenlignes med sprogets grammatik, mens indholdet kan sammenlignes med ordforrådet. Datasættet er da en tekst, der er opbygget af sætninger, der udtrykker sproget [Guptill & Morrison, 1995].

De objekter, der findes i datasættet, eksisterer selvsagt ikke i virkeligheden, men er en del af den opfattede virkelighed. For eksempel kan specifikationen definere forskellige vejtyper vha. vejbredeklasser, og den opfattede virkelighed er da et filter, der tildeler enhver vej en vejklasse. Det kan også være at specifikationen anviser, at der skal foretages en egentlig måling af vejbredden. Det vil føre til en helt anden opfattet virkelighed. Så snart der foretages en klassifikation af objekter, vil der opstå problemer ved grænserne af disse klasser. Dette skyldes af der, for det meste, er en flydende overgang i virkeligheden, og det derfor er nødvendigt at dele objekterne op og dermed foretage en generalisering.

Når den virkelige verden skal repræsenteres i eksempelvis et GIS, er det nødvendigt at udvælge de objekter, der skal indgå i data. Denne udvælgelse er et vigtigt element i specifikationen. Der defineres ting som minimumsstørrelse og topologiske relationer, og det beskrives, hvilke elementer, der skal tages med, og hvordan. Denne proces

involverer handlinger, der transformerer den virkelige verden til felter, objekter, attributter og relationer, også kaldet data. Disse handlinger kan f.eks. være generaliseringer, simplificeringer, interpoleringer eller ekstrapoleringer [Guptill & Morrison, 1995].

## 7 Objektorienteret GIS

Traditionelt modelleres geografiske data vha. relationelle datamodeller, og da denne metode kom frem i 1970'erne var det et stort fremskridt for anvendelsen af GIS. På samme måde er der nogen som mener at objektorienterede datamodeller er det mest betydningsfulde udvikling inden for GIS teknologien de sidste 10 år [DeMartino & Hrnicek, 2001].

En måde hvor forskellen mellem den objektorienterede datamodel og den traditionelle relationelle model kan illustreres, er ud fra Professor Doug Ross' beskrivelse af en datamodel [Woodsford, 1995]:

Model = Data + Struktur + Algoritme

Med dette menes, at en brugbar model er afhængig af de rigtige forhold mellem disse tre elementer. Den relationelle fremgangsmåde understøtter en model med data og struktur, men holder algoritmerne udenfor modellen og overlader dem til den anvendte applikation. Derimod bringer den objektorienterede fremgangsmåde algoritmerne ind i modellen, og gør dermed modellen applikationsuafhængig.

Ud over at den traditionelle relationelle fremgangsmåde ikke indeholder algoritmer direkte i modellen, er strukturen også anderledes opbygget. Et karakteristika ved en objektorienteret datamodel er dens hierarkiske struktur, muligheden for komplekse datatyper samt mulighed for nedrivning, hvilket beskrives i de følgende afsnit. Til at modellere objektorienterede modeller benyttes modelleringssproget Unified Modelling Language (UML) ofte, og der gives derfor en kort introduktion til dette i det følgende. Selve notationerne beskrives løbende sammen med deres betydning.

### 7.1 UML

Objektorientering har været benyttet i en række år inden for programmering og modellering, men det er som sagt først inden for de seneste 10 år, at fremgangsmåden er dukket op i forbindelse med GIS. En af fordelene ved objektorienterede GIS er, at datamodellen i højere grad kan afspejle den virkelige verden [Woodsford, 1995]. På den måde er det muligt at modellere den objektorienterede database vha. et modelleringssprog som f.eks. UML. Dette sprog og dets notationer benyttes i denne rapport og beskrives derfor i de følgende afsnit. UML benyttes ligeledes i forbindelse med etablering af datamodeller i

programmet ArcGIS, der er benyttet under dette projekt, og fremgangsmåden for dette beskrives nærmere i Appendiks 3.

UML er et standardiseret modelleringssprog, der er en samling af forskellige metoder udviklet af Grady Booch, Ivar Jacobson og James Rumbaugh [Fowler, 2000] [Pedersen, 2000], og er en del af OMG-standarden (Object Management Group). UML kan anvendes til visualisering, dokumentering, specificering og konstruktion af modeller [Pedersen, 2000].

UML er et grafisk modelleringssprog, der indeholder en lang række diagramtyper. I denne rapport benyttes der dog kun objektogrammer, da denne fremgangsmåde benyttes til at udvikle modeller, der er dataorienterede [Fowler, 2000].

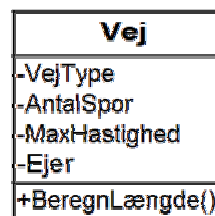
Styrken ved UML ligger i at det er muligt for personer, med forskellige tilgangsvinkler til emnet, at forstå hinanden. Dette er med til at reducere det gap, der kan opstå, når specifikationen skal omsættes til et produkt [Christiansen & Olsen, 2003], idet det giver en metode til at udtrykke datamodeller, der har et ensartet sprog, og som folk på forskellige niveauer kan forstå.

## 7.2 Objektklasser og nedarvning

En af de største fordele ved objektorienterede datamodeller er at denne form for modeller afspejler den måde, data eksisterer i den virkelige verden [Shekhar & Chawla, 2003] [Zeiler 2000] [DeMartino & Hrnicek, 2001]. Traditionelle modeller kræver derimod en oversættelse mellem de forskellige objekter og modellen, er eksempelvis kan være en relationel.

Den objektorienterede datamodel gemmer objekter i klasser, der er logiske grupper af data [DeMartino & Hrnicek, 2001]. Hver klasse har forudbestemte egenskaber, der er unikke for det pågældende objekt og er bygget direkte ind i objektet.

En objektklasse kan altså betegnes som en skabelon for et objekt og hver objektklasse indeholder en række attributter og metoder [Pedersen, 2000]. På figur 7-1 ses et eksempel på en objektklasse. Denne klasse har navnet Vej, attributterne VejType, AntalSpor, MaxHastighed og Ejer samt metoden BeregnLængde() og illustreres som en firkantet kasse, der er opdelt i tre rum, som vist på figuren. Rummene indeholder hhv. navn, attribut og metode. Attributterne beskriver hvilke data objekterne fra denne klasse skal indeholde, mens

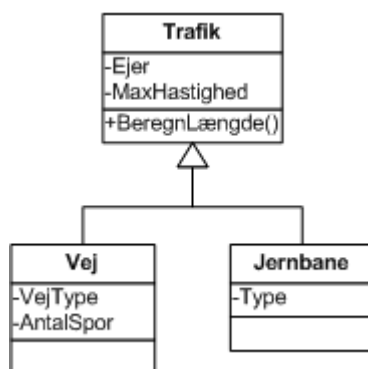


Figur 7-1  
Objektklasse



metoderne gør det muligt at rette, hente eller foretage beregninger ud fra eksempelvis attributterne.

I objektorienterede datamodeller er det muligt at samle en gruppe objektklasser, der har en række fælles egenskaber i form af attributter og metoder. Denne gruppe af objektklasser betegnes herved som underklasser eller specialiseringer af en overordnet generel klasse, og kan arve denne klasses egenskaber. Det gøres ved at opbygge objektklasserne i en hierarkisk struktur, som vist på figur 7-2. Her nedarver objektklasserne Vej og Jernbane egenskaberne fra den overordnede klasse Trafik. Vej og Jernbane indeholder samtlige egenskaber, der er defineret i Trafik, men sammenfaldende egenskaber er her kun illustreret i den generelle klasse, Trafik. Generaliseringen angives vha. pilen mellem den generelle klasse, Trafik og de to specialiseringer, Vej og Jernbane. Denne form for notation er en del af UML, som beskrevet tidligere.



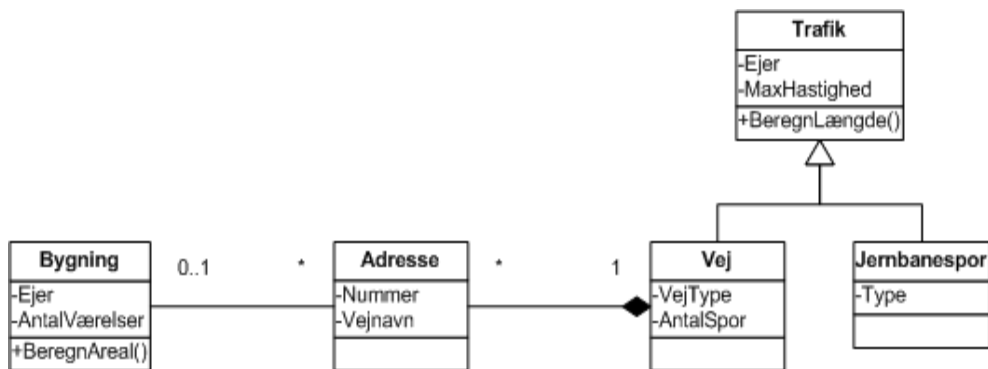
Figur 7-2 Nedarvning

Et objekt repræsenterer en entitet i den virkelige verden, og kan skabes ved at der oprettes en instans af en objektklasse. På den måde modelleres objektet entitetens egenskab i form af attributter, og opførsel som metoder.

### 7.3 Relationer

Den objektorienterede datamodel indeholder, ligesom den relationelle, relationer, der kan forbinde objekter fra en objektklasse til objekter fra en anden. Relationerne kan inddrages i tre kategorier, associeringer (associations), generaliseringer (generalizations) og aggregeringer (aggregations) [Shekhar & Chawla, 2003]. Generaliseringer er beskrevet i foregående afsnit og illustreres bedst vha. figur 7-2.

Associeringer er relationer mellem instanser af objektklasser [Fowler, 2000] og kan f.eks. være en forbindelse mellem objekter fra klassen Bygning og objekter fra klassen Adresse som illustreret på figur 7-3. Associeringer mellem to klasser kan have multiplicitet, hvor en forbindelse f.eks. kan være en-til-mange, og indikerer hvor mange objekter fra hver klasse, der kan indgå i relationen. Det uddybes i det efterfølgende kapitel. Notationen for denne multiplicitet er i UML defineret som vist på figur 7-3, hvor en adresse kan have forbindelse med ingen eller ét objekt fra klassen Bygning mens ét objekt fra klassen Bygning kan relateres til ingen eller mange objekter fra klassen Adresser. \* repræsenterer altså nul til uendeligt antal relationer.



Figur 7-3 Relationer. Fra venstre er det associering, komposit aggregering og til højre generalisering

Aggregeringer kan inddeles i to grupper; svage og stærke aggregeringer. I UML-notationen benævnes de stærke som kompositte (composition) [Fowler, 2000]. Generelt kan aggregeringer betegnes som en-del-af relationer. Forskellen mellem stærke og svage aggregeringer er at de stærke gør, at et objekt skal være en del af ét andet objekt, mens en svag aggregering godt kan relatere til flere andre objekter. Derudover gør den stærke aggregering at objektet lever og dør med det objekt, det er del af [Fowler, 2000]. En stærk aggregering illustreres i UML ved en sort pil (ruderformet), som vist på figur 7-3, mens en svag aggregering vises med en tilsvarende pil, der er hvid.

## 7.4 Metoder

En anden stor fordel ved anvendelse af en objektorienteret datamodel er at det er muligt at tilføje funktioner i form af metoder til hver objektklasse. Med disse metoder er det muligt at tilføje adfærd til de

enkelte objekter. Metoderne kan foretage en række specifikke opgaver, der kan udføres når det kræves. Det kan f.eks. være en funktion til at hente eller beregne en bestemt værdi ud fra attributterne fra et objekt, men metoderne kan ligeledes udføre mere komplekse opgaver såsom at rette objektets attributter, oprette nye objekter eller at dele objektet op i to.

Med traditionelle GIS-applikationer, der benytter relationelle datamodeller, skriver en programmør kildekoden til disse metoder og implementerer dem direkte i applikationen som en funktion. Hvis der til et bestemt datasæt skal benyttes nogle specifikke funktioner, skal disse skrives og den nye version af applikationen skal derefter distribueres til alle, der benytter de pågældende data. Dette gør det nødvendigt at sikre, at alle har den seneste version af applikationen for at anvende datasættet.

Ved at indlejre metoderne i data, som det er muligt i en objektorienteret datamodel, kan det sikres, at metoderne altid følger data. På den måde adskiller den objektorienterede model sig fra den relationelle, der kræver at funktionerne håndteres sideløbende. Ved at benytte en objektorienteret datamodel, er det muligt at foretage ændringer i metoderne centralt, idet objektmodellen distribueres til samtlige brugere, uden det herefter er nødvendigt at foretage ændringer hos hver enkelt person.

Ulempen ved at benytte en objektorienteret datamodel er, at den skal planlægges nøje under designet af databasen, og ikke efterfølgende når data er lagt ind i databasen. Dette gør, at der kræves mere tid, og der stilles højere krav til ressourcerne inden dataudviklingen kan begynde [DeMartino & Hrnicek, 2001].

Fordelen ved at indlejre metoderne i datamodellen er, at det reducerer mængden af arbejde, der skal til for at vedligeholde applikationerne. Derudover er det muligt at indlejre automatiske kontroller af data, frem for at skulle foretage denne kontrol manuelt eller vha. funktioner, der er implementeret i en applikation. Når objekter føjes til den objektorienterede datamodel, kan data herved valideres automatisk vha. de indlejrede funktioner, og brugeren kan få at vide, hvad der er galt, når der opstår en fejl.

Der er mange muligheder for at tilføje specielle opførsler til de forskellige klasser når der benyttes en objektorienteret datamodel. Forskellige metoder til at kontrollere data beskrives i et senere kapitel.

## 7.5 Afslutning

Ved at anvende en objektorienteret datamodel opnås en række fordele for både udvikleren og for brugeren af datamodellen, men alle har ikke nødvendigvis fordel af alle mulighederne. En del af fordelene er ikke noget brugeren umiddelbart har glæde af [Woodsford, 1995]. Ud over de tidligere nævnte fordele, er de objektorienterede datamodeller bl.a. gode til at håndtere store datamængder. Dette kan brugeren ikke direkte se, men får alligevel glæde af de yderligere muligheder dette medfører.

Dér hvor objektorienteringen har sin styrke, er at den gør modelleringen og efterfølgende analyser mere simple, ved at få det til at passe til den måde folk tænker på og til deres intuition [Zeiler 2000].

I det følgende kapitel beskrives det, hvordan den objektorienterede fremgangsmåde kan benyttes i forbindelse med databaser.

## 8 Databaser

For at opnå en forståelse for hvordan et GIS er opbygget er det nødvendigt med et vist kendskab til princippet bag databaser [Worboys, 1995], da et GIS reelt er et værktøj til at behandle og håndtere data, der kan sammenlignes med et DataBase Management System (DBMS). De mest anvendte databaser til GIS er traditionelt set relationelle databaser, mens udbredelsen og udviklingen af systemer, der håndterer objektorienterede databaser, har stor fremgang. Objektorienterede databasesystemer har langt fra overtaget markedet fra de relationelle systemer, selvom nogle mener at det vil ske inden så længe [Leavitt, 2001].

De seneste år er der kommet en del systemer, der kombinerer fordelene ved den objektorienteret fremgangsmåde med den traditionelle relationelle database [Pedersen, 2000] [Zeiler 2000] [MacDonald]. Den relationelle, den objektorienterede og den kombinerede databaseform beskrives i de følgende afsnit, for at introducere databaser og for at give et overblik over, hvordan de bruges i forbindelse GIS.

### 8.1 Introduktion til databaser

Da computersystemer begyndte at blive udbredte i 1950'erne og 60'erne, var det muligt at gemme store mængder informationer i filer og dermed reducere omkostningerne, tiden og antallet af fejl i forbindelse med manuel datahåndtering. Efterhånden som brugen af computere udviklede sig, steg kravene til databehandlingen og håndteringen af forskellige datafiler også. Computerens udbredelse gjorde også, at det ikke kun var programmører, der benyttede de nye muligheder, men også folk inden for andre fagområder. Derfor var det heller ikke alle brugere, der havde forudsætninger til at foretage ændringer i data. Det gjorde at det blev nødvendigt, at nogen måtte skrive et program, der kan håndtere data i alle disse filer og derved gøre det muligt at foretage forskellige former for søgninger og ændringer. Et problem, der herved opstod, var at datafiler, blev etableret uafhængigt af hinanden, og derved var det ikke sikkert at to filer, der indeholdt den samme type information, også havde samme format. For at undgå disse problemer blev det nødvendigt at udvikle en anden og bedre datastruktur [Worboys, 1995].

Et andet vigtigt aspekt inden for håndtering af data er, at sikre at data er uafhængige af de applikationer de benyttes i. At data er uafhængig

af den benyttede applikation betyder bl.a., at filer ikke skal have en fast struktur, som det kræves når data læses direkte fra tekstfiler. Dataafhængigheden opnås altså ved at det program, der håndterer den fysiske placering af data, er skjult for brugeren [Worboys, 1995].

Et resultat af erkendelsen af ovenstående problemer var udviklingen af databaser i slutningen af 1960'erne [Date, 2000].

De ting, der har indflydelse på en database, kan opdeles i fire dele [Date, 2000]; data, bruger, hardware og software, hvor data og brugeren er de vigtigste faktorer. For at foretage forskellige operationer, såsom at indsætte, slette og udtrække data i databasen, er det nødvendigt at have en softwarepakke, der kan håndtere data i en database. Dette stykke software betegnes DataBase Management System (DBMS) og gør det muligt at foretage ændringer og vedligeholde databasen genne en række funktionaliteter. Dette sker gennem et specialiseret programmeringssprog kaldet et interaktionssprog (query language) [Worboys, 1995], der hører til DBMS'et, hvor Structured Query Language (SQL) er det mest udbredte, og det benyttes bl.a. i forbindelse med relationelle databaser. For at anvende og analysere data i en database er det nødvendigt at kommunikere med databasen og dette gøres f.eks. gennem SQL.

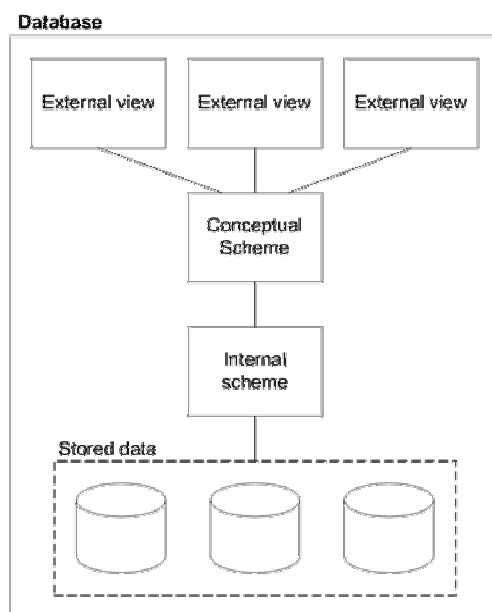
### 8.1.1 Databasestruktur

Der findes en række forskellige måder hvorpå et DBMS kan opbygges. Her er det valgt at beskrive opbygningen ud fra det Amerikanske Institut for standarder, ANSI/SPARC (American National Standards Institute/Standards Planning and Requirements Committee), som opfatter en database som en måde at vise data. Fremgangsmåden, som ANSI/SPARC foreslår, er en arkitektur med tre niveauer [Date, 2000] [Worboys, 1995] [Petersen, 1992]:

- Eksterne niveau (external view)
- Konceptuelle niveau (conceptual scheme)
- Interne niveau (internal scheme)

På den måde er der en klar opdeling mellem meningen med informationen, brugerens syn på data og den måde data fysisk er struktureret. Herved opnår databasen den fleksibilitet, der gør data applikationsuafhængig, og som er et vigtigt element i en database.

Disse tre niveauer er illustreret på figur 8-1 og beskrives nærmere i det følgende.



Figur 8-1 Arkitektur for databaser [Worboys].

Uafhængigheden opnås, som sagt bl.a. ved at adskille det interne niveau fra det konceptuelle niveau. Det interne niveau håndterer data og implementeringsdetaljerne, og fortæller hvilke data der er gemt og hvordan. Effektiviteten er det vigtigste på dette niveau og datastrukturen er valgt ud fra kriterier om at databasen skal være hurtig og effektiv.

Det konceptuelle niveau beskriver datamodellen for hele databasen uden at komme ind på den fysiske implementering. På det konceptuelle niveau er der indeholdt en beskrivelse af hvilke informationer, der skal repræsenteres i databasen.

En database kan have flere brugere på samme tid, der hver især har forskellige krav og behov. Derfor findes der på figur 8-1 flere eksterne niveauer, der repræsenterer hver enkelt brugers tilgang til databasen. Generelt har brugerne af databasen kun behov for at anvende dele af databasen og har forskellige adgangsbehov. Det kan f.eks. være at en bruger kun skal have mulighed for at læse databasen, mens en anden skal kunne editere i data [Worboys, 1995] [Date, 2000].

De tre niveauer kan ændres uafhængig af hinanden, og gør det muligt angive at en logisk beskrivelse af databasen, uden det er nødvendigt at specificere den fysiske struktur (physical data independence). Samtidig er det muligt at ændre den konceptuelle opbygning, uden det

har betydning for det eksterne niveau og brugerne af databasen (logical data independence) [Date, 2000].

## 8.2 Databasetyper

Der findes en række forskellige typer databasesystemer f.eks. hierarkiske, netværks-, relationelle og objektorienterede databaser, hvor den relationelle database er den mest udbredte. Hierarkiske databaser og netværksdatabaser benyttes enkelte steder, men de er ikke særlig fleksible i den måde der navigeres rundt i databasen, og beskrives ikke nærmere i denne rapport, men yderligere information kan findes i [Date, 2000]. Objektorienterede databaser er, i forhold til de andre nævnte typer, en nyere form for databasesystem, der har udviklet sig meget gennem de sidste ti år. I det følgende beskrives relationelle databaser efterfulgt af en beskrivelse af objektorienterede databaser. De seneste år er endnu en ny type databaser kommet frem, nemlig objekt relationelle databaser, der er en udvidelse af relationelle databaser. Denne type beskrives ligeledes, da den indeholder en række fordele for geografiske data.

### 8.2.1 Relationelle databaser

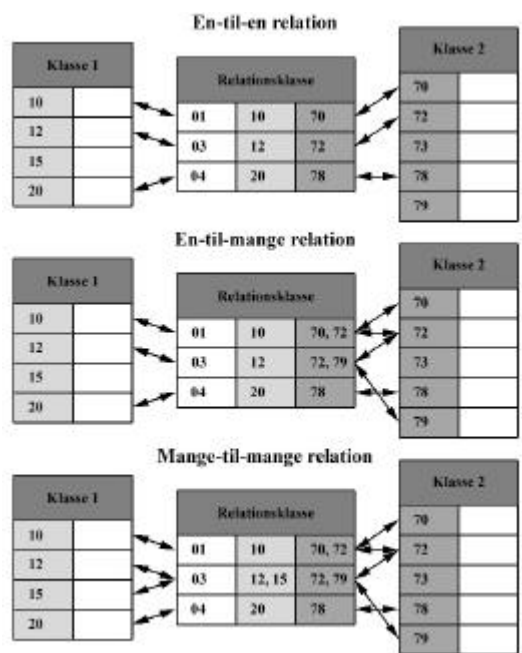
De relationelle databaser (RDB) har domineret markedet for databaser siden 1970'erne, og der er delte meninger, om det vil ændre sig i den nærmeste fremtid [Date, 2000] [Leavitt, 2001]. Der er dog en del andre systemer der er på vej frem, men det beskrives i et senere afsnit.

Grunden til at relationelle databasesystemer er de dominerende skyldes hovedsageligt, at de er baseret på logik og matematik [Date, 2000], at strukturen er meget simpel [Worboys, 1995]. I en relationel database opfattes data i databasen af brugeren som rækker (rows, records eller tuples) og kolonner (columns) i en eller flere tabeller. Hver tabel har et navn og indeholder et endeligt antal kolonner. En kolonne i en tabel har ligeledes et navn og indeholder et felt (field) for hver række. Hver kolonne indeholder en bestemt datatype, f.eks. heltal, der gælder for alle felter i den pågældende kolonne. I den traditionelle relationelle database kan hvert felt i en tabel kun indeholde en simpel værdi eller en relation. Dette er en begrænsning, der er blevet rettet op på i den sidste standard for relationelle databaser, SQL3 [Shekhar & Chawla, 2003], og beskrives nærmere senere i afsnittet om de objekt relationelle databaser.

Ud over tabeller med data består en relationel database også af relationer mellem forskellige tabeller. Relationer kan betegnes som en nøgle mellem en værdi/attribut i en tabel og en anden værdi/attribut i



en anden tabel dvs. en en-til-en relation, men det kan også være en-til-mange, mange-til-en eller mange-til-mange relationer. Når en relation refererer til flere rækker i en anden tabel er det nødvendigt med en mellemliggende tabel, der indeholder de værdier, der henvises til. Eksempler på de forskellige relationer i en relationel database er illustreret på figur 8-2. Det ses, at en relation er en tabel med et antal kolonner og et antal rækker, svarende til antallet af relationer.



Figur 8-2 De forskellige typer relationer; en-til-en, en-til-mange, og mange-til-mange.

Som beskrevet, er en relation ikke andet end en struktureret tabel, der indeholder referencer til to andre tabeller. Ud over relationer understøtter relationelle databaser en række simple funktioner eller operationer, der kan foretages på tabellerne i databasen. De enkelte funktioners engelske betegnelser er: *union*, *join*, *divide*, *restrict*, *project*, *intersection* og *difference* og de foretager forskellige former for handlinger på én eller flere tabeller [Worboys, 1995]. Disse funktioner forklares ikke nærmere her, men er angivet for at indikere at simple operationer findes i relationelle databaser.

Den metode, der er beskrevet ovenfor, er som sagt den måde hvormed brugeren ser data, men den fysiske opbevaringsmetode for data afhænger af hvilken database der benyttes. Det kan f.eks. være ved brug af sekventielle filer, pointers osv. hvor det eneste krav er, at det

kan præsenteres som en tabel på det eksterne niveau [Date, 2000]. Dette betyder at tabellerne repræsenterer en abstraktion af den måde, data fysisk er gemt i det interne niveau.

Til at håndtere de relationelle databaser og foretage operationer på sådanne databaser findes der en lang række relationelle database management systemer, RDBMS. Nogle af de mest benyttede databasesystemer er f.eks. Microsoft Access, IBM DB/2 og Oracle 7.x.

Relationelle databaser er gode til at håndtere store mængder strukturerede tal og bogstaver, men er begrænset af, at de kun kan indeholde simple datatyper og ikke brugerdefinerede datatyper [Leavitt, 2001]. Derfor arbejdes der på at udbrede brugen af objektorienterede databaser og derved fjerne de problemer, der er ved relationelle databaser.

### 8.2.2 Objektorienterede databaser

Relationelle databaser er tilstrækkeligt til at strukturere og håndtere data, og benyttes i en lang række applikationer, til mange former for analyser. Til håndtering af data, der har en kompliceret understruktur, og som er organiseret f.eks. hierarkisk, er den relationelle database ikke fyldestgørende. Dette er som regel tilfældet ved geografiske data, hvor de forskellige rumlige objekter i sig selv er sammensatte [Worboys, 1995]. Et alternativ er da at benytte et objektorienteret database management system (OODBMS).

Brugen af OODBMS har sit udspring i kombinationen af principperne i databasestyling og de objektorienterede programmeringssprog som f.eks. Java, Smalltalk og C++ [Date, 2000] [Worboys, 1995] [Obasanjo, 2001]. Objektorienterede databaser er altså databaser, der kan håndtere objektorienterede datamodeller og brugerdefinerede typer. Dette indbefatter, at denne form for databaser adskiller sig fra den traditionelle relationelle database, ved at det bl.a. er muligt at nedarve klasser og tilføje adfærd til klasser i form af funktioner, som beskrevet i det foregående kapitel.

Ved at datamodellen i databasen er identisk med datamodellen i det program, der benytter databasen, vil etableringen af databasen være lettere end ved en relationel database. Dette gør, at det ikke er nødvendigt at foretage en oversættelse, mellem den måde databaseprogrammet bruger data, og den måde data er opstillet. Herved gøres datastrukturen mere naturlig, mængden af kildekode reduceres og det er lettere at vedligeholde og genbruge denne kildekode [Leavitt, 2001] [Worboys, 1995].

Der er i skrivende stund ingen bred enighed om en standard for hvad et OODBMS består af og udviklingen er stadig på et tidligt stadie. Der eksisterer dog en række databaseproducenter, der har udviklet systemer til objektorienterede databaser. Her kan f.eks. nævnes Versant, O2 og Gemstone. Derimod er udbredelsen af databasesystemer, der er en kombination af de objektorienterede metoder og relationelle databaser i en kraftig vækst og denne form for databasesystemer beskrives i det følgende.

### 8.2.3 Objekt relationelle databaser

Der er to ting der er med til at reducere udviklingen af de objektorienterede databaser. Det første er at brugen af dem ikke er særlig udbredt, selvom der efterhånden findes en række forskellige produkter, der kan benyttes. Dette gør, at der ikke er de nødvendige økonomiske ressourcer og vilje til at gøre denne type produkter bedre. Det vil sige at udviklingen er inde i en ond cirkel, der skal brydes, for at udbredelsen af de objektorienterede databaser kan blive større [Shekhar & Chawla, 2003].

Det andet problem er, at interaktionssproget SQL er det mest brugte sprog, når det handler om databaser, og SQL hænger i høj grad sammen med relationelle databaser. I SQL specificerer brugeren det ønskede resultat, og ikke hvordan det skal fremkomme [Shekhar & Chawla, 2003]. I stedet ligger denne opgave inde i nogle funktioner i databaseprogrammet (DBMS) og kaldes fra SQL. Dette gør at databasen er afhængig af DBMS'en.

En af de store forskelle mellem RDBMS og OODBMS er, at SQL ikke kan håndtere komplekse datatyper, men kun de traditionelle typer heltal osv. Der er dog sket en stor udvikling inden for de seneste år. I den seneste version af specifikationen for SQL, SQL3 er det muligt at arbejde med ADT og andre principper fra den objektorienterede verden. Dette har udviklet sig til en generation af databaser kaldet objekt relationelle databaser (ORDBMS).

Objekt relationelle databasesystemer er en udvidelse til de relationelle databaser, og bygger et objektorienteret lag oven på en relationel database. Dette er bl.a. gjort i udvidelsen til IBM's DB/2 (Universal Database), Unisys' OSMOS, Oracle 8.x og Informix's Universal Server.

Fordelen ved ORDBMS er, at den understøtter brugerdefinerede datatyper, hvilket den traditionelle RDBMS ikke gør. Dette er en stor fordel, når der arbejdes med utraditionelle databaser som f.eks. geografiske databaser, hvor komplekse datatype som f.eks. punktet,

benyttes. At definere sådanne typer korrekt, stiller store krav til databaseudviklerne og derfor har en række producenter udviklet produktpakker, der kan hjælpe brugerne. Oracle har f.eks. udviklet sådan en produktpakke [Shekhar & Chawla, 2003]. Ligeledes har Environmental System Research Institute (ESRI) sådan en produktpakke, i form af ArcGIS, der bl.a. arbejder med en objektorienteret datamodel der kaldes Geodatabasen [Pedersen, 2000]. Geodatabasen beskrives nærmere i det følgende.

#### 8.2.3.1 Geodatabasen

Geodatabasen er en objektorienteret datamodel introduceret af ESRI og kan benyttes i deres programpakke ArcGIS. Formålet med Geodatabasen er bl.a. at gøre det muligt at tilføje og knytte adfærd til de enkelte objekter på dataniveau i et GIS-datasæt [Zeiler 2000].

Geodatabasen fungerer som en objektorienteret database, men er egentlig opbygget i en relationel database med relaterede tabeller [Zeiler 2000]. For brugeren er denne relationelle del skjult og ArcGIS styrer strukturen og sammenhængen mellem databasens tabeller og præsenterer derved en objektorienteret geografisk datamodel. Herved vil brugeren ikke have adgang til det interne niveau og vil heller ikke have glæde af denne mulighed. Det er dog muligt ved hjælp af ArcGIS at foretage ændringer og forbedre datamodellen i Geodatabasen. På den måde bringes brugeren tættere på det interne niveau, og dér hvor data fysisk gemmes.

Der findes en lang række fordele ved at anvende ArcGIS og Geodatabasen, dog også en del ulemper. En af de største fordele er som sagt at det er muligt at benytte objektorienteret datamodellering, herunder anvendelse af brugerdefinerede objekttyper. Derudover er det muligt at tilføje funktionaliteter, der bl.a. gør det muligt at kontrollere data ud fra en række opstillede regler [Zeiler 2000].

En anden egenskab ved ArcGIS, der er en stor fordel, er at programmet, ud over geodatabasen, også kan håndtere en lang række andre formater, der ikke er objekt relationelle som Geodatabasen. Det er f.eks. muligt at benytte shapefiler, der også er et format til at gemme rumlige informationer og som blev introduceret af ESRI for en del år siden, samt en lang række andre tilsvarende formater [Zeiler 2000]. Denne mulighed gør, at udbredelsen af programmet vil foregå hurtigere, end hvis det kun kunne håndtere Geodatabasen. Herved har brugerne større mulighed for at skifte fra andre produkter til ArcGIS. Det største problem er, at produktet er relativt dyrt i forhold til tidligere produkter fra ESRI.

Fremgangsmåden for hvordan en geodatabase etableres i ArcGIS er beskrevet i Appendiks 2 og 3. Der er to måder at gøre dette på, hvor den ene er en metode, der benytter en række værktøjer, der findes i ArcGIS. Den anden metode er mere kompliceret, og foregår ved at datamodellen modelleres i en UML-model i programmet Microsoft Visio, og eksporteres til en database, hvorfra modellen kan importeres i ArcGIS. Fordelen ved denne metode er, at der kan opbygges en mere kompleks datamodel, og der kan tilføjes et vilkårligt antal brugerdefinerede funktioner vha. metoder til hver objektklasse. Metoderne kan bl.a. anvendes til verificering af om objekterne overholder en række opstillede regler, som tidligere beskrevet. Fremgangsmåden for de to metoder er beskrevet i Appendiks 2 og 3.

### **8.2.4 Afslutning**

Brugen af relationelle databaser og objektorienterede databaser er meget forskellige, men samtidig indeholder de alligevel en række ligheder. En tabel i en relationel database kan sidestilles med en klasse i en objektorienteret database og på samme måde kan en række sammenlignes med en instans af en klasse. En række i en tabel i en relationel database har dog ikke mulighed for at indeholde en adfærd, som en instans af en klasse kan. Kolonner i en tabel i en relationel database kan sammenlignes med en classes attributter, men kan kun indeholde primitive datatyper.

Der stilles store krav til rumlige databaser. Et af de krav, der stilles, når der anvendes GIS og geografiske data er, at det skal være muligt at søge og analysere geografiske placeringer af de rumlige objekter.

Den relationelle model er altså god til at håndtere f.eks. topologiske relationer men er derimod ikke så anvendelig til at repræsentere komplekse hierarkiske relationer. Både topologiske og hierarkiske relationer kan håndteres i en objektorienteret model [Shekhar & Chawla, 2003]. I det følgende kapitel beskrives bl.a. de topologiske relationer mellem rumlige objekter.

## 9 Modeller

Langt de fleste GIS er baserede på relationelle databaser, men udbredelsen af objektorienterede databaser er voksende, som tidligere beskrevet. Det betyder at data enten er organiseret som en samling af rækker, eller hvis der er tale om objektorienterede databaser, objekter. Hver af disse rækker eller objekter indeholder en identifikation samt værdier, der f.eks. kan være tal, tekststreng, tidsangivelser eller koordinater. En stor samling af disse rækker eller en stor samling objekter, vil indeholde en omfattende mængde information, og kan derfor være svær at overskue. For at strukturere data benyttes der normalt to forskellige modelleringsmetoder for rumlig information, feltbaserede modeller og objektbaserede modeller, og der er fundamental forskel på de to modeltyper.

De feltbaserede modeller består af en rumlig struktur, hvori der til hvert element i strukturen er knyttet nogle værdier. Et ortofoto eller en digital højdemodel kan være eksempler på feltbaserede modeller. Feltbaserede modeller består oftest af regulære net, men kan også bestå af data i et irregulært net, som f.eks. TIN-modeller [Worboys, 1995].

De objektbaserede modeller baserer sig på objekter, hvortil der er knyttet forskellige attributter, hvoraf nogle kan være rumlige. Top10dk's geometridel er et eksempel på en objektbaseret model, mens den tilhørende digitale højdemodel er et eksempel på en feltbaseret model. Den væsentligste forskel ligger i, at de feltbaserede modeller består af en rumlig struktur, hvortil der er knyttet attributter, hvorimod de objektorienterede modeller består af en række objekter, der har referencer til den rumlige dimension. Disse to modeltyper er udtryk for to forskellige grundlæggende teorier, der kan sammenlignes med teorien om lysets fysik. Det kan enten opfattes som partikler (objektbaseret) eller bølger (feltbaseret) [Worboys, 1995].

Det er forskelligt, hvilken slags model, det er hensigtsmæssigt at benytte, afhængig af hvad der skal modelleres. Hvis det, der skal modelleres, er uden bestemt form, eller har glidende og kontinuerte overgange, er det oftest den feltbaserede model der benyttes. Feltbaserede modeller bliver i udstrakt grad brugt i forbindelse med fly- og satellitmålinger. Objektmodeller benyttes i højere grad til at modellere faste strukturer, der er lette at identificere, som for eksempel vejnet, matrikler og bygninger.

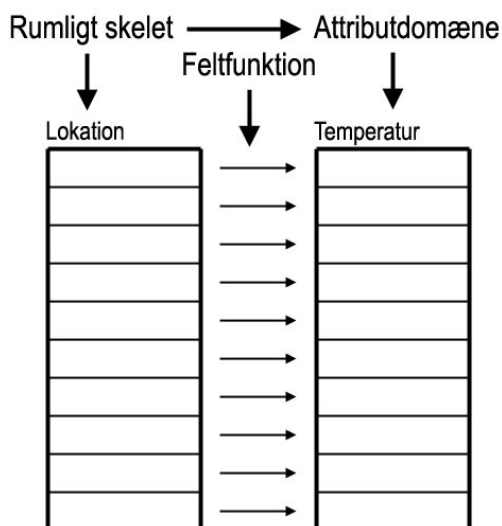
Feltbaserede modeller har mere karakter af at være rå data, mens de objektbaserede modeller i højere grad er data, der er fortolket og systematiseret. I det følgende vil de to modeller blive gennemgået.

## 9.1 Feltbaserede modeller

En feltbaseret model går grundlæggende ud fra, at den variabel der modelleres, har en værdi overalt i rummet. Det betyder i princippet, at der skal lagres uendelige mængder information. Da dette ikke er muligt, er det nødvendigt at foretage en generalisering, og gemme data i et endeligt antal punkter i rummet, og derved at begrænse mængden af data. Denne struktur af punkter kaldes det rumlige skelet (spatial framework). Punkter i skelettet kan være regulært fordelt i et net, f.eks. i det todimensionale euklidiske rum med punkter for hver enhed ud af begge akser, eller irregulært fordelt, f.eks. med punkter på de steder, hvor observationer er foretaget, uden en fast struktur. Mange luftbårne systemer har sensorer, der opsamler data i et regulært net. Hvis data er indsamlet irregulært, er det nødvendigt at interpolere, for at tilpasse det til et regulært net. En af fordelene ved at benytte et regulært net er, at det ikke er nødvendigt at lagre koordinatinformation for de enkelte punkter i nettet. I stedet angives et startpunkt, orienteringen af nettet, samt intervallerne mellem punkterne ud af de to akser.

Somme tider kan det være fordelagtigt at lagre data i et irregulært skelet. Dels kan repræsentationen blive mere præcis, dels kan der spares plads, da det kun er nødvendigt at have en høj punkttæthed i de dele af data, hvor der forekommer store variationer.

De feltbaserede modeller bygger på tre grundlæggende ting; det rumlige skelet  $S$ , en feltfunktion  $f()$ , og et attributdomæne  $A$ , som vist på figur 9-1.



Figur 9-1. En feltbaseret model.

Det rumlige skelet er et afgrænset net, der er lagt hen over et udsnit af den verden, der ønskes modelleret. Dette net kan f.eks. være baseret på et euklidisk plan, hvor nettet er udspændt regulært over dette. Et andet kendt rumligt skelet, er længde- og breddegradssystemet, der refererer til jordens overflade.

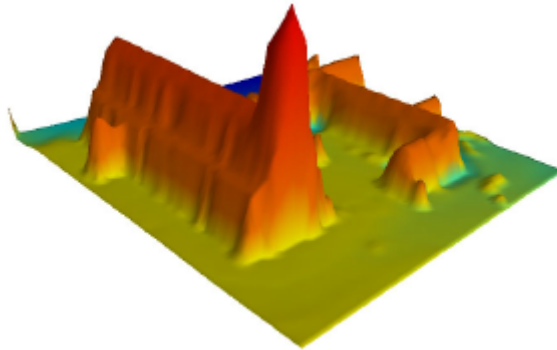
Attributdomænet  $A_i$  er en række værdier, der skal afbildes i feltmodellen. Det kan f.eks. være reelle tal, der viser temperaturen, eller en klassificering, eksempelvis ”mark”, ”skov” eller ”bebygget areal”. Feltet  $f_i$  eller feltfunktionen er da en funktion, der tildeler det rumlige skelet værdier fra attributdomænet. En feltbaseret model kan indeholde  $n$  sådanne felter:

$$f_i(S) = A_i \quad \text{for } 1 \leq i \leq n$$

for ethvert punkt i det rumlige skelet [Shekhar og Chawla, 2003].

På figur 9-2 ses en visualisering af en højdemodel af en del af Odense, skannet med en flybåren laserscanner, i et irregulært net. Disse målinger bliver filtreret og resamplet i et regulært net, i dette tilfælde 1x1 meter. Denne form for data er et typisk eksempel på en feltbaseret datamodel. Der er opbygget et rumligt skelet, idet et område er inddelt i et afgrænset net af felter. De enkelte dele af nettet associeres med en eller flere attributter, hvor det i laserdataeksemplet, er den målte højde i meter over en geoide, dvs. at feltfunktionen modellerer jordens overflade. Feltfunktionen kunne også modellere intensiteten af retursignalet.





Figur 9-2. Visualisering af en feltbaseret model.  
[Olsen et al., 2000]

Formålet med data er at få adgang til information om det observerede fænomen. Jo mere information, der kan udvindes af data, jo mere værdi kan det siges at have. Derfor er det nødvendigt med nogle operationer, der kan udvinde information fra data. I det følgende beskrives disse operationer.

### 9.1.1 Operationer på felter

For at foretage en analyse på flere felter samtidig, er det nødvendigt, at de har det samme rumlige skelet. Hvis en analyse ønskes foretaget på to felter, der ikke har det samme rumlige skelet, er det nødvendigt at foretage en resampling af det ene datasæt, således at data bliver transformeret fra det ene rumlige skelet til et andet. Der findes en række forskellige metoder til at foretage denne resampling, men det vil ikke blive beskrevet her, og der henvises eksempelvis til [Wolf & Dewitt, 2000].

Der findes metoder, der kan benyttes til at analysere den information, der ligger i felterne. I det følgende vil et udvalg af disse blive beskrevet.

#### 9.1.1.1 punktbaserede operationer

De punktbaserede operationer beregner en ny værdi i et nyt output-felt i det rumlige skelet ud fra værdien af netop dette punkt i input-felterne. Som eksempel ses på en simplificeret model, hvor der indgår to felter, der har det samme to-dimensionelle rumlige skelet, og der hver især udtrykkes ved funktionerne

$$f(x, y) = \begin{cases} 1 & \text{for } (x, y) = \text{"Skov"} \\ 0 & \text{ellers} \end{cases}$$

$$g(x, y) = \begin{cases} 1 & \text{for}(x, y) = \text{"mark"} \\ 0 & \text{ellers} \end{cases}$$

En lokal operation på disse to felter kan være sum, der udtrykkes ved:

$$(f + g)(x, y) = \begin{cases} 1 & \text{for}(x, y) = \text{"skov"} \text{ eller } \text{"mark"} \\ 0 & \text{ellers} \end{cases}$$

Dette sker simpelt ved at lægge summen af værdierne i hvert enkelt punkt i inputfelterne i det tilsvarende punkt i outputfeltet. Der er dog mulighed for, at der opstår fejl, hvis der i samme punkt er registreret både skov og mark. Dette er en fejl i inputdata, og derfor kan dette punkt evt. gives en fejlkodet værdi.

Af andre lokale operationer kan nævnes punktvis maksimum, minimum og differens.

#### 9.1.1.2 Områdebaserede operationer

I modsætning til de lokale operationer, der opererer på værdierne i netop et punkt i inputfelterne, analyserer de områdebaserede operationer også de punkter, der ligger i nærheden af punktet. Dvs. at værdier i nærheden af punktet influerer på den afledte værdi i outputfeltet. Det antages at der findes et fokus-punkt  $p$  i det rumlige skelet  $F$ . Desuden eksisterer der en funktion,  $n()$ , der definerer de punkter, der ligger i nærheden af  $p$ , samt en feltfunktion  $f()$ . Operationen udfører følgende:

- 1) Beregn  $n(p)$  dvs. bestem de punkter, der ligger i nærheden af  $p$  (dette inkluderer for det meste også  $p$  selv)
- 2) Bestem  $f(n(p))$  i de udvalgte punkter, altså beregn værdierne i de udvalgte punkter  $n(p)$  vha. feltfunktionen.
- 3) Afled en værdi i outputfeltet ud fra de værdier, der er bestemt i trin 2. Denne værdi beregnes alt efter hvilken type områdebaseret operation, der benyttes.

Et eksempel på en områdebaseret operation er beregning af hældninger i en digital højdemodel, eller vægtet gennemsnit af punkterne i nærheden af fokus-punktet. Inden for billedbehandling har brugen af områdebaserede operationer stor udbredelse [Carstensen, 2000].

### 9.1.1.3 Zonebaserede operationer

Det kan være hensigtsmæssigt at opdele et felt i zoner, dvs. områder, der hver især har nogle fælles karakteristika baseret på feltfunktionen. Et eksempel på hvordan en sådan inddeling i zoner kan foretages, ses nedenfor:

$$\text{Hvis } f(p) = 0 \quad \Rightarrow \quad p \in \text{zone A}$$

$$\text{Hvis } 0 < f(p) < 5 \quad \Rightarrow \quad p \in \text{zone B}$$

$$\text{Hvis } f(p) \geq 5 \quad \Rightarrow \quad p \in \text{zone C}$$

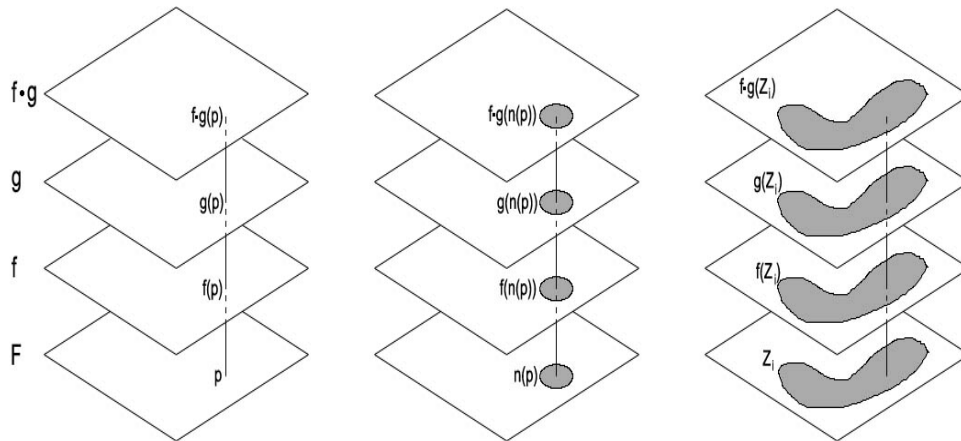
Punkterne deles ind i zoner afhængigt af feltets værdi i punktet. Denne opdeling gør det bl.a. muligt at foretage analyser på dele af feltet, der har fællestræk, men det kan også være nyttigt at opdele et felt i zoner, f.eks. i forbindelse med luftbårne målinger, hvor der kan foretages en helt eller delvis automatisk klassificering af de arealer, der er målt.

Zonebaserede operationer er netop analyser, der baserer sig på zoneopdelte felter. De fungerer i høj grad ligesom områdebaserede operationer:

- 1) Find den zone  $Z_i$  som  $p$  tilhører.
- 2) Bestem  $f(p_i)$  for alle  $p_i \in Z_i$ , i de punkter der er med i  $Z_i$ , altså beregn værdierne for punkterne i zonen vha. feltfunktionen.
- 3) Afled en værdi i outputfeltet ud fra de værdier, der er bestemt i trin 2. Denne værdi beregnes alt efter hvilken type zonebaseret operation der benyttes.

Der er en række interessante zonebaserede operationer, f.eks. gennemsnitsværdi i zonen eller sum af alle værdier i zonen. Zonen kan også bruges som et filter til at afgrænse et andet felt. Zonebaserede operationer benyttes eksempelvis til temperaturkort og administrative opdelinger [Worboys, 1995].

De tre forskellige operationstyper er illustreret på figur 9-3.



Figur 9-3: De tre forskellige operationer; punktbase-rede-, områdebase-rede- og zonebase-rede operationer. Nederst ses det rumlige skelet  $F$ , i midten to feltfunktioner, der benyttes som input, og øverst det resulterende output felt.

De feltbase-rede modeller skal ses som et alternativ til de objektbase-rede modeller, der beskrives i det følgende.

## 9.2 Objektbase-rede modeller

Frem for at opdele verden i en endelig rumlig struktur og tildele værdier til denne struktur, som det gøres i de feltbase-rede modeller, består en objektbase-ret model af en lang række objekter, der hver især har en reference til et rumligt objekt. Disse objekter modellerer entiteter i den virkelige verden, og er defineret som:

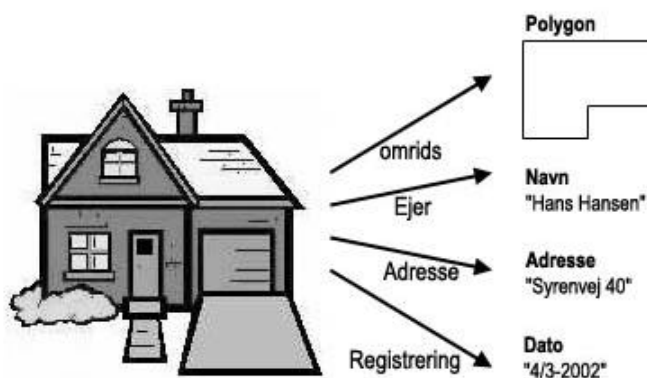
- Identificerbare (kan registreres)
- Relevante (har interesse)
- Mulige at beskrive (har visse karakteristika)

I praksis kan alting være et objekt [Date, 2000], men for at objekterne skal have værdi for brugeren af modellen, skal de entiteter, som objekterne modellerer, overholde de ovenstående kriterier. Entiteterne har en række egenskaber, som det er nødvendigt at repræsentere i det tilsvarende objekt:

- **Attributter**
- **Adfærdsmæssige egenskaber**
- **Strukturelle egenskaber**

I objektet modelleres disse egenskaber vha. en række attributter. Disse attributter kan deles op i to grundlæggende typer; værdier og variable. Værdier er simple objekter, der altid benyttes i deres oprindelige form, dvs. der ikke bliver nedarvet fra dem. Det kan f.eks. være heltal (7, 60) eller tekststreng ("København"). Variable er komplekse objekter, der er foranderlige, dvs. at de er klasser for sig selv. Det kan f.eks. være klassen "polygon" eller klassen "person" [Worboys, 1995].

For objekter i en rumlig model, er det muligt at dele attributterne op i fire grundlæggende grupper. Rumlige, grafiske, tidsmæssige og alfanumeriske. De rumlige attributter er repræsentationen af den virkelige entitets geometri i objektet. De grafiske attributter er metoder til at visualisere de rumlige attributter på skærmen eller på en printer. De tidsmæssige attributter lagrer vigtige tidspunkter, som f.eks. hvornår en bygning er opført, eller hvornår den er registreret i datasættet. De alfanumeriske attributter beskriver andre vigtige egenskaber, f.eks. en bygnings adresse, vandforbrug eller ejers navn. På figur 9-4 ses et eksempel på en entitet fra den virkelige verden (en bygning) og de relevante egenskaber, omrids, ejer, adresse og registreringsdato. Længst til højre på figuren ses de attributter, der er lagret i databasen.



Figur 9-4 Et objekt og dets attributter.

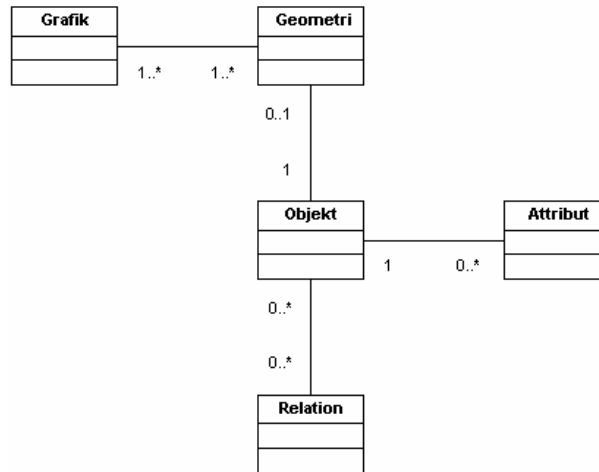
Ethvert objekt er af en bestemt klasse. En objektklasse har tilknyttet nogle funktioner eller metoder, der kan benyttes i tilknytning til

ethvert objekt af den pågældende klasse, f.eks. til at hente eller sætte attributter, eller foretage beregninger. Dette blev beskrevet i kapitel 7.

Som nævnt ovenfor, kan et objekt både have rumlige og grafiske attributter tilknyttet, og det er vigtigt, at der skelnes mellem disse to. De rumlige attributter eksisterer udelukkende for at modellere det, der nu er meningen der skal modelleres, mens de grafiske attributter er de dele, der repræsenterer objektet. Ved transformationen fra det rumlige til det grafiske objekt, sker der ofte en abstraktion og/eller generalisering. F.eks. kan en bygning have en højt detaljeret polygon som geometri. Den grafiske repræsentation ændres alt efter hvilket målestoksforhold, objektet vises ved. Ved målestoksforhold mindre end 1:10000 skal den grafiske repræsentation være lig den rumlige repræsentation, ved 1:10000 til 1:20000 en simplificeret polygon, og ved højere et punkt [Shekhar & Chawla, 2003].

De fleste objekter i en objektbaseret model til et GIS er objekter med en rumlig reference, f.eks. et bygningsobjekt med en tilknyttet polygon som bygningsafgrænsning. Bygningen er et rumligt refererende objekt, mens polygonen er et rumligt objekt. Det er vigtigt at gøre sig denne forskel klart. Et hus er ikke en underklasse af en polygon, men den har en attribut, der hedder ”bygningsafgrænsning”, der er af typen polygon [Worboys, 1995].

På figur 9-5 ses en model af et typisk objekt i et GIS. Objektet repræsenteres af en geometri, der har en eller flere grafiske repræsentationer. Geometrien er et rumligt objekt, som f.eks. en polygon, en linie eller et punkt. Dette rumlige objekt kan repræsenteres på forskellige måder afhængigt af applikation, målestok, anvendelse eller lignende ting, som beskrevet tidligere. Ud over geometrien kan objektet have nogle attributter, der ikke er rumlige. Desuden kan objektet have en række relationer til andre objekter, idet forskellige objekter kan have rumlige forbindelser imellem dem.



Figur 9-5 Model af et rumligt objekt.

Den geometri, der er en del af et rumligt refererende objekt, kaldes som tidligere nævnt for rumlige objekter. I det følgende vil de mest almindelige rumlige objekter blive beskrevet.

### 9.2.1 Rumlige objekter

På figur 9-6 ses et nedarvnings-hierarki for de mest almindelige rumlige objekter i UML-notation. Der er mange forskellige bud på hvordan denne opbygning skal se ud, men her er det forsøgt at beskrive en model, der er overskuelig og samtidig omfattende, og er baseret på de gængse opfattelser [Worboys, 1995] [OpenGIS, 1999].





En linie er altså en samling af liniestykker, der ligger i forlængelse af hinanden, således at for liniestykket  $x$ , der er en del af linien, vil endepunktet  $b_x$  være fælles med liniestykket  $x+1$ 's startpunkt  $a_{x+1}$ . Dette gælder for samtlige liniestykker i linien. En linie kan eksempelvis repræsentere en vej eller en flod. En ring er en specialisering af objektclassen Linie, hvor start- og endepunkt er det samme.

En polygon er en geometrisk figur, der har udstrækning i to dimensioner. Den består af én ydre ring og et vilkårligt antal indre ringe, der fungerer som huller i polygonen. De indre ringe må som regel ikke røre kanten af polygonen. Polygoner kan bruges til at repræsentere ting som bygningsafgrænsninger, matrikler eller søer.

Den sidste klasse er Geometrisamling, der kan indeholde forskellige andre geometriske objekter. Det kan f.eks. bruges når der foretages forskellige operationer eller analyser på objekter, og resultatet ikke er et enkelt af de tidligere nævnte objekter, men består af flere dele, f.eks. to polygoner.

De rumlige objekter er en af de vigtigste dele af et objektbaseret GIS. Det er jo netop disse objekter, der differentierer de geografiske informationssystemer fra andre informationssystemer. Det er ikke kun inden for GIS at der benyttes rumlige objekter. Også i CAD-programmer er rumlige objekter en vigtig del, og der er da også en flydende overgang mellem GIS- og CAD-data. Dér hvor forskellen ligger, er at GIS fokuserer på data, der refererer til et objekt i rummet, hvor et CAD-system i højere grad fokuserer på det rumlige objekt, og ikke på data [Worboys, 1995]. I både GIS og CAD-systemer er det muligt at udføre forskellige operationer på de rumlige objekter. I det følgende vil disse operationer blive gennemgået.

### 9.2.2 Operationer på rumlige objekter

Begrebet "et rumligt objekt" kan virke lidt diffust, for hvad er egentlig dette rum, som objektet befinder sig i? Et rumligt objekt eksisterer i rummet, idet dette rum definerer den relation, der findes mellem de forskellige objekter [Worboys, 1995]. Sagt på en anden måde er rummet en struktur eller beholder, hvori ting eksisterer.

Det mest almindelige rum er det euklidiske rum. Et euklidisk rum er en todimensional model, hvor koordinaterne er bygget op ud fra et fast punkt (origo) samt to på hinanden vinkelrette linier (akser), der skærer hinanden i dette punkt. Et punkt i dette rum, er et unikt par reelle tal  $(x,y)$ , der angiver punktets afstand til origo ud af de to akser. Det kan ofte være hensigtsmæssigt at betragte et punkt som en vektor, da der

derived kan foretages additioner, subtraktioner og multiplikationer på disse. To af disse vektorer kan da beskrive et liniestykke, og ét eller flere liniestykker kan repræsentere en linie, som beskrevet ovenfor.

#### 9.2.2.1 Transformationer af rumlige objekter.

En transformation er en proces, hvor et objekt påvirkes, således at nogle af dets rumlige egenskaber bliver ændrede, mens andre bevares. Det er forskelligt hvilke egenskaber, der bevares for de objekter, der gennemgår transformationen, alt efter typen af transformation. Herunder beskrives en række af de mest almindelige typer:

- *Euklidiske transformationer*: Bevarer form og størrelse på objekterne. Translation er et eksempel på en Euklidisk transformation.
- *Formbevarende transformationer*: Bevarer formen på objekterne. Et eksempel på en formbevarende transformation er skalering. Euklidiske transformationer er også formbevarende.
- *Affine transformationer*: Bevarer de affine egenskaber for objekterne, som for f.eks. parallelisme. Affine transformationer kan eksempelvis være rotationer, spejlinger og forskydninger. Formbevarende transformationer er også affine.
- *Projektive transformationer*: bevarer de projektive egenskaber for objekterne. Projektive egenskaber kan forklares som de figurer der kan fremkomme, hvis et objekts geometri bliver projiceret fra en punktformet lyskilde op på et lærred, f.eks. kan en cirkel projicering være en ellipse. Affine transformationer er også projektive.
- *Topologiske transformationer*: De topologiske egenskaber bevares for objektet eller objekterne, f.eks. at to liniers endepunkter er sammenfaldende både før og efter transformationen. Senere beskrives flere topologiske relationer mellem de forskellige rumlige objekter.

De ovenfor beskrevne transformationer grupperer nogle af de operationer, der kan foretages på rumlige objekter. Disse operationer benyttes, f.eks. når objekter skal oprettes i databasen.

I det følgende beskrives en række topologiske relationer, der kan være nogen af de egenskaber, der skal bevares, hvis der foretages en transformation på objekter i databasen.

### 9.2.3 Topologiske relationer

Der findes forskellige metoder til at beskrive topologiske relationer mellem to objekter. En af metoderne, der bl.a. bruges af OpenGIS-konsortiet er beskrevet af Egenhofer [Egenhofer, 1994] [Worboys, 1995] [Shariff, Egenhofer og Mark, 1998], og kaldes for 9-skæringsmatricemodellen (9-intersection matrix model). Denne model udtrykker de mulige skæringer, der er mellem to rumlige objekter, X og Y. Selvom de to objekter ikke har den samme dimensionelle udstrækning, er det muligt at benytte denne model, f.eks. hvis de to objekter er en linie (udstrækning i én dimension) og en polygon (udstrækning i to dimensioner). Der skelnes mellem objekternes indre ( $^{\circ}$ ), deres afgrænsning ( $\partial$ ) og deres ydre (-). Skæringsmatricen udtrykkes da som:

$$I(X, Y) = \begin{pmatrix} ^{\circ}X \cap ^{\circ}Y & ^{\circ}X \cap \partial Y & ^{\circ}X \cap -Y \\ \partial X \cap ^{\circ}Y & \partial X \cap \partial Y & \partial X \cap -Y \\ -X \cap ^{\circ}Y & -X \cap \partial Y & -X \cap -Y \end{pmatrix}$$

Det første element øverst til venstre beskriver skæringen mellem X's indre og Y's indre osv. Elementerne i skæringsmatricen kan enten være sande eller falske, således at hvis der er skæring mellem de to elementer, står der 1, og hvis ikke, står der 0 i matricen. Dette betyder at der er  $2^9$  eller 512 muligheder for skæringer mellem to rumlige figurer. Relationerne er nummereret fra 0 til 511, idet de binære værdier i skæringsmatricen er omregnet til decimaltal, på den måde at værdierne aflæses fra oven mod højre.


De tre mest almindelige objekttyper er punkt, linie og polygon, som er beskrevet tidligere. Da et punkt ikke har nogen udstrækning, er det indre og kanten af et punkt det samme, og de to rækker eller søjler i matricen vil altså være identiske. Det er dog kun en begrænset del disse skæringer, der kan realiseres. For eksempel er skæringen, der er beskrevet ved en skæringsmatrice med 0 (nul) på samtlige pladser, ikke muligt, når der er tale om rumlige objekter i op til tre dimensioner. Ved denne skæring nr. nul, må der hverken være skæring mellem de to objekters indre, kant eller ydre, hvilket ikke er fysisk muligt. De mulige skæringsmatricer mellem de forskellige objektpar, bestående af punkt, linie og polygon, beskrives derimod i

Appendiks 1. I det følgende tages der udgangspunkt i disse mulige skæringsmatricer.

En linie har udstrækning i én dimension, og derfor er det indre af linien defineret som den del der ligger mellem start- og endepunkt. Kanten er start- og endepunktet, og det ydre er alt andet end det indre og kanten.

En polygon har en veldefineret grænse, der er defineret af den yderste ring. Her undersøges det kun polygoner uden indre ringe, men modellen kan også bruges til polygoner med indre ringe. Det indre er det der ligger inden for denne ring, og det ydre er det der ligger udenfor.

Fordelene ved denne metode er, at den kan implementeres i et program forholdsvis let, og den tager hensyn til mange af de muligheder, der er for relationer mellem to geometriske objekter. Desuden er det en overskuelig metode, f.eks. for en person, der skal definere hvilke topologiske relationer, der må være mellem to objekter, idet skæringsmatricen benyttes til at definere hvilke skæringer, der kan tillades. Der er dog også en række problemer med metoden, især når det ene objekt er en linie. Somme tider kan det være interessant at vide hvad to objekters fællesgeometri er, eller hvor mange gange de to rumlige objekter skærer hinanden, men det er ikke muligt at se vha. skæringsmatricer, hvor der kun er tale om binære værdier. Et eksempel på dette ses ved skæringsnummer 335 mellem to linier, vist på figur tabel 9-1. Her er det ikke muligt at se, ud fra skæringsmatricemodellen, hvilken type fællesgeometri, der er mellem de to linier. Skæringen kan både være et punkt og en linie.

Nr.	Eksempel	Skæringsmatrice
335		$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

Tabel 9-1 Skæringsmatrice for skæring nr. 135.

Udover dette, kan det være interessant om to linier krydser hinanden, således at der er dele af den ene linie på begge sider af den anden linie, eller om de bare rører hinanden i et punkt eller en linie.

Da det især er topologiske relationer, hvor en eller flere linier indgår, hvor disse problemer opstår, kan det være hensigtsmæssigt at etablere

nogle underklasser i skæringsundersøgelsen, hvor det eksempelvis er muligt at definere, hvilken type fællesgeometri to objekter kan have.

Det kan også være interessant at vide eller kunne definere, hvor mange gange to objekters elementer skærer hinanden. I den ovenstående model anvendes der binære værdier til at vise om to elementer skærer hinanden, men hvis der i stedet anvendes heltal, er det muligt at specificere antallet af tilladte skæringer. Ofte er det interessant at vide, om mere end en skæring er tilladt.

De topologiske relationer, kan bruges til at definere sammenhængen i den rumlige del af data, og dermed også til at finde fejl, de steder hvor de topologiske regler ikke overholdes, dvs. der hvor der forekommer topologiske relationer, der ikke er tilladte. Disse fejl kan opstå af forskellige grunde, der beskrives nærmere i det følgende kapitel.

## 10 Fejl

For at rette fejl i et datasæt, er det nødvendigt først at identificere dem. Dette kræver kendskab til, hvor fejlene kan opstå og hvilke typer fejl, der kan forekomme.

Fejl kan betegnes som de tilfælde, hvor det genererede data ikke stemmer overens med virkeligheden. Det kan enten være, at et objekt i datasættet ikke lever op til reglerne i specifikationen, eller at objektet ikke svarer til den virkelige verden.

### 10.1 Fejl i produktionsprocessen

I produktionsprocessen indsamles data i henhold til specifikationen, der beskrives senere. Der er dog en række steder i produktionen, hvor der er risiko for, at der opstår fejl. Disse steder er beskrevet af [Cockcroft, 1997], og kan opdeles i tre faser; indsamling af rådata, datagenerering og datalagring. Indsamlingen af rådata er de observationer, der foretages i den virkelige verden, f.eks. at tage et flyfoto. Under datagenereringen fortolkes rådata, og dermed omdannet til data i datasættet, og kunne eksempelvis være fotogrammetrisk opmåling ud fra flyfotos. Datalagringen omhandler, hvilken lagringsmetode, der benyttes. Disse tre faser beskrives nærmere i det følgende.

#### 10.1.1 Indsamling af rådata

Dataindsamling kan betegnes som den proces, hvor der foretages observationer i den virkelige verden. Den kan foretages på en række forskellige måder, såsom markmålinger med teodolit, totalstation eller GPS, eller forskellige luftbårne optagelser, som f.eks. flyfoto, radarmålinger fra fly eller satellit eller højdemålinger foretaget med laser. Disse metoder bruges i høj grad til at indsamle information om den geometriske del af geografiske data. Herudover kan der foretages en lang række målinger af attributværdier, der kan spænde over trafiktællinger og indsamling af geologiske prøver, til måling af forureningsniveauer eller hustyper.

Det er også muligt at producere data ud fra andre data, eksempelvis eksisterende papirkort eller andre databaser, men det er da vigtigt at sørge for, at disse data er så aktuelle, som det er nødvendigt. Hvilke datakilder, der skal benyttes, angives i specifikationen, og dette beskrives nærmere i kapitel 13.

Det er nødvendigt finde en metode, der kan levere data, der lever op til de opstillede krav. Det skal bestemmes hvilke instrumenter, der skal benyttes til indsamlingen, hvilke målemetoder, der skal anvendes, samt hvordan det indsamlede data skal analyseres.

I forbindelse med valg af instrument kan der opstå fejl, hvis instrumentet ikke kan måle nøjagtig nok til at indsamle de krævede data. Det kan f.eks. være at de anvendte flyfoto ikke har tilstrækkelig høj opløsning. Der er ofte tale om en afvejning af økonomi kontra præcision i forbindelse med dette valg, og det afgøres af producenten.

Når målingen foretages, er det vigtigt at instrumenterne er kalibreret og opstillet korrekt, at indsamlingen er præcis, samt at det er de rigtige ting der måles. Disse fejl kan formindskes ved hjælp af træning af de personer, der udfører målingerne. Hvis der er tale om uerfarne personer, kan der opstilles udførlige guider til hvordan målingen skal foregå, og herved afhjælpe dette problem.

Efter dataindsamlingen, foregår der en analyse af de indsamlede data, så det er muligt at generere de egentlige data ud fra dette. Her kan der opstå fejl i den indledende analyse af data, f.eks. analysen af feltbaserede data. Der kan være filtre eller automatiske metoder til analysen af data, hvor der kan opstå fejl.

Der er en flydende overgang mellem dataindsamlingen og datagenereringen, da der ved indsamlingen af data også genereres data, der eventuelt skal raffineres efterfølgende.

### 10.1.2 Datagenerering

Generering af data kan foregå forskellig vis. De forskellige metoder kan foranledige forskellige typer fejl. Ved at undersøge hvilke typer fejl der kan opstå ved de forskellige metoder, kan der dannes et billede af, hvilke fejl der kan forekomme i et geografisk datasæt. Der findes bl.a. følgende metoder til indsamling af data [Servigne et al., 1999]:

- *Digitalisering vha. fotogrammetrisk arbejdsstation eller lignende.* Fortolkningen er visuel og subjektiv, hvilket kan føre til fejl. Ved denne metode kan der forekomme problemer med præcisionen af indsamlede punkter, og der kan også opstå fejlfortolkninger. Desuden kan det forekomme, at punkter eller objekter digitaliseres flere gange, og der dermed forekommer dubletter.
- *Indtastning af data direkte via keyboard.* Her kan der forekomme forskellige typer fejl, når en person indtaster data.

Datakilden kan være aflæst forkert, og da der ikke altid er adgang til datakilden, og dens kvalitet, er det ikke muligt at kontrollere dette. Desuden er det let at foretage en simpel tastefejl, f.eks. at komme til at tilføje, bytte om på eller glemme et ciffer.

- *Automatisk indsamling.* Ved denne fremgangsmåde kan der forekomme problemer med computerens fortolkning af datakilden. Dette kan skyldes fejl i de algoritmer, der benyttes, at der er fejl i datakilden, eller at datakilden kan fortolkes på flere måder, som algoritmen ikke tager højde for.
- *Overførsel af data fra andre systemer.* Kvaliteten af det importerede data er ikke altid kendt, og derved kan der forekomme fejl i forbindelse med overførslen af data. Desuden kan der forekomme semantiske fejl på grund af forskellige systemer, eksempelvis forskellige datastrukturer, koordinatsystemer o.l. konverteringer.
- *Generering af data vha. eksisterende objekter i databasen.* Når en eksisterende database benyttes, kan der forekomme fejlphobning på baggrund af de benyttede objekter.

Udover disse metoder, kan der også opstå fejl på grund af fejlagtige fortolkninger af kildematerialet. Forskellige typer data er ikke altid afgrænset entydigt, og det kan give anledning til fejl. For det første kan det være at data ikke er vist repræsentativt i indsamlingsøjeblikket. For det andet kan det være svært at tolke, præcis hvor grænsen går, f. eks. mellem skov og mose, eller mellem klassifikationen på to forskellige marktyper. Det første problem kan løses ved at sørge for at få et repræsentativt billede af virkeligheden, eventuelt ved at foretage flere målinger. Det andet kræver træning af operatørerne. Desuden er det nødvendigt at klassificere objekterne korrekt, og sørge for at dette kan lade sig gøre.

Det ses, at der kan opstå fejl i dataindsamlingen. Nogle af fejlene kan findes ved blot at gennemse datasættet, f. eks. en polygon der ikke er lukket, eller et punkt der er uden for dataområdet, mens andre kræver, at der ses på topologi. Eksempelvis kan to overlappende polygoner både være eller ikke være en fejl, alt efter hvordan den virkelige verden skal tolkes. Desuden kan nogle fejl findes, ved at sammenligne med en referencedatakilde. Dette beskrives i kapitel 16.



### 10.1.3 Datalagring

Data kan lagres løbende under genereringen af data, enten direkte, så hvert enkelt objekt bliver lagret ligeså snart, det registreres, eller lagres i mindre dele, således at når der er registreret et vist antal objekter lokalt, overføres de til den samlede database. Den sidste metode gør, at det er muligt, at der opstår dubletter i data, hvis eksempelvis to operatører registrerer data samtidigt. Dette problem kan afhjælpes ved at gøre det muligt, at se allerede digitaliserede objekter løbende i processen. Lagringsmetoden har betydning for præcisionen, f.eks. kan der forekomme afrundingsfejl, hvis den værdi, der skal tilføjes til databasen, ikke kan lagres med tilstrækkeligt mange betydende cifre.

Der kan også opstå problemer med den rumlige nøjagtighed i feltbaserede systemer, da modeller med høj opløsning er meget pladskrævende, mens nøjagtigheden må ofres, hvis der ønskes en model, der ikke fylder så meget. Det er altså et spørgsmål om pladsbesparing eller detaljeringsgrad [Cockcroft, 1997].

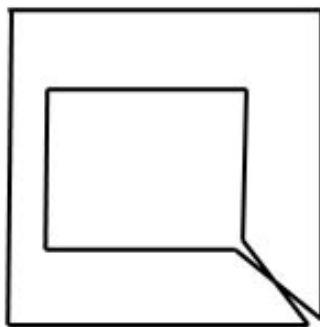
## 10.2 Fejltyper

Som nævnt består geografisk data af objekter, der har en række attributter, der både kan være rumlige og ikke-rumlige. Der er mulighed for at der er fejl, som er opstået af en af de ovenfor beskrevne grunde.

De fejl, der opstår i de rumlige attributter, kan deles op i tre grupper. Den første gruppe kaldes de strukturelle fejl, og opstår oftest på grund af valget af database og datastruktur. Den anden gruppe består af de fejl, hvor der ses på geometrien for ét objekt af gangen, og de kaldes her de geometriske fejl. Den tredje gruppe er dem, der kan opdages ved at se på sammenhængen mellem det enkelte objekt og det resterende datasæt, kaldet de topologiske fejl. Derudover kan der være egentlige strukturelle fejl, men de opstår oftest på grund af valget af database og datastruktur. En del fejl opstår, fordi operatøren overser eller mistolker en datakilde. De fejl, der beskrives herunder, er fejl der netop kan opdages pga. den karakter de har. De resterende fejl kræver sammenligning med referencedata, der har en bedre kvalitet. Dette begreb forklares nærmere i det efterfølgende kapitel. Hvis de to datasæt sammenlignes, er det muligt at finde de steder, hvor de ikke stemmer overens, og hvor der dermed må være fejl i et af dem.

### 10.2.1 Strukturelle fejl

De strukturelle fejl stammer fra datastrukturen. Datastrukturen skal lagre data, så det passer til datamodellen. Somme tider forekommer der elementer, der kan håndteres af datamodellen, men som ikke kan håndteres af datastrukturen. Det kan f.eks. være objekter, der består af flere polygoner. Som et eksempel kan der ses på en firlænget gård, hvor gårdspladsen ikke er en del af bygningen, og den derved ikke er en del af det bygningsobjekt, der skal repræsentere gården i datasættet. Dette kan f.eks. lagres som en polygon med en indre ring, som vist på figur 10-1, hvor polygonens afgrænsning folder ind i sig selv. En anden mulighed er, at bygningen repræsenteres af to polygoner, der har en retning tilegnet, så den ydre ring angiver, at den omkredser et objekt, og den indre ring angiver, at det, der ligger udenfor, er en del af objektet. Den første metode kan være uhensigtsmæssig. For det første fordi der er en risiko for, at der foretages fejlagtige digitaliseringer. For det andet kan det forringe fitness for use, da det ikke er en virkelighedstro repræsentation af en bygning. Den anden metode kan også give problemer i digitaliseringsprocessen, da det kan være svært at holde styr på, hvad der er ind og ud på polygonerne. Desuden er der en risiko for, at informationen om hvad der er ind og ud på polygonerne går tabt i eventuelle konverteringer af datasættet. De fleste GIS kan håndtere polygoner med indre ringe, men det er ikke muligt for brugeren at se, hvilken metode der benyttes.



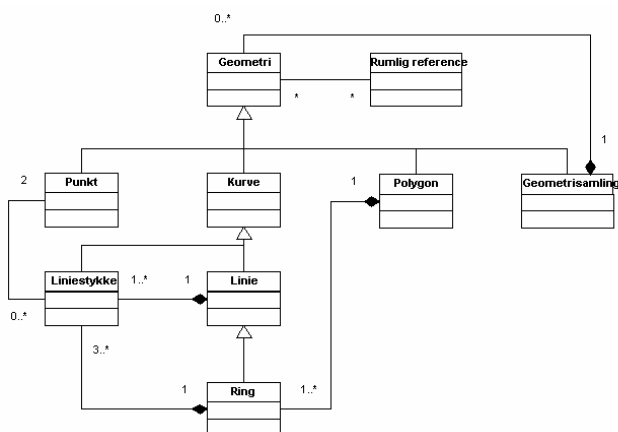
Figur 10-1. En dårligt digitaliseret polygon med hul.

Hvis en datastruktur ikke lagrer objekterne hensigtsmæssigt, kan det føre til ovenstående strukturelle fejl. Eftersom disse fejl stammer fra datastrukturen, er det fejl, der kan opstå på grund af nogle fejltagelser, der er foregået, da selve databasen blev opbygget. For at forhindre fejlene i at opstå, er det enten nødvendigt at vælge en datamodel, der

ikke giver mulighed for den slags fejl, eller sørge for, at operatøren er bekendt med de problemer der kan opstå, og derved undgår dem [Servigne et al., 1999].

### 10.2.2 Geometriske fejl

Data skal repræsentere den virkelige verden på en passende måde, der er defineret i specifikationen. I et GIS er det vigtigt at netop objektets geometri er så nøjagtig som mulig, og lever op til specifikationen. I et traditionelt 2d objektbaseret GIS findes der normalt de typer geometriske repræsentationer, der er vist på figur 10-2 og som også er beskrevet tidligere i kapitel 9. For at generalisere modellen beskrives den her vha. de tre objekttyper, punkt, linie og areal.



Figur 10-2 De gængse geometrityper (samme som figur 9-6)

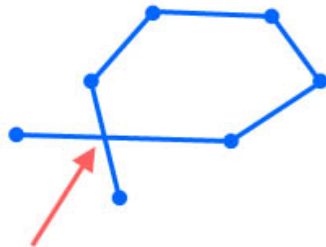
Geometriske fejl er fejl i et enkelt objekts geometriske repræsentation, dvs. det punkt, den linie eller den polygon, der repræsenterer objektets geometri. I det følgende beskrives en række fejl, der kan opstå for de forskellige geometrityper.

#### 10.2.2.1 Punkt

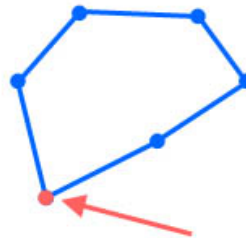
Et punkt har ingen udstrækning, og derfor vil de eneste geometriske fejl, der kan opstå, være tilfælde, hvor der er benyttet en forkert geometri. En trægruppe kan repræsenteres som et punkt. Hvis det i stedet registreres som en polygon, vil der være tale om en geometrisk fejl. Det er let at finde den slags fejl, og i eksempelvis ArcGIS er det ikke muligt at lave denne type fejl, da objekterne fra en klasse, kun kan have den samme type geometri.

### 10.2.2.2 Linie

Ligesom for et punkt, vil det være en fejl hvis der findes andre geometrityper end linie i en klasse, der repræsenteres af geometrien linie. De mulige geometriske fejl afhænger af hvordan objektclassen er defineret i specifikationen. Ofte vil det ikke være tilladt at have en linie, der krydser sig selv. I Top10dk er det ikke tilladt for to linier (eller den samme linie) at krydse hinanden, som illustreret figur 10-3, uden at der er et fællespunkt i det punkt, hvor de krydser [Kort & Matrikelstyrelsen, 2001]. En linie må typisk heller ikke være lukket, så den danner en ring, som vist på figur 10-4, specielt i forbindelse med netværk, da det herved ikke er muligt at foretage netværksanalyser [Pedersen, 2000].



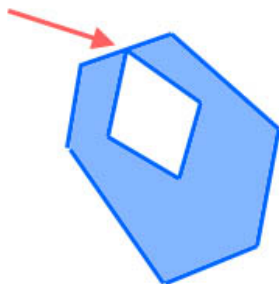
Figur 10-3 Linie, der er lukket



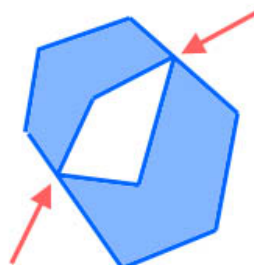
Figur 10-4 En linie der krydser sig selv

### 10.2.2.3 Polygon

En polygon er en lukket linie, der derved danner en ring. Ofte er det muligt at en polygon kan have indre ringe, der danner huller i polygonen. I forbindelse med disse indre ringe kan der også opstå geometriske fejl, som beskrevet tidligere. På figur 10-5 ses en polygon med en indre ring, der rører polygonens kant. Dette vil i mange tilfælde betragtes som en fejl, da det vil være mere korrekt, hvis kanten af polygonen i stedet gik ind i polygonen, og derved danner det, der før var defineret af den indre ring. På figur 10-6 ses en polygon, hvis indre ring rører to steder på den ydre ring. Det ville her være mere korrekt at dele polygonen op i to polygoner, der rører hinanden to steder.



Figur 10-5 En polygon, med en indre ring, der rører den ydre.



Figur 10-6 En polygon med en indre ring, der rører to steder på den ydre.

Ofte vil det også være en fejl, hvis polygonens kant skærer sig selv. Hvis det er tilfældet bør polygonen ligeledes deles op i to, så de to resulterende polygoner rører hinanden i det punkt, hvor den oprindelige polygons kant skar sig selv. Det er også en fejl, hvis polygonen ikke har en udstrækning i to dimensioner, således at den udgør en linie eller et punkt.

En anden type fejl for polygoner, men også for linier og punkter, er redundante punkter, dvs. punkter, der ligger så tæt på hinanden, at der ikke kan skelnes imellem dem, ved det målestoksforhold, data skal bruges ved. Disse punkter kan udgøre en fejl, f.eks. hvis det skal undersøges hvor mange hjørnepunkter der er i en polygon. Denne slags fejl opstår let i digitaliseringsprocessen, men de er til gengæld også lette at identificere og rette automatisk, ved hjælp af simple algoritmer [Christiansen & Olsen, 2002].

I visse tilfælde kan der også være huller i kanten af polygonen, alt efter hvordan denne defineres i databasen. I nogle tilfælde skal start- og endepunkt være identiske, for at en polygon er lukket, og hvis dette ikke gøres automatisk, risikeres det, at der opstår den slags huller. Det er også forholdsvis simpelt at finde den slags fejl, idet det undersøges om første og sidste punkt i alle polygoner er det samme.

De geometriske fejl opstår altså generelt i forbindelse med designet af datamodellen og under registreringen, og selvom de er forholdsvis lette at finde, er det vigtigt at være opmærksom på dem.

### 10.2.3 Topologiske fejl

Tidligere i kapitel 9 er de forskellige topologiske relationer, som to rumlige objekter kan have til hinanden, beskrevet. I specifikationen er

der regler for hvilke typer topologiske relationer, objekter kan have til hinanden. Disse afhænger af objektets geometri, men kan også afhænge af andre attributter, eksempelvis højdeattributter i 2d-GIS.

Specifikationen kan f.eks. definere, at en bygning skal ligge fuldstændig inde i en matrikel. Det vil sige, at bygninger ikke må skære kanten af matriklen, og heller ikke ligge udenfor. Hvis matrikelgrænserne ikke er defineret korrekt, eller hvis operatøren har glemt at dele en bygning op, der går hen over grænsen af en matrikel, vil der opstå en fejl. En anden type fejl kan være, hvis operatøren digitaliserer et vejnet, der skal hænge sammen alle de steder, hvor veje mødes. Her kan der opstå fejl, hvis operatøren ikke sørger for, at det er præcis det samme punkt, vejene ender i. Et andet eksempel kan være, at stoppesteder for busser skal ligge på en vej. Hvis der ikke sørges for at stoppestedet har fællespunkt med en vej, vil der opstå fejl.

Topologi angiver de rumlige relationer, objekter har til andre objekter i datasættet. Det er altså muligt vha. topologi at angive, hvordan de forskellige objekter i datasættet skal relatere indbyrdes. Det er en fejl, hvis reglerne for disse topologiske relationer ikke overholdes, og det ville derfor være en fordel, hvis disse fejl automatisk registreres, så det er muligt at rette dem. Dette beskrives i et senere kapitel 15.

Det er svært, måske endda umuligt, at generere et fejlfrit datasæt bestående af geografisk data. Det er derfor bl.a. vigtigt at dokumentere hvor mange fejl, der findes i det pågældende datasæt. Dette kan gøres vha. en række kvalitetselementer, der beskrives i det følgende kapitel.

## 11 Dokumentation af kvalitet

Værdien af enhver database ligger i kvaliteten og brugbarheden af det indeholdte data. Det ses specielt, når mange former for information anvendes sammen [Worboys, 1995]. Når dette er gældende, er det f.eks. ønskeligt at kunne følge den fejløphobning, der sker under en analyse [Guptill & Morrison, 1995] [Christensen & Friis-Christensen, 2003]. Ofte kan dette være besværligt eller helt umuligt af to grunde. For det første, at det f.eks. er vanskeligt at sammenligne to vidt forskellige ting. Nogle gange er det dog nødvendigt at foretage denne vurdering, og der eksisterer metoder til at gøre det. For det andet, at fitness for use i dag hovedsageligt vurderes ud fra en overordnet beskrivelse af data, ved brug af metadata [Christensen & Friis-Christensen, 2003]. Dette er med til at begrænse anvendelsen, da informationen om kvaliteten ikke ligger på direkte dataniveau, men i stedet findes selvstændig i en produktbeskrivelse eller en kvalitetsrapport.

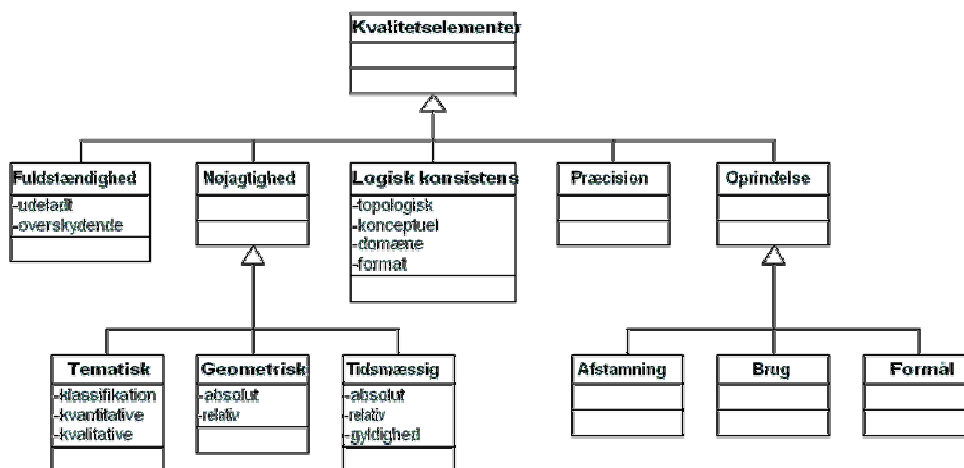
For at vurdere kvaliteten af geografiske data, er det derfor nødvendigt at opstille nogle fundamentale måder at beskrive kvaliteten på. En måde at gøre dette er ved at angive en række kvalitetselementer, der gør det muligt at vurdere og sammenligne kvaliteten af forskellige kilder. Ud fra disse kvalitetselementer samt andre relevante informationer om data kan fitness for use vurderes. Kvalitetselementerne kan både være kvantitative, men også kvalitative, og de har derfor forskellige anvendelsesmuligheder. Derudover er det også forskellige steder i produktionsprocessen, de kan registreres.

I det følgende beskrives de enkelte kvalitetselementer, der har betydning for vurderingen af kvaliteten af geografiske data. Beskrivelsen tager udgangspunkt i den internationale standardorganisation ISO, der angiver det grundlæggende inden for datakvalitet.

### 11.1 Kvalitetselementer

Der er forskellige måder at beskrive kvaliteten af geografiske data på, og de forskellige forfattere inden for emnet har hver deres syn på hvilke elementer, der skal til for at muliggøre dette. Størstedelen af elementerne er de i hovedtræk enige om, men der er enkelte elementer, der har forskellige betegnelser. Derudover har enkelte forfattere yderligere elementer eller anden opdeling af disse. De

enkelte elementers opdeling og betegnelserne, der benyttes i denne rapport, er illustreret på figur 11-1 og beskrives nærmere i det følgende.



Figur 11-1 Kvalitetslementerne

Kvalitetslementerne kan som sagt både være kvantitative og kvalitative. Herunder beskrives først de kvantitative kvalitetslementer.

### 11.1.1 Kvantitative elementer

De kvantitative kvalitetslementer kan, som betegnelsen siger, kvantificeres ved en række talværdier. Ud fra de kvantitative elementer er det muligt direkte at sammenligne kvaliteten af data med de krav der stilles til data i specifikationen. I det følgende beskrives disse kvantitative kvalitetslementer samt de tilhørende underelementer.

#### 11.1.1.1 Fuldstændighed

Traditionelt set er fuldstændighed (completeness) defineret som den del af datakvalitet, der beskriver forholdet mellem antallet af objekter fra en objektklassen, der er repræsenteret i data og antallet af objekter fra tilsvarende klasse, i det perfekte datasæt, hvor alle objekter indgår [Guptill & Morrison, 1995].

Der kan også være tale om fuldstændighed inden for attributter [Guptill & Morrison, 1995]. Her er graden af fuldstændighed beskrevet, ud fra hvor mange attributter der er med, i forhold til det der er angivet i specifikationen. Det vil sige, at fuldstændighed



handler både om, hvorvidt alle objekter er til stede, men også om, om alle attributter eller egenskaber er til stede [Christensen, 1997].

Kvalitetselementet fuldstændighed kan opdeles i to elementer [ISO, 2001] [Friis-Christensen et al., 2001] [Christensen & Friis-Christensen, 2003] [Christensen, 1997] [Guptill & Morrison, 1995]; *commission* og *omission*, der kan oversættes som hhv. *overskydende* og *udeladt*. At noget er *overskydende* betyder at noget er med, men ikke burde være det, og kan kvantificeres ved det antal, der er for meget, i forhold til det samlede antal, der burde være med. *Udeladt* betyder derimod at noget burde være med, men ikke er det, og kan kvantificeres ved det antal, der er med i forhold til det samlede antal, der burde være med.

Fuldstændighed er knyttet til hele datasættet eller til den enkelte objektklasse, og måles i forhold til en reference. Kontrollen foregår, når alle data er til stede, og der ses på de enkelte objekter fra det samlede datasæt. Det er ikke muligt at kontrollere underelementet udeladt (*omission*) undervejs i genereringen af data, da grunden til, at et objekt ikke er med, kan være, at operatøren ikke er nået så langt endnu. Derimod kan underelementet *overskydende* (*commission*) løbende kontrolleres, men det kræver, ligesom underelementet udeladt, et referencegrundlag og det er derfor ikke altid muligt. Dette beskrives senere i dette kapitel.

#### 11.1.1.2 Nøjagtighed

Nøjagtighed (*accuracy*) kan defineres som forskellen mellem et objekt eller en attribut og en sammenlignelig observation, som har en kendt højere nøjagtighed [Guptill & Morrison, 1995]. Nøjagtigheden kan inddeles i tre underelementer og beskrives herunder.

##### 11.1.1.2.1 Tematisk nøjagtighed

Tematisk nøjagtighed (*thematic accuracy*) [Friis-Christensen et al., 2001] [ISO, 2001] [Christensen & Friis-Christensen, 2003] eller attributnøjagtighed [Guptill & Morrison, 1995] [Christensen, 1997] er det element, der beskriver hvor godt et objekt og dets egenskaber er klassificeret. Elementet kan beskrives ud fra tre dele. Den første del er *rigtigheden af klassifikationen* (*Classification correctness*) [ISO, 2001] [Christensen & Friis-Christensen, 2003] [Christensen, 1997], der angiver hvor godt data er i overensstemmelse med et referencedatasæt, der har en kendt højere nøjagtighed. Det er f.eks. om en bygning er klassificeret som en bygning og ikke en sø. Den anden del angiver *rigtigheden af de kvalitative egenskaber* [ISO, 2001] [Friis-Christensen et al., 2001] [Christensen & Friis-Christensen,

2003] og beskriver hvor stor en andel af de kvalitative egenskaber, der er rigtige. De kvalitative egenskaber er f.eks. navnet på en vej eller kommune. Den sidste del angiver *rigtigheden af de kvantitative egenskaber* [ISO, 2001] [Friis-Christensen et al., 2001] [Christensen & Friis-Christensen, 2003] [Christensen, 1997], der beskriver, hvor godt de kvantitative egenskaber er bestemt, f.eks. om antallet af etager i en bygning eller en bygnings højde er rigtig. Disse tre elementer kan kvantificeres ved antallet af fejl i forhold til det samlede antal af den målte egenskab.

Som det ses, knytter disse elementer sig til objekters attributter. ligesom fuldstændighed, kan den tematiske nøjagtighed løbende kontrolleres, da det også her er et spørgsmål om at kontrollere de data, der er registreret i forhold til andre data med en højere nøjagtighed. Da sådan en reference ikke altid er til stede under registreringen af data, skal det samlede datasæt kontrolleres efter produktionen, ved her at sammenligne det med andre kendte data, og herved sikre at producenten overholder de krav der stilles i specifikationen. Efter denne kontrol bestemmes og registreres elementerne, hvilket beskrives i kapitel 16.

#### 11.1.1.2.2 Geometrisk nøjagtighed

Den geometriske nøjagtighed (spatial accuracy) [Friis-Christensen et al., 2001] [Christensen & Friis-Christensen, 2003] eller den positionelle nøjagtighed (positional accuracy) [ISO, 2001] [Guptill & Morrison, 1995] [Christensen, 1997] er det mest brugte element til at angive kvaliteten af geografiske data [Christensen, 1997]. Den geometriske nøjagtighed kan deles op i en absolut nøjagtighed og en relativ nøjagtighed. Den *absolutte geometriske nøjagtighed* beskriver positionen af et objekt i forhold til en reference med en kendt nøjagtighed, f.eks. om en bygning ligger det rigtige sted i forhold til virkeligheden. Den *relative geometriske nøjagtighed* beskriver nøjagtigheden af objekters position i forhold til hinanden [Friis-Christensen et al., 2001] [Christensen & Friis-Christensen, 2003] [ISO, 2001] [Guptill & Morrison, 1995] f.eks. om to bygninger har den rigtige afstand imellem hinanden.

Den absolutte og relative nøjagtighed hænger sammen på den måde at, hvis to objekter har en høj absolut nøjagtighed, må deres indbyrdes relative nøjagtighed nødvendigvis også være høj. Det omvendte gælder ikke nødvendigvis. To bygninger kan godt ligge rigtigt i forhold til hinanden, men begge være forskudt i den samme retning. Herved er den relative nøjagtighed mellem de to bygninger høj, mens den absolutte nøjagtighed er lav for dem begge.

Geometrisk nøjagtighed knytter sig til objekternes rumlige egenskaber [Friis-Christensen et al., 2001] og kvantificeres ofte vha. spredningen.

#### 11.1.1.2.3 Tidsmæssig nøjagtighed

En vigtig faktor for at vurdere kvaliteten af data, er at have en viden om, hvilket tidspunkt de stammer fra, og hvor god denne tidsangivelse er. Den tidsmæssige nøjagtighed (temporal accuracy) hænger i høj grad sammen med de resterende kvalitetselementer, og det er derfor også vigtigt at tage tiden med ind i overvejelserne når geografiske data benyttes. Det er f.eks. ikke muligt, at tale om fuldstændighed, uden også at tænke på hvornår data er produceret. Det kan være, at fuldstændigheden er høj på et givet tidspunkt, men hvis data ikke opdateres løbende, kan der være sket mange ændringer og tilføjelser siden. Hvis det er tilfældet, skal der tages højde for dette under vurderingen af data.

Der foregår hele tiden ændringer i den virkelige verden, og forskellige typer objekter ændrer sig med forskellig hastighed. Ajourføringen af data skal derfor tilpasses så det afspejler den virkelige verden [Guptill & Morrison, 1995].

Tidsmæssig nøjagtighed beskriver kvaliteten af de aspekter i data, der har med tiden at gøre, og kan både være relativt og absolut [Christensen & Friis-Christensen, 2003]. Den *absolutte tidsmæssige nøjagtighed* angiver nøjagtigheden af de tidsmålinger, der indgår i data (measurement) [ISO, 2001] [Friis-Christensen et al., 2001] [Guptill & Morrison, 1995]. Den *relative tidsmæssige nøjagtighed* omhandler den tidsmæssige konsistens (consistency) af data og indikerer om begivenhederne kommer i den rigtige rækkefølge relativt [ISO, 2001] [Friis-Christensen et al., 2001] [Christensen & Friis-Christensen, 2003] [Guptill & Morrison, 1995]. Derudover angiver *den tidsmæssige gyldighed* (validity) [ISO, 2001] [Friis-Christensen et al., 2001] om de tidsangivelser, der findes i data, også eksisterer. Elementerne knytter sig altså alle til objekters tidsmæssige egenskaber [Friis-Christensen et al., 2001].

#### 11.1.1.3 Logisk konsistens

Logisk konsistens (logical consistency) beskriver, i hvilken grad data passer til den struktur, som er angivet i specifikationen for data [Christensen & Friis-Christensen, 2003]. Logisk konsistens er hovedsagelig et spørgsmål om *topologisk sammenhæng* i data [ISO, 2001] [Friis-Christensen et al., 2001] [Christensen & Friis-Christensen, 2003] [Christensen, 1997] [Guptill & Morrison, 1995], og beskriver hvor godt de topologiske regler, der er opstillet i

specifikationen, er overholdt. En topologisk regel beskriver hvordan objekter og deres rumlige attributter forholder sig til hinanden. Dette kan f.eks. være, at veje skal hænge sammen i et netværk eller at to bygninger ikke overlapper hinanden. Dette er beskrevet tidligere i kapitel 9.

Der findes også andre elementer inden for logisk konsistens. Et af disse elementer er *begrebsmæssig sammenhæng* (conceptual consistency) [ISO, 2001] [Christensen & Friis-Christensen, 2003], der angiver hvor godt objekterne er klassificeret i forhold til de klasser, der er angivet i specifikationen. Dette kan f.eks. være om klassen "Bygning" eksisterer i data og er navngivet som beskrevet i specifikationen. Det skal ikke forveksles med om en bygning er klassificeret rigtigt, hvilket hører til tematisk nøjagtighed.

Et andet element er *domænesammenhæng* (domain consistency) [Friis-Christensen et al., 2001] [ISO, 2001] [Christensen & Friis-Christensen, 2003], der angiver, om data overholder de værdier, der er indeholdt i værdidomænet. Et eksempel på dette er at terrænhøjder i Danmark burde ligge mellem -100 og 300 meter over havets overflade, og der derfor ikke findes højder, der er 4000 meter over.

Derudover skal der være en *formatsammenhæng* (format consistency) [ISO, 2001] [Friis-Christensen et al., 2001] [Christensen & Friis-Christensen, 2003], der beskriver den del af data, der overholder de krav der er stillet til dataformatet. Det er f.eks. at antallet af etager i en bygning skal angives med et heltal og ikke et decimaltal.

Elementerne skal alle bestemmes efterfølgende for hele datasættet, men det er muligt at kontrollere dem undervejs i produktionen og derved undgå at levere et produkt, der ikke lever op til de stillede krav. Disse regler kan opstilles matematisk [Guptill & Morrison, 1995], og det er derfor muligt at foretage kontrollen automatisk, hvilket beskrives nærmere i kapitel 15. Hvis kontrollen foretages automatisk når et objekt er registreret, vil det ikke være nødvendigt med en efterfølgende kontrol af det samlede datasæt, da der herved ikke vil være nogen egentlige fejl tilbage, hvis reglerne og automatiseringen er opstillet korrekt. Der er dog undtagelser i disse regler, og her må producenten foretage en vurdering. Dette kan f.eks. være, at objekter fra klassen "Bygning" ikke må overlape et objekt fra klassen "Sø". Hvis der rent faktisk ligger en bygning ude i en sø, eksempelvis på pæle, kan dette betragtes som en undtagelse.

#### 11.1.1.4 Præcision

Præcisionen (precision) er den størrelse, som det anvendte instrument kan måle med og forholder sig derved mere til målemetoden, frem for de enkelte målepunkter [Christensen & Friis-Christensen, 2003]. En anden måde at beskrive dette element på, er ud fra [Guptill & Morrison, 1995], der skriver at præcision kan defineres som den detaljeringsgrad, der er brugt til at foretage en observation. Dette kan eksempelvis illustreres ved, at 4,321 er mere præcis end 4,3 men det er ikke nødvendigvis mere nøjagtig, hvis den eksakte værdi er 4,3.

Da dette element ikke bestemmes i forhold til en form for reference, skal størrelsen findes under produktionsprocessen, og ikke bestemmes efterfølgende som f.eks. geometrisk nøjagtighed. Præcisionen er med til at kvantificere det kvalitative kvalitetselement oprindelse (origin) [Christensen & Friis-Christensen, 2003], der beskrives i det følgende afsnit, og indeholder bl.a. information om, hvilken målemetode der er benyttet. Ved at benytte dette kvantitative kvalitetselement, er det lettere at sammenligne forskellige dataindsamlingsmetoder, uden at have et dybere kendskab til, hvad der kan forventes af disse.

#### 11.1.2 Kvalitative elementer

De kvalitative kvalitetselementer kan betegnes som yderligere information i form af metadata og knytter sig til forskellige ting afhængig af elementet. Udtryk og begreber for de kvalitative kvalitetselementer er ikke helt enslydende i de forskellige kilder. Der er fire begreber, der ofte benyttes til mange af de samme ting, og de er; origin, lineage, usage og purpose. Specielt origin og lineage er der ikke helt enighed om. I [Friis-Christensen et al., 2001] benyttes der kun tre af de nævnte begreber; origin, lineage og purpose, men de beskrives ikke nærmere, der henvises blot til [Guptill & Morrison, 1995]. [ISO, 2001] definerer også tre begreber, men her er det lineage, usage og purpose, mens der i [Christensen, 1997] benyttes to af disse: lineage og usage. Dette skyldes dog, at [Christensen, 1997]'s begreber er baseret på en ældre udgave af ISO-standarden, CEN fra 1996, hvor der nu er tilføjet et yderligere element; purpose. Derimod nævnes der kun én i [Christensen & Friis-Christensen, 2003] nemlig origin, som indeholder alle informationerne fra de andre kvalitative kvalitetselementer. Det samme er gældende i [Guptill & Morrison, 1995], der bare kalder det for lineage. Hos [Guptill & Morrison, 1995] er origin et underelement sammen med en række andre mere specifikt beskrivende elementer.

For at samle de forskellige måder at beskrive de enkelte kvalitative elementer på, er det her valgt at tage udgangspunkt i [Christensen & Friis-Christensen, 2003], der bruger origin som det overordnede kvalitetselement og tilføje lineage, usage og purpose fra [ISO, 2001] som underelementer. Ud fra de forskellige kilder beskrives denne fremgangsmåde i det følgende.

#### 11.1.2.1 Oprindelse

Oprindelsen (origin) er de informationer, der siger noget om, hvor data kommer fra. Ved at opdele dette element i tre underelementer, afstamning (lineage), brug (usage) og formål (purpose) sikres det, at alle vigtige oplysninger er med, når kvaliteten skal vurderes. Disse kvalitative kvalitetselementer skal angives sammen med et objekt, attribut eller relation [Christensen & Friis-Christensen, 2003], når et objekt registreres, og gøres derfor under produktionen af data, med undtagelse af elementet brug, men det beskrives nærmere i afsnittet herom.

De kvalitative kvalitetselementer bygges op i en hierarkisk struktur, så elementet har forbindelse til alle de dele, et objekt består af [Christensen & Friis-Christensen, 2003]. Det skal eventuelt være muligt at få denne information helt ned på punktniveau og ikke bare til hvert objekt, men dette beskrives i et senere kapitel.

##### 11.1.2.1.1 Afstamning

*Afstamningen* (lineage) [ISO, 2001] [Friis-Christensen et al., 2001] [Guptill & Morrison, 1995] [Christensen, 1997] kan beskrives som historien eller livscyklus for data, f.eks. hvor data stammer fra, hvilket firma / hvilken person, der har registreret data, metode for registrering samt hvilken datakilde der er benyttet. Vurderingen af alle andre kvalitetselementer er påvirket af denne afstamning.

Formålet med at angive hvor data stammer fra kan ses fra to sider, fra producentens side og fra brugerens side. Fra producentens side kan afstamningen bruges til at sikre sine data, ved at holde styr på hvor og hvordan de er fremkommet. Brugeren ser afstamningen som stamtavlen for data, men for brugeren er det ikke absolut nødvendigt at vide alle ting omkring afstamningen af data. Det er måske ikke nødvendigt at kende operatørens navn hvis brugeren ikke kender til hans/hendes evner [Guptill & Morrison, 1995]. Brugeren kan først drage nytte af oplysningen, hvis han finder ud af, at denne operatør har lavet en del fejl i data. Data fra den pågældende operatør kan så udvælges og kontrolleres. Afstamningen skal derfor knyttes til samtlige dele i datasættet.

Det overordnede mål med informationen om hvor data stammer fra, er at bevare historieoplysninger for fremtiden [Guptill & Morrison, 1995].

#### 11.1.2.1.2 Brug

*Brugen* (usage) [ISO, 2001] [Friis-Christensen et al., 2001] [Christensen, 1997] af data beskriver hvem der tidligere har anvendt data, og hvad de har været brug til. Derudover kan det angives, hvilke erfaringer der er med data.

Dette element knytter sig til hele datasættet, men det er ikke noget producenten kan tilføje et nyt produkt. Det er derimod noget, der skal tilføjes efterhånden som produktet anvendes, hovedsageligt med det formål at vise hvad produktet kan bruges til overfor potentielle brugere. Det vil være med til at øge tilliden til et produkt hvis det er muligt at se, at andre har haft nytte af det.

#### 11.1.2.1.3 Formål

Et element, der også kan være med til at forbedre grundlaget for valg af data, er *formålet med data* (purpose) [ISO, 2001]. Formålet med data angives for at forklare, hvad data oprindeligt var tiltænkt, og hvorfor det blev produceret.

Formålet knytter sig også til hele datasættet og kan angives af producenten.

## 11.2 Bestemmelse af kvalitetselementer

Kvalitetselementerne bestemmes to forskellige steder i produktionen af geografisk data; under en intern kontrol og en ekstern kontrol [Friis-Christensen et al., 2001] [Christensen & Friis-Christensen, 2003].

*Den Interne kvalitetskontrol* checker om data overholder regler for konsistens, f.eks. topologi eller om en attribut overholder et givet domæne, hvilket beskrives senere kapitel 15. Den kontrol, der foretages under den interne kontrol, foregår på baggrund af de andre objekter i samme datasæt, og kræver derfor ikke kendskab til noget andet referencegrundlag. Derudover skal de kvalitative elementer ligeledes angives under den interne kontrol.

*Den eksterne kvalitetskontrol* checker derimod data i forhold til en form for reference for at se om et objekt også eksisterer i et kendt datasæt. Elementer der checkes under ekstern kontrol er fuldstændighed og nøjagtighed. Nøjagtigheden og fuldstændigheden kan, i nogle tilfælde, kontrolleres under produktionen, men da det

kræver et referencegrundlag, er det relevant at foretage denne kontrol når hele datasættet er færdiggenereret.

Nedenstående tabel 11-1 viser en samlet oversigt over hvor de forskellige kvalitetselementer kontrolleres og angives.

Kvalitetselementer		Intern	Ekstern
Kvantitative elementer	<b>Fuldstændighed</b>	-	<b>K</b>
	<b>Tematisk nøjagtighed</b>	-	<b>K</b>
	<b>Geometrisk nøjagtighed</b>	-	<b>K</b>
	<b>Tidsmæssig nøjagtighed</b>	-	<b>K</b>
	<b>Logisk konsistens</b>	<b>K</b>	-
	<b>Præcision</b>	<b>A</b>	-
Kvalitative elementer	<b>Afstamning</b>	<b>A</b>	-
	<b>Brug</b>	-	-
	<b>Formål</b>	<b>A</b>	-

Tabel 11-1 Angivelse af hvor kontrollen af kvalitetselementerne foregår. K betyder at den egentlige kvalitets kontrol foretages og kvalitetselementet angives her. A betyder at kvalitetselementet skal angives under denne kontrol.

Et vigtigt aspekt, der kan gøre vurderingen af kvaliteten af geografiske data bedre, er en standardisering af beskrivelsen af datakvalitet [Guptill & Morrison, 1995]. De elementer, der her er præsenteret, hænger i høj grad sammen med den måde, ISO foreslår datakvalitet skal beskrives på, og det er vigtigt at forsøge at anvende den samme fremgangsmåde, så data fra forskellige kilder kan vurderes på samme grundlag. ISO lægger samtidig op til, at det er muligt at tilføje yderligere elementer, der kan være med til at præcisere netop de pågældende data.



# Del 3 - Analyse

På baggrund af den foregående teori, indeholder den følgende analyse en beskrivelse af en metode til at producere geografiske data. Denne produktionsproces består bl.a. af at specificere en række regler i en specifikation, samt af en kontrol, der er med til at sørge for, at data overholder de regler, der er opstillet i denne specifikation. Disse regler er bl.a. af topologisk og geometrisk karakter, som beskrevet i teoridelen.

I analysen gives desuden et eksempel på, hvordan nogle af disse regler kan oprettes, vha. en applikation. Denne applikation er ikke implementeret, men er kun et tænkt eksempel, hvor brugergrænsefladen og funktionaliteten beskrives og illustreres.

I denne del af rapporten henvises der til de appendikser, der findes bagerst i rapporten, og som bl.a. uddyber nogle af de konkrete fremgangsmåder i forbindelse med ArcGIS, såsom etablering af en Geodatabase samt definerings af topologiske regler.



## 12 Produktionsmodel

Det overordnede formål med dette projekt er at forbedre produktionen og dokumentere kvaliteten af data. For at gøre dette, opstilles der en produktionsmodel for geografisk data i det følgende, hvor de tiltag, der kan afhjælpe de beskrevne problemer, uddybes.

Det er muligt at opstille en række forskellige syn på hvad kvaliteten af data er. De kan f.eks. inddeles i følgende fem syn på kvalitet [Christensen & Friis-Christensen, 2003] [Christensen, 1997]:

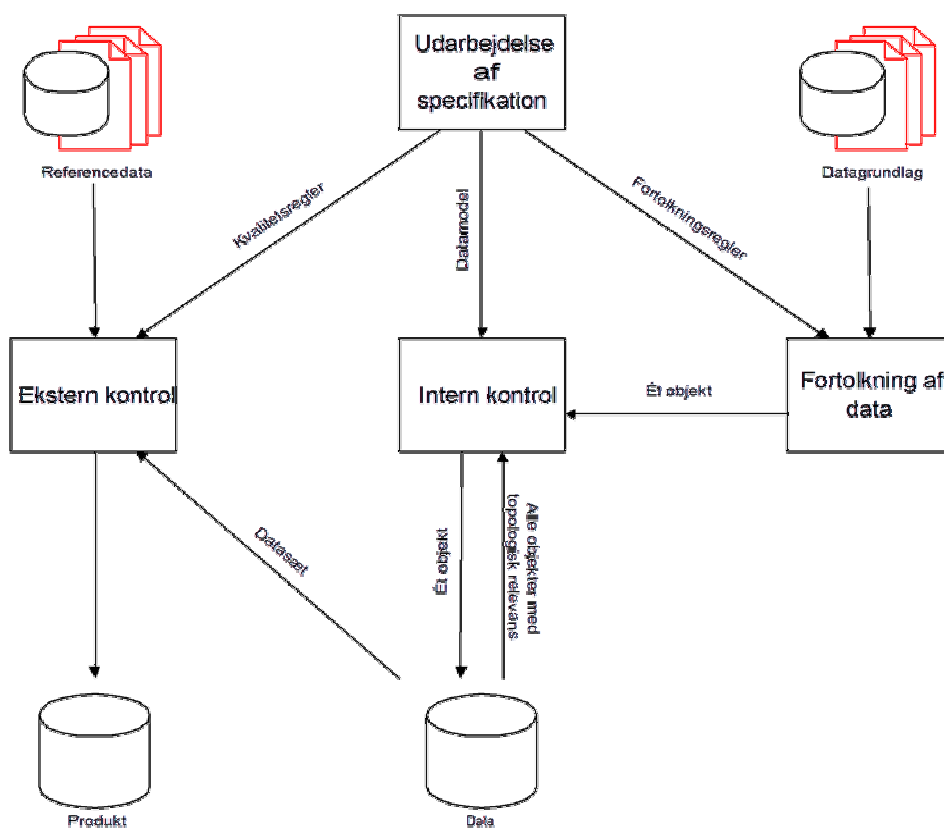
- 1) *Kvalitet i forhold til det perfekte.* Kvalitet er ensbetydende med det bedst mulige, hvor alt er perfekt og kvaliteten måles i forhold til dette.
- 2) *Kvalitet i forhold til producenten.* Producentens syn på kvalitet, hvor produktet skal leve op til en givet specifikation, og der ikke er nogen fejl.
- 3) *Kvalitet i forhold til brugeren.* Brugers syn, hvor det ønskes at produktet skal passe til det, det skal bruges til (fitness for use).
- 4) *Kvalitet i forhold til produktet.* Det produktbaserede syn, hvor kvaliteten er et sæt målbare karakteristika, der skal overholde en vis værdi for at tilfredsstille brugeren.
- 5) *Kvalitet i forhold til økonomien.* Det økonomisk baserede syn, hvor kvaliteten skal måles i forhold til hvad det koster at opnå denne kvalitet.

Disse forskellige syn på hvordan kvalitet kan opfattes, er alle nogle syn, der tager udgangspunkt i et færdigt produkt, og ser det ud fra hvordan dette produkt lever op til de forventninger, der stilles til produktet. Det er f.eks. om produktet lever op til specifikationen, til brugernes behov og om produktet er mere værd end det har kostet.

Alternativt kan der ses på de samme ting, med udgangspunkt i produktionen af produktet. Producenten vil gerne producere et produkt, der lever op til specifikationen, men det er specifikationen, der afgør, om produktet også lever op til kravene fra de resterende syn. Producenten specificerer de krav der stilles i specifikationen, og

dette gøres ud fra overvejelser om hvilke behov der er fra brugerne samt økonomiske begrænsninger. I denne analyse tages der udgangspunkt i denne tilgangsvinkel til kvalitet af et produkt bestående af geografisk data.

Produktionsmodellen for geografiske data er illustreret på figur 12-1, fra udarbejdelse af produktionspecifikation til det endelige produkt, med generering og kontrol af data undervejs. Denne produktion kan deles ind i fire processer. Producenten starter med at udarbejde en specifikation, der bl.a. indeholder de regler, som produktet skal overholde. Disse regler beskrives nærmere i det kapitel 13, om specifikationen, hvor det beskrives, hvad specifikationen skal indeholde.



Figur 12-1 Produktionsmodel for produktionen af geografiske data.

Efter at reglerne er defineret i specifikationen, skal data produceres i henhold til denne. Dette gøres ved at fortolke en eller flere datakilder. Disse datakilder kan, som det er i de fleste tilfælde når der er tale om geografiske data, være flyfotos, men det kan lige så godt være andre

former for data, f.eks. papirkort eller eksisterende databaser. I fortolkningsreglerne i specifikationen er det defineret, hvilke typer datakilder, der skal benyttes som grundlag for det pågældende produkt. Ud fra disse fortolkningsregler registreres data, ét objekt ad gangen. Denne proces uddybes i kapitel 14.

Når et objekt er registreret efter de instruktioner, der er beskrevet i specifikationen, skal objektet kontrolleres for bl.a. topologiske fejl, hvilket sker i den interne kontrol. Når et objekt er kontrolleret, og der ikke er opstået fejl, sendes det videre til en database, hvor det indgår i kontrollen af det næste objekt. Denne del af produktionsmodellen beskrives nærmere i kapitel 15 om den interne kontrol, hvor det ligeledes beskrives hvad der skal ske hvis en fejl opdages.

Når registreringen er færdig, sendes hele det pågældende datasæt videre til den eksterne kontrol, hvor det sammenlignes med et referencegrundlag. Herved kontrolleres det, om datasættet overholder de krav, der er stillet i kvalitetsreglerne i specifikationen. Referencedata kan være af forskellig karakter, og beskrives nærmere under kapitel 16 om den eksterne kontrol.

## 13 Specifikation

En specifikation er et dokument, der indeholder en detaljeret teknisk beskrivelse af et produkt, samt hvilke krav produktet skal opfylde. Specifikationens oplysninger kan bruges til en lang række ting, f.eks. under produktionen af produktet, men også når produktet efterfølgende skal anvendes. I disse tilfælde er det producenten selv, der skriver specifikationen, så de ved hvordan produktet skal produceres. Derudover kan producenten vha. specifikationen, også beskrive, hvad produktet indeholder overfor mulige brugere. Det kan også være at det er brugerne, der skriver specifikationen, for at angive hvilke behov de har [ISO, 2003], men i denne analyse tages der udgangspunkt i at det er producenten, der skriver specifikationen.

Specifikationer kan opdeles i to typer:

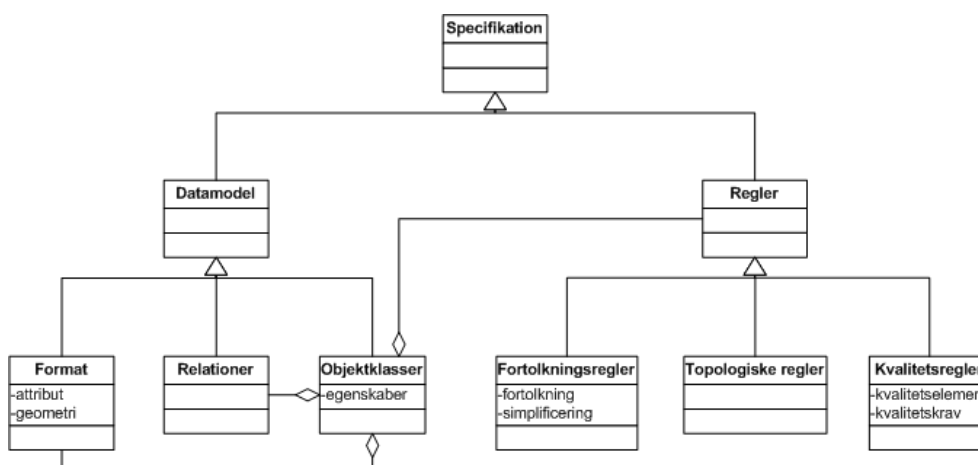
- Produktionsspecifikation
- Produktspecifikation

Produktionsspecifikation angiver, hvordan et produkt skal fremstilles, og hvad det skal indeholde, mens produktspecifikation beskriver, hvordan produktet er blevet til, og hvad det indeholder. Disse to typer specifikationer er på mange måder ens, for en produktionsspecifikation kan ud over at benyttes i produktionen, også bruges til at beskrive produktet med som produktspecifikationen. Omvendt kan produktspecifikationen ikke nødvendigvis bruges til at producere produktet efter. Produktspecifikationen kan benyttes som metadata, der beskriver hvad der indgår i data, samt hvilke krav, der er stillet. Produktionsspecifikationen kan anvendes til at generere produktspecifikationen for det datasæt, der er produceret i overensstemmelse med den pågældende specifikation [ISO, 2003]. Denne analyse omhandler produktionsspecifikationen.

### 13.1 Opbygning

En specifikation består af to overordnede dele; en beskrivelse af det datamæssige indhold og datastruktur samt en række regler, som indholdet skal overholde [Guptill & Morrison, 1995] [Christensen & Kamper, 2002]. Denne opbygning er illustreret på figur 13-1. Beskrivelsen af indholdet angiver hvordan datamodellen skal se ud, herunder hvilke objekter data skal indeholde, hvordan disse objekter skal opbygges samt hvilke relationer de skal have. Reglerne beskriver

bl.a. de krav, der stilles til de forskellige objekter og deres indbyrdes topologiske forhold, f.eks. at en bygning ikke må ligge ude i en sø, men også hvordan de skal registreres. Disse ting beskrives nærmere i de følgende afsnit.



Figur 13-1 Opbygning af specifikation

### 13.1.1 Datamodel

I datamodellen i specifikationen sættes de overordnede rammer for hvilke objektklasser datasættet skal bestå af, samt hvilke attributter og egenskaber objekterne skal indeholde. Ud over indholdet består datamodellen også af regler, hvilket er illustreret ved aggregeringen mellem objektklasserne og reglerne, og beskrives nærmere i det følgende.

#### 13.1.1.1 Objektklasser

Det vigtigste ved datamodellen er, at den angiver hvilke objektklasser datasættet skal indeholde. Objektklasserne består af en række egenskaber. I datamodellen angives det derfor også hvilke attributter, hvert objekt i en klasse skal bestå af. Det kan f.eks. være at hvert objekt skal have en objektkode, en bestemt geometri type eller forskellige kvalitetselementer, der skal angives for hvert objekt, osv.

Et aspekt, der er vigtigt ved en specifikation, er, at den kan ændre sig med tiden, og det er derfor vigtigt at vide hvilken specifikation, der er benyttet til registrering af de enkelte dele i datasættet [Christensen & Kamper, 2002]. Derfor bør en af attributterne være registreringsdato, så det er muligt at få information om hvilke krav, der er stillet til den

pågældende registrering. Derudover kan der indgå informationer om en række af de kvalitetselementer, der blev beskrevet i kapitlet om datakvalitet. Dette uddybes dog senere i kapitlet om den eksterne kontrol, kapitel 16.

Ud over attributter, kan en objektklasse indeholde en række funktioner som beskrevet i kapitel 7 om objektorienteret GIS. Det kan f.eks. være en funktion, der beregner den samlede værdi af bygningerne på en matrikel eller andre praktiske funktioner. Disse funktioner angives ligesom attributterne i datamodellen.

En objektklasse er opbygget således, at der stilles krav til hvilken type, en attribut kan være, f.eks. at attributten skal være et kommatall eller et heltal, samt om hvilke værdier der er tilladte. Disse krav specificeres i formatreglerne, der beskrives i det følgende.

#### 13.1.1.2 Formatregler

Specifikationen indeholder en række formatregler af forskellig karakter, der er en del af objektklasserne i datamodellen. Formatreglerne kan opdeles i to dele; attributregler og geometriske regler. En attributregel er f.eks. krav til, hvilken type en attribut skal være, dvs. om en attribut skal være af typen heltal, tekststreng eller noget helt andet. Det kan også være mere komplekse typer eller brugerdefinerede typer, som f.eks. geometri. Det er her, det er en fordel at benytte en objektorienteret database, hvor netop dette er muligt.

Attributreglerne består også af de regler, der angiver hvilket domæne, de enkelte attributter skal ligge inden for. Det kan f.eks. være, at der ikke er nogle koter, der ligger over 300m over havoverfladen i Danmark eller at bygninger med et grundplansareal på under 25m<sup>2</sup> ikke skal med i datasættet. Derudover kan denne form for domæner også angive nogle værdier, som en attribut kan have, hvilket beskrives i kapitel 15.

Formatreglerne består som nævnt også af en række geometriske regler, der har med geometrien for et rumligt objekt at gøre. De geometriske regler stiller krav til de forskellige geometrityper, f.eks. at en polygon skal være lukket og ikke må overlape sig selv. På samme måde må en linie heller ikke krydse sig selv. Dette er beskrevet tidligere i kapitel 10, og uddybes nærmere i kapitel 15.

#### 13.1.1.3 Relationelle regler

Ud over krav til hvad de enkelte objektklasser skal indeholde, angiver datamodellen også hvilke relationer, der skal være mellem forskellige



objekter fra de enkelte klasser. Disse relationer har primært betydning for brugen af data og ikke i så høj grad for produktionen hvis datagrundlaget f.eks. er et flyfoto. Derimod kan relationer have stor betydning inden for mange andre datakilder. Det kan være en relation mellem to rumlige objekter, der ikke har fysisk kontakt. For eksempel kan en adresse have relation til en vej, selvom adresse angives med et punkt, der ikke nødvendigvis ligger på vejen, men ved siden af den. På samme måde kan der være relationer mellem objekter, der ikke nødvendigvis kan repræsenteres i et kort. Det kan f.eks. være at en bygning har en relation til en person (ejer). Disse typer relationer mellem objektklasser angives i datamodellen.

### 13.1.2 Regler

Alle informationer i en specifikation kan betegnes som en form for registreringsregler for hvordan data skal genereres, lagres o.l. I ovenstående er det angivet hvilke objekter, der skal med i datasættet. Derudover indeholder specifikationen tre typer regler; topologiske regler, fortolkningsregler samt kvalitetsregler som vist på figur 13-1. Først beskrives fortolkningsreglerne.

#### 13.1.2.1 Fortolkningsregler

Fortolkningsregler er regler, der beskriver hvordan en operatør skal fortolke den information, der skal omsættes til objekter i datasættet. Denne type regler er regler, der ikke umiddelbart kan automatiseres, såsom at et dige skal registreres på midten af digets top. Overgangen mellem de regler, der ikke kan automatiseres, og dem der kan, er meget flydende. En regel som f.eks. at bygninger skal registreres ved tagudhæng, er en fortolkningsregel, og operatøren skal vurdere, hvor tagudhængen er, og hvad der er en bygning. Denne regel kan dog til en vis grad automatiseres, hvis data genereres ud fra et flyfoto, vha. billedgenkendelse. At være i stand til at genkende eksempelvis bygningstage, stiller store krav til de metoder, der benyttes, og på nuværende tidspunkt er menneskers fortolkningsevne stadig bedre end en computers. Dette vil sikkert ændre sig med tiden, og lige præcis genkendelse af afgrænsningen på bygningstage er nok et af de letteste objekter at genkende fra flyfotos, selvom der findes et utal af forskellige slags tage. Det er til gengæld noget sværere at klassificere hvilken type objekt, der er registreret, dvs. om det er en beboelsesbygning eller om det er en erhvervsbygning, når der benyttes flyfotos som datakilde. Der er altså en lang række problemer ved automatisk genkendelse, men nogle af dem vil formentligt blive elimineret i fremtiden.

Fortolkningsreglerne kan også betegnes som et sæt oversættelsesregler, der kan overføre en type data til en anden. Det kan være en oversættelse fra raster til vektor som når der benyttes flyfotos, men det kan også være fra andre former for vektordata f.eks. med en højere detaljeringsgrad. Her fortolkes disse data ved at fortælle hvilke objekter fra det ene datasæt, der skal overføres til det andet samt hvordan data skal generaliseres. På den måde er det i højere grad muligt at foretage processen automatisk vha. en række prædefinerede matematiske procedurer. En nærmere beskrivelse af disse muligheder gives i kapitel 14.

Som det ses er fortolkningsreglerne meget afhængige af, hvad der benyttes som datagrundlag for produktionen. Derfor er det også fortolkningsreglerne, der bestemmer hvilke data, der skal benyttes som datagrundlag, dvs. om det skal være flyfotos eller anden form for kilde. På samme måde er det også i fortolkningsreglerne det angives hvordan et produkt skal produceres samt hvilke metoder, der skal benyttes hertil.

#### 13.1.2.2 Topologiske regler

I specifikationen defineres det ligeledes hvordan forskellige objekters rumlige placering skal være i forhold til hinanden. Dette gøres vha. regler for hvilke topologiske relationer, der må være mellem objekter. Disse topologiske regler kan kontrolleres ved at opstille relationerne matematisk som beskrevet i kapitel 9.

En topologisk regel kan f.eks. være, at en vej ikke må skære en bygning, eller en bygning ikke må ligge på mere end én matrikel. Derudover kan en topologisk regel også være at to bygninger, der ligger tæt op af hinanden skal registreres som én bygning. Disse regler kan alle kontrolleres matematisk, men det kan være, at der skal en fortolkning til, for at rette de fejl der måtte opstå. Ved hjælp af en fortolkning er det muligt, i henhold til eksemplerne, at rette bygningen, vejen eller matriklen, så de overholder de opstillede regler.

Der findes en lang række topologiske relationer som beskrevet tidligere. I et senere kapitel gives der derudover nogle eksempler på, hvordan disse kan implementeres og kontrolleres automatisk. De topologiske regler kan kontrolleres i det øjeblik et objekt er registreret og ved at sammenligne med de relevante objekter der allerede er registreret. Kontrollen af de topologiske regler beskrives i kapitel 15 om den interne kontrol.

### 13.1.2.3 Kvalitetsregler

Alle de ovenstående regler skal selvfølgelig overholdes, men for nogle af reglernes vedkommende, er det acceptabelt hvis de kun overholdes for en vis andel af datasættet. Det kan f.eks. være at fuldstændigheden af objekter fra objektklassen Drivhuse kun behøver at være 97 %. Herved er det ikke nødvendigt at overholde reglen om, at et drivhus skal registreres i alle tilfælde. Kvalitetsreglerne er angivet for at beskrive et minimum for, hvordan datasættet skal være, og ikke for at begrænse kvaliteten. Under fortolkningen af datagrundlaget ønskes det at alle drivhuse registreres, men kvalitetsreglerne angiver at det er i orden hvis dette ikke overholdes fuldt ud. Et andet krav kan være, at registrerede punkter maksimalt må have en middelfejl på 1 meter i planen og koten. Dette betyder, at resultatet gerne må være bedre, men datasættet som minimum skal overholde dette krav.

Denne type regler er alle nogle, der har med datakvalitet at gøre, som beskrevet tidligere. Kvalitetsreglerne er af forskellig art og kan vurderes flere forskellige steder, f.eks. under selve produktionen af datasættet i den interne kontrol, eller efterfølgende i den eksterne kontrol. De kvalitetselementer, der fremkommer af datasættet i den eksterne kontrol, skal sammenlignes med de krav, der er angivet i specifikationen. Derudover er de krav, der stilles i specifikationen til kvaliteten af datasættet med til at definere, hvad datasættet skal overholde.

Kvalitetsreglerne kan opdeles i to dele. Den ene del er en beskrivelse af hvilke af de kvalitetselementer, der er beskrevet i kapitlet om datakvalitet, der skal angives i data, mens den anden del beskriver kravene til de enkelte elementer. Det betyder at den første del bestemmer at f.eks. fuldstændigheden skal registreres, mens den anden del bestemmer, at denne fuldstændighed skal være minimum 97 % for eksempelvis drivhuse.

Begge dele skal kontrolleres, og denne kontrol foregår efter datasættet er produceret i den eksterne kontrol, som illustreret på figur 12-1. Nogle af kvalitetselementerne kan først registreres ved denne eksterne kontrol mens andre elementer kan angives under registreringen af et objekt, eller lige efter. I et tidligere kapitel er det forklaret, hvor de forskellige kvalitetselementer skal registreres, og dette beskrives nærmere i forbindelse med de enkelte dele af produktionsprocessen, i de følgende kapitler.

### 13.1.3 Afslutning

Ved at etablere en datamodel ud fra specifikationen på en stringent måde, er det muligt at sikre, at alle krav kommer med. Datamodellen skal indeholde de objekter, der er beskrevet i specifikationen, samt de tilhørende regler.

Selve datamodellen beskriver strukturen i data, og skal ses som en helhed under produktionen, mens reglerne skal benyttes forskellige steder i produktionen som vist på figur 12-1.

I det følgende beskrives det hvordan en specifikation kan udarbejdes, så den bliver klar og tydelig for eksempelvis operatøren, der benytter den.

## 13.2 Udarbejdelse

Der er mange måder at foretage selve udarbejdelsen af en specifikation på. Specifikationen skal udarbejdes på baggrund af hvordan det ønskes, at data skal repræsentere den virkelige verden, som beskrevet tidligere. Ønskerne til data kan stamme mange forskellige steder fra. Det er hovedsageligt dem, der skal bruge data som et produkt, der kommer med disse ønsker.

Under udarbejdelsen af specifikationen, er det processen med at omforme ønskerne til en specifikation, der er det største problem. For at løse dette problem kan der opstilles en række rutiner, der gør det lettere for dem, der udarbejder specifikationen, at overføre ønskerne til specifikationen. Disse rutiner kan have forskellig karakter afhængig af hvilken slags ønsker der er tale om. Det kan være nogle standard spørgsmål, der stilles for at få en struktureret specifikation [Christiansen & Olsen, 2003], og derigennem sørge for at alle facetter i specifikationen klarlægges. En anden mulighed er at tage udgangspunkt i en eksisterende lignende specifikation og udvide den til også at dække de nye ønsker. Der er fordele og ulemper ved begge disse metoder.

Fordelene ved at benytte en eksisterende specifikation, er at mange af de grundlæggende krav, allerede er definerede og fokus kan lægges på de ting, der skal ændres. Herved undgås en masse arbejde ved at skulle konkretisere meget simple ting igen og igen. Der kan opnås et endnu bedre resultat, hvis brugerens ønsker også tager udgangspunkt i en eksisterende specifikation, og brugeren har kendskab til den på forhånd.

Omvendt rettes de småproblemer, der kan have været ved den eksisterende specifikation, ikke nødvendigvis hvis en eksisterende

specifikation benyttes. Derudover kan det være svært at ændre de grundlæggende elementer i specifikationen, uden at skulle lave det hele om. Det kan f.eks. være at ting, der er ændret i den nye datamodel, har indflydelse på de nogle af de relationer eller regler, der eksisterede mellem objekter fra den gamle datamodel, og som ikke indgår længere. På den måde risikeres det, at der relateres til objekter, der slet ikke indgår i datasættet. Det kan også være at ændringerne gør, at nogle kontroller, der ikke direkte har berøring med ændringerne, ikke fungerer, f.eks. hvis der benyttes en værdi fra en objektklasse, der ikke eksisterer mere. Derfor er det mange gange nødvendigt at starte helt forfra med en ny specifikation, når et produkt skal produceres. Det er selvfølgelig afhængigt af hvor stor forskel der er på den eksisterende og den nye specifikation og hvor god den eksisterende er.

### 13.2.1 Metode

Problemet ved etableringen af en specifikation er, at den er baseret på ideer specificeret af mennesker og benyttes af mennesker [Christiansen & Olsen, 2003]. Når der er mennesker indblandet i udarbejdelsen af en specifikation, er der også stor mulighed for at der opstår fejl såsom misforståelser eller mangler. Dette kan løses ved at udarbejde og kontrollere specifikationen ud fra et sæt fastlagte steps og arbejdsgange som det f.eks. gøres i SQC [Gilb, 2002] som beskrevet i [Christiansen & Olsen, 2003]. Denne form for rutiner kan lægges ind i en selvstændig applikation, og derved gøre den proces, der skal til for at skrive en specifikation, mere homogen og standardiseret. Dette er bl.a. gjort i [Cockcroft, 1998], der har udviklet et system, hvor det er muligt at angive nogle ønsker til hvad en specifikation skal indeholde. Der gives nogle få eksempler på hvordan datamodel og regler kan indtastes i deres applikation. Datamodellen oprettes ved at definere objektklasser med tilhørende attributter samt regler for de pågældende attributter. Derudover kan der ligeledes angives regler for de forskellige objekter. Det er kun topologiske regler der er implementeret i Cockcrofts eksempel, men en udvidelse burde være mulig.

Fordelen ved den metode der anvendes i [Cockcroft, 1998] er at den benytter en standardiseret database til at gemme specifikationen i, hvilket gør den programafhængig, og den kan derfor i princippet benyttes i forbindelse med enhver GIS-applikation. En anden fordel er at det, sammen med indtastning af regler for attributterne, er muligt at indtaste en tekst, der beskriver reglen med ord så reglen kan læses af alle.

Mange af de samme ting som er beskrevet i [Cockcroft, 1998] er ligeledes mulige i ArcGIS [Zeiler, 2001] [MacDonald, 2001]. Her er det muligt at oprette objektklasser med attributter og tilhørende domæneregler. Derudover er det også muligt at angive forskellige relationelle og topologiske regler som i [Cockcroft, 1998] samt meget andet, hvilket beskrives i kapitel 15, samt i appendiks 2, 3 og 4.

De to applikationer giver ikke mulighed for at definere fortolkningsregler og er derfor rettet mod eksisterende data og ikke en produktion af data. Princippet bag de to systemer kan stadig benyttes under en produktion, men det kræver, at der tilføjes en mulighed for at angive fortolkningsregler, der kan oversætte en datakilde til et produkt, som beskrevet tidligere. Det kan f.eks. gøres ved at skrive reglerne i tekstform som en operatør får op på skærmen eller skrive dem i et sprog, som fortolkeren kan forstå. Dette beskrives nærmere i det følgende kapitel.

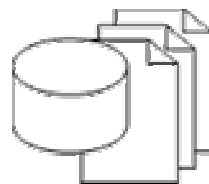
## 14 Fortolkning

Ud fra kravene, der er opstillet i specifikationen, skal der foretages en fortolkning af et givet datagrundlag, for at generere et produkt bestående af geografiske data. Der findes mange måder at indsamle information til produktionen af geografiske data. Der er sket en udvikling gennem mange år med brugen af teodolitter, flyfotos, GPS og laserscanning, og alle disse metoder benyttes i vid udstrækning i dag. Til generering af Top10dk benyttes der i dag udelukkende flyfotos [Gottlieb, 2003], men det er muligt at forestille sig, at andre former for datakilder kan anvendes som supplerende datagrundlag eller helt erstatte flyfotos.

### 14.1 Datagrundlag

Datagrundlaget kan være af forskellig art, lige fra det mest traditionelle, når der tales om geografiske data, flyfotos, til andre former for data, såsom observationer foretaget med GPS eller laserscanning. En anden type data, der kan ligge til grund for produktionen af geografiske data, kan være eksisterende data, så det ikke er nødvendigt at foretage nye registreringer. Hvis disse eksisterende data har en bedre nøjagtighed og/eller detaljering end det datasæt, der produceres, vil datakilden være velegnet. Det kan både være som den eneste kilde eller som supplement til en anden datakilde. Eksempelvis kan en datakilde give ændringsudpegning af bygninger, idet kilden indeholder en præcis liste over alle bygninger. Denne liste kan angive et enkelt punkt for hver bygning, og en anden datakilde kan da benyttes til at registrere bygningsafgrænsningerne.

Datagrundlaget er illustreret som på figur 14-1 i den opstillede produktionsmodel på Figur 12-1. Dette skyldes, at datagrundlaget kan bestå af mange forskellige typer datakilder, lige fra eksisterende databaser til papirkort. Derudover kan datagrundlaget også være den virkelige verden, f.eks. hvis en operatør går ud og ser på de fysiske ting, der skal repræsenteres i data. Dette kan være tilfældet når der benyttes GPS eller totalstation, hvorved digitaliseringen foretages ud fra fortolkninger af den virkelige verden. Lige



Figur 14-1  
Datagrundlag

meget hvilken metode der benyttes, er det nødvendigt med en række regler for, hvad det er, der skal registreres. Til dette benyttes fortolkningsreglerne.

## 14.2 Fortolkningsregler

For at være i stand til at omdanne datagrundlaget til det ønskede produkt, kræver det bl.a. en række fortolkningsregler. Fortolkningen foretages ud fra de krav, der stilles i specifikationen, hvor det bl.a. angives, hvilket datagrundlag der skal benyttes, samt hvilken registreringsmetode der skal anvendes. Herunder ses en liste over de aspekter der har indflydelse på fortolkningen.

- Datagrundlag
- Registreringsmetode
- Objektklasser
- Oversættelse
- Registreringsinstrukser

Informationer om datagrundlag og registreringsmetode indgår, sammen med viden om objektklasserne fra datamodellen, i den grundlæggende viden, der skal til for at være i stand til at foretage en registrering.

Derudover kræves en viden om hvilken objektklasse et objekt fra datagrundlaget skal fortolkes eller oversættes til. For eksempel skal der en fortolkning til, når en bygning i et flyfoto skal omsættes til en bygning i datasættet. Det kan være svært at beskrive denne oversættelse helt stringent, og det er derfor meget afhængig af den operatør, der foretager registreringen. Mennesket er stadig i de fleste tilfælde bedst til at fortolke billeder. Dette gør fortolkningsprocessen kompliceret, når der er tale om anvendelse af flyfotos som datagrundlag. Hvis der derimod benyttes eksisterende data som datagrundlag, kan det være lettere at automatisere denne oversættelse. Herved består oversættelsen kun i at angive hvilken objektklasse, i datagrundlaget, der svarer til en bestemt klasse i datasættet.

Fortolkningsreglerne beskriver bl.a. hvilken oversættelse, der skal foregå, når en entitet i datakilden skal overføres til et objekt i datasættet. Fortolkningen kan derudover også bestå af en



simplificering af de eksisterende data, f.eks. hvis detaljeringsgraden ikke ønskes så høj i det endelige produkt.

Ud over viden om fortolkningsreglerne og datamodellen, skal fortolkningen ske med en baggrundsviden om de resterende regler fra specifikationen. Det er f.eks. viden om de topologiske regler, samt viden om hvilke krav der stilles til kvaliteten af det endelige produkt. Viden om hvordan de forskellige objekter skal relatere til andre, har betydning for den måde de skal registreres på. Det kan f.eks. være at afstanden mellem samtlige punkter enten skal være større end én meter, eller også skal punkterne være sammenfaldende. En anden regel, der kan være nødvendig at kende til under fortolkningen er for eksempel, at en polygon skal være lukket og ikke må krydse sig selv. Det er vigtigt at operatøren kender til denne type regler, men de kan kontrolleres under den efterfølgende interne kontrol, der beskrives i næste kapitel. Herved kan operatøren koncentrere sig om at fortolke og overlade de geometriske og topologiske regler til den automatiske kontrol.

Det samme er gældende for den kvalitet der ønskes af det endelige produkt. Det kan være til gavn for en operatør, at vide hvor stor en nøjagtighed eller fuldstændighed, der ønskes. I de fleste tilfælde skal operatøren gøre sit bedste, men der kan være situationer, hvor det ikke er nødvendigt, for at overholde de krav, der stilles i specifikationen. Et eksempel på dette kan være, at det objekt, der repræsenterer et træ, ikke nødvendigvis skal registreres lige hvor stammen står, men som et punkt inden for trækronens udbredelse. Herved kan registreringen foretages hurtigere, end hvis målingen skal være helt præcis.

De enkelte krav, der stilles til fortolkningen, er afhængige af hinanden. Registreringsmetoden er eksempelvis i høj grad afhængig af hvilket datagrundlag, der benyttes og omvendt. Samtidig vil registreringsmetoden og dermed også datagrundlaget, have indflydelse på hvilke krav, der kan stilles til kvaliteten. Det er f.eks. ikke muligt at opmåle med centimeters nøjagtighed hvis datagrundlaget er satellitbilleder, foretaget i 600 km højde med en pixelstørrelse på 4 meter.

#### **14.2.1 Håndtering af fortolkningsregler**

Den måde hvorpå fortolkningsreglerne skal håndteres, er selvfølgelig afhængig af hvad datagrundlaget er og hvilken registreringsmetode, der benyttes. Hvis der tages udgangspunkt i en fotogrammetrisk opmåling, skal fortolkningsreglerne være i et sprog, som operatøren kender. Her er det nærliggende at benytte almindeligt sprog, men det

kan også være en specielt udviklet terminologi, til netop registrering af geografisk information.

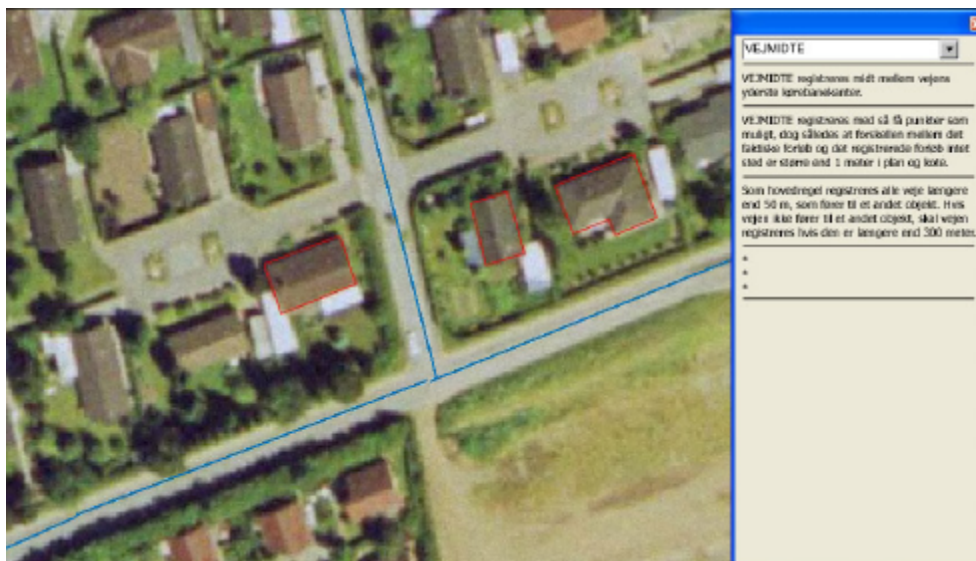
Håndteringen af fortolkningsregler kan opdeles i de følgende to dele:

- Oprettelse af regler
- Anvendelse af regler

Fortolkningsreglerne oprettes, som beskrevet tidligere under specifikationen, ud fra nogle givne rutiner, der sikrer at alle krav kommer med i specifikationen. Derudover er det nødvendigt, at operatøren kender til de resterende regler fra specifikationen. Derfor skal disse også skrives i et sprog, som operatøren kan forstå og overskue. Dette kan gøres ved at reglerne både skrives på matematisk form og på tekstform, således at både operatøren kan forstå dem, og de kan benyttes i en applikation.

Hvis reglerne er skrevet på tekstform, kan operatøren have reglerne stående ved siden af de billeder, der registreres fra, og derved opnå et bedre overblik. Denne fremgangsmåde til registrering af objekter vha. fotogrammetri, er i princippet sådan det gøres i langt de fleste tilfælde i dag. Det, der er fordelene ved denne metode er, at operatøren ikke skal koncentrere sig om at overholde eksempelvis de topologiske regler, da dette kontrolleres efterfølgende, når et objekt er registreret.

Et eksempel på hvordan den beskrevne metode kan implementeres, er illustreret på nedenstående figur 14-2. Her er det muligt for operatøren at se de relevante regler for den type objekt, der er ved at blive registreret. Helt præcist hvordan reglerne skal vises, afhænger i høj grad af hvilket fotogrammetrisk system, der anvendes.



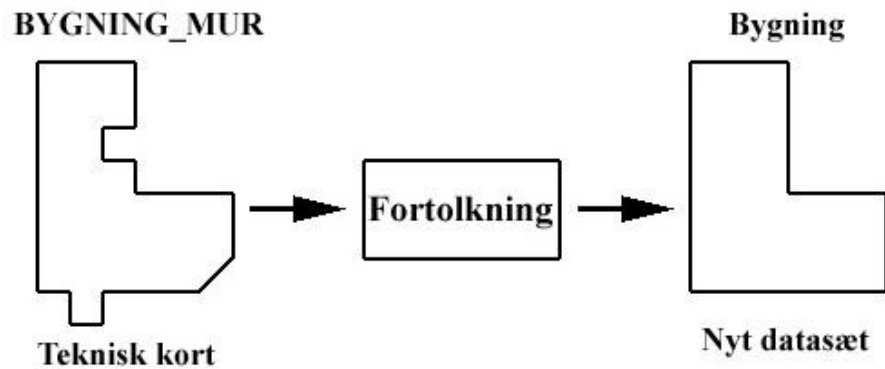
Figur 14-2 Eksempel på implementering af fortolkningsregler.

På figuren er operatøren i gang med at registrere et objekt fra klassen Vejmidte, men det er muligt at vælge en anden klasse vha. drop-down menuen øverst til højre. Til højre ses ligeledes nogle af de tilhørende regler, som objekter fra denne klasse skal overholde. De regler, der vises til højre, har ingen indflydelse på selve registreringen, bortset fra at det kun er de relevante fortolkningsregler, der vises. Der foretages altså ikke en automatisk kontrol af om disse regler overholdes.

Ovenstående eksempel viser hvordan det kan se ud hvis datagrundlaget er flyfotos. Hvis der derimod benyttes en anden datakilde som datagrundlag, f.eks. nogle andre eksisterende data, er det muligt at foretage store dele af fortolkningen automatisk.

Et eksempel på brugen af eksisterende data kan være, at benytte et digitalt teknisk kort som datagrundlag, som illustreret på figur 14-3. Her sker fortolkningen af en bygning fra et teknisk kort, med mange detaljer, til et mere simpelt objekt i det nye datasæt.

Fortolkningen består i dette tilfælde af to dele. Den første del er en oversættelse fra objektklassen "BYGNING MUR" fra det teknisk kort til objektklassen "Bygning" i det nye datasæt. Denne oversættelse er altså til for at relatere en objektklasse i datakilden til en objektklasse i det nye datasæt.



Figur 14-3 Fortolkning af en bygning fra et teknisk kort til et mere generaliseret datasæt.

Den anden del består af en simplificering af objektet. Simplificeringen kan i de fleste tilfælde foretages automatisk, selvom det kan være en kompliceret proces. Denne simplificering er et projekt i sig selv og beskrives ikke nærmere i denne rapport, men en række eksempler er givet i [Christiansen & Olsen, 2002].

Både ved anvendelse af flyfoto og eksisterende datakilder som datagrundlag, foretages registreringen ét objekt af gangen. Objektet skal efterfølgende kontrolleres for geometriske og topologiske fejl ud fra de gældende regler. Dette gøres, som sagt, under den interne kontrol, der beskrives i det følgende kapitel.

## 15 Intern Kontrol

Det endelige produkt skal helst være så fejlfrit som muligt. Derfor er det nødvendigt at forsøge at finde så mange fejl, så tidligt i produktionen som muligt [Christiansen & Olsen, 2003].

Data skal overholde de regler, der er defineret i specifikationen. Nogle af disse regler kan automatisk kontrolleres i den interne kontrol, når et objekt er registreret under fortolkningen. De ting, der kan kontrolleres her, er bl.a. formatreglerne, dvs. om objektets attributter har værdier, der er inden for de domæner, der er specificeret, og at de har det rigtige format, f.eks. at attributten "kote" er en talværdi. Desuden kan det undersøges om de geometriske regler overholdes, samt at der ikke er topologiske fejl i data. Den logiske konsistens af data kan beskrives vha. et af kvalitetselementerne, der er opstillet tidligere i kapitel 11. Formålet med de regler og kontroller, der beskrives i det følgende er altså, at forbedre konsistensen af data, samt finde yderligere fejl, hvis der er mulighed for det. Disse yderligere fejl kan være sket i forbindelse med fortolkningen, og kan f.eks. være fejltolkninger eller manglende værdier.

De kontroller, der foretages i den interne kontrol, og som beskrives her, skal foregå automatisk. Om kontrollen kan foretages automatisk, afhænger af to ting. For det første hvilken type kontrol der er tale om. For det andet hvilken teknologi der er til rådighed. Når der er fundet en fejl, er det også fordelagtigt, at den bliver rettet. Hvis den kan rettes automatisk, er det optimalt, men da dette ikke altid kan lade sig gøre, kan fejlene også markeres, og derefter rettes manuelt.

Formålet med den interne kontrol, er altså at forhindre fejl, at finde de fejl, der automatisk kan detekteres, og rette de fejl, der automatisk kan rettes.

I det følgende beskrives en række kontrolmetoder inden for de forskellige typer regler.

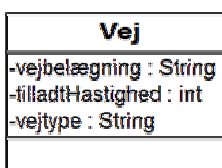
### 15.1 Formatregler

Formatreglerne består af to dele, attributregler og geometriske regler. Attributregler sørger for, at objekternes attributter er i overensstemmelse med de krav, der er opsat i specifikationen. I det følgende beskrives de værktøjer, der kan benyttes, for at reglerne er overholdt. De geometriske regler sørger for at objektets geometri

ligeledes stemmer overens med de regler, der er angivet i specifikationen, som beskrevet i kapitel 13.

### 15.1.1 Attributregler

De mulige attributregler belyses igennem det følgende eksempel. En klasse, Vej, i et datasæt, har forskellige attributter tilknyttet, f.eks. ”vejbelægning”, der er en tekststreng og ”tilladt hastighed”, der er en heltalsværdi. Vejnettet består af nogle forskellige vejtyper, såsom



Figur 15-1 Klassen Vej

motorvej, hovedvej og bivej. Klassen Vej er illustreret på figur 15-1.

Hver enkelt attribut er af en bestemt type, f.eks. heltal. Den mest fundamentale formatregel er, at alle objekter har korrekte værdier i de relevante attributter. Det ville være en fejl, hvis der stod en tekststreng i stedet for en talværdi i attributten ”tilladt hastighed”. Heldigvis er det muligt at definere hvilke datatyper, de forskellige attributter har, og automatisk tjekke at indtastningen er korrekt, i de fleste almindelige GIS-systemer, såsom ArcGIS og MapInfo.

Vejtyperne har hver især nogle grænser for, hvilke værdier attributterne kan have. Det kan eksempelvis defineres, at motorvej kan have hastigheder fra 90-130 km/t, hovedvej 50-90 km/t og bivej 30-60km/t. For at sikre at denne værdi er registreret korrekt, er det muligt at lægge nogle restriktioner på, hvilke værdier det er muligt at tildele attributten. Det kan f.eks. være at hastighed skal være mellem 30 og 130 km/t. En sådan regel kaldes et værdimængdedomæne (range domain). Den slags regler kan kun etableres for talværdier og datoer [MacDonald, 1999]. Attributten vejbelægning er en tekststreng, og det er derfor ikke muligt at benytte værdimængdedomæner for den attribut. Det er alligevel muligt at definere nogle regler for hvilke værdier, vejbelægning kan have, ved at benytte et kodet værdidomæne (coded value domain). Her defineres en række værdier, som attributten kan have, og når data herefter skal indtastes i databasen, er det kun muligt at vælge mellem de værdier der i forvejen er defineret. Det kan f.eks. være ”grus”, ”brosten”, ”asfalt” og ”beton”.

De fleste veje har belægning af asfalt, og derfor kan det være fordelagtigt at en ny vej i databasen automatisk fik tildelt værdien "asfalt" i attributten vejbelægning. Dette kan gøres ved at definere standardværdier (default values). Det kan både gøres for værdimængdedomæner, kodede værdidomæner og andre attributter, der ikke har tilknyttet domænerregler. Attributten kan derefter ændres til en anden valid værdi, hvis det er nødvendigt.

Som beskrevet tidligere, er der også forskellige vejtyper, og disse typer har hver især deres specielle værdidomæne for attributten "tilladt hastighed". En klasse som "Vej" kan i sådanne tilfælde underinddeles i subtyper, der hver har deres egne domæneværdier, standardværdier og andre regler. På den måde er det altså muligt, eksempelvis at definere subtypen "motorvej", der har værdimængdedomænet 90-130km/t og standardværdien 110 km/t for attributten "tilladt hastighed". Derudover har subtypen værdierne "asfalt" og "beton" som kodet værdidomæne og defaultværdien "asfalt" for attributten vejbelægning. På den måde sikres det, at det er lettere for operatøren, at foretage en korrekt registrering af værdierne for motorvej.

Denne form for kontrol af indtastede data, er ikke en egentlig kontrol, men mere en restriktion, så det ikke er muligt for operatøren at indtaste data, der helt sikkert er forkert. Dette er derved med til at gøre det lettere for operatøren at foretage korrekte registreringer. Det er altid bedre at forhindre fejl i at opstå, end at finde dem senere hen, men det er selvfølgelig også bedre at finde og rette fejlene end at levere et produkt, der indeholder fejl.

I Appendix 2, beskrives det bl.a., hvordan formatregler oprettes i ArcGIS.

### **15.1.2 Geometriske regler**

Objekter i rumlige databaser har ofte enten relationer til objekter, med rumlige geometriske attributter, eller har selv rumlige geometriske attributter. Denne geometri kan, som beskrevet tidligere, være f.eks. punkt, linie eller polygon. Specifikationen definerer hvordan disse geometriske objekter skal registreres, og hvordan de kan se ud. Det kan f.eks. være, at en linie ikke må skære sig selv, eller at der ikke må være hul i en polygon. De fejl, der kan opstå, er beskrevet nærmere i kapitel 10. Det er muligt at identificere disse fejl, ved at opstille regler, som f.eks. kan være en algoritme, der undersøger, om en linie skærer sig selv. Det ville være forholdsvis simpelt at implementere. Linien kan deles op i liniestykker, og det kan undersøges, om nogle af disse liniestykker skærer hinanden.

Det bør undersøges om disse regler er overholdt samtidig med at registreringen foregår. De geometriske regler svarer i mange henseender til de topologiske regler, der beskrives i et senere afsnit, hvad angår metoder til at undersøge om de er overholdt.

## 15.2 Kvalitetsregler

Det er nødvendigt, at de kvalitetslementer, der skal indgå som en del af data, også gør det. I den interne kontrol kan det undersøges, om de kvalitetslementer, der skal registreres under fortolkningen af data, er til stede. For det første er det muligt at undersøge, om de rent faktisk er registrerede, og for det andet er det muligt at undersøge om der er sandsynlighed for, at de er korrekte. Det er simpelt at undersøge om de er blevet registreret. Det er bare at undersøge, om der er tilføjet data på de steder, hvor kvalitetslementerne skal lagres. Det kan endvidere undersøges om det er sandsynligt, at de tidsangivelser, der er angivet i datasættet, er korrekte. Det kan gøres ved at tjekke, om tidsangivelserne er inden for en prædefineret tidsramme, f.eks. en time før og ikke efter det nuværende tidspunkt. Kvalitetslementerne afstamning og formål er sandsynligvis de samme, for en begrænset del af datasættet, og derfor kan dette sættes som standardværdi, som beskrevet ovenfor i afsnittet om attributregler. Præcision hænger sammen med afstamning, idet det er den metode, der benyttes til at registrere data, der beskrives i afstammingsoplysningerne, og præcisionen kvantificerer denne metodes forventede nøjagtighed. På den baggrund er det muligt at opstille et værktøj, hvor operatøren kan angive den metode, der benyttes til dataindsamling, og automatisk bestemme præcisionen og udfylde de to elementer. På den måde kan det sikres, at disse informationer passer til hinanden, og at de er med i data.

## 15.3 Topologiske regler


I specifikationen er der også beskrevet en række topologiske regler, der skal overholdes. De topologiske regler kan dels finde frem til registreringsfejl, hvor operatøren har foretaget en forkert opmåling, eller klassificeret et objekt forkert. Derudover er det også muligt, at finde fejl, der hvor data ikke er i overensstemmelse med specifikationen. Det er dog ikke alle disse fejl, der kan findes på baggrund af de topologiske regler.

I det følgende beskrives de funktioner, der skal bruges for at undersøge om den topologiske sammenhæng er korrekt. Til dette bruges bl.a. 9-skæringsmatricemodellen, der er beskrevet tidligere.



Derefter følger en beskrivelse af et værktøj, der kan benyttes til at opbygge et topologisk regelsæt for data.

Som beskrevet i kapitel 9, er der nogle områder hvor den simple 9-skærings-matricemodel, ikke slår til. En af løsningsmulighederne til dette, er at benytte den Dimensionalt Udvidede 9-SkæringsMatricemodel (DU9SM-model), der er beskrevet i [OpenGIS, 1999]. Her angives dimensionen af skæringen som -1, 0, 1 eller 2. Det vil altså sige, at de steder, hvor der står 1 i den simple skæringsmatrice, vil der stå 0, 1 eller 2 i den dimensionalt udvidede skæringsmatrice. De steder hvor der står 0 vil der stå -1. Hvis fællesgeometrien mellem to dele f.eks. er todimensional, vil der stå 2 i DU9SM-modellen. Den simple skæringsmatricemodel er beskrevet i Appendiks 1. Ved skæring 511, der er illustreret i tabel 15-1, ses den simple og den dimensionalt udvidede 9-skæringsmatricemodel.

Nr.	Eksempel	Simpel	Dimensionalt udvidet
511		$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 2 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 2 \end{pmatrix}$

Tabel 15-1 Skæringsmatricer for skæring nr. 511.

OpenGIS konsortiet har forsøgt at gruppere de relationer, der kan være mellem to geometriske objekter, i syv forskellige funktioner. I det følgende gennemgås først nogle værktøjer til at bestemme relationen mellem to objekter, og derefter beskrives de syv funktioner.

Det første værktøj, der skal bruges til at bestemme relationen mellem to geometriske objekter, er en funktion, der kan bestemme skæringsgeometrien mellem de to. Denne funktion kaldes *Intersect*. Funktionen returnerer fællesgeometrien, og det er herved muligt, at bestemme dimensionen af denne. En anden funktion, der er nyttig er *Equal*, der benytter *Intersect* til at bestemme, om to geometriske objekter *X* og *Y* er ens. De to objekter er altså ens, hvis:

$$Intersect(X, Y) = X \wedge Intersect(X, Y) = Y$$

Det er muligt at opstille en funktion, der udtrykker to geometriske objekters relation til hinanden, og som benytter DU9SM-modellen.

Her kaldes denne funktion *Relate*. Denne funktion tilhører klassen geometri, og tager som operatører to parametre, den anden geometri, der relateres til, samt en tekststreng på 9 tegn, der repræsenterer skæringsmatricen. For en skæring  $a$  er de mulige værdier  $p$  for denne tekststreng  $\{T, F, *, 0, 1 \text{ og } 2\}$  hvor:

$$p = T \Rightarrow \dim(a) \in \{0,1,2\} \quad p \neq \emptyset$$

$$p = F \Rightarrow \dim(a) = -1 \qquad p = \emptyset$$

$$p = * \Rightarrow \dim(a) \in \{-1,0,1,2\} \qquad \text{alle muligheder}$$

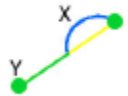
$$p = 0 \Rightarrow \dim(a) = 0 \qquad 0\text{-dimensional skæring}$$

$$p = 1 \Rightarrow \dim(a) = 1 \qquad 1\text{-dimensional skæring}$$

$$p = 2 \Rightarrow \dim(a) = 2 \qquad 2\text{-dimensional skæring}$$

Dette er beskrevet nærmere i [OpenGIS, 1999]. Denne funktion kan realisere alle de skæringsmatricer, der er beskrevet i Appendiks 1. Det vil desuden være fordelagtigt at udvide denne notation, således at det også er muligt at angive, om  $p$  er en geometrisamling  $G$ , f.eks. hvis resultatet af *intersect* giver et punkt og en linie ligesom i eksempelvis skæring 341, som vist på figur 15-2. Dette kan f.eks. være:

$$p = G \Rightarrow \dim(a) = \{0,1,2\} \wedge a = \text{kompleks geometri}$$

Nr.	Eksempel	Simpel	Dimensionalt udvidet
341		$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & F & 1 \\ F & G & F \\ 1 & F & 2 \end{pmatrix}$

Tabel 15-2 Skæring 341 og skærings-matricen for denne.

En funktion, der kan optælle antallet af geometriske objekter i den komplekse geometri kan også være hensigtsmæssig, da det herved er muligt at bestemme antallet af skæringer mellem to objekter. Denne funktion kan f.eks. kaldes *GeomCount*.

### 15.3.1 OpenGIS-funktioner

I det følgende beskrives de funktioner, der er defineret af OpenGIS konsortiet, idet der gøres et forsøg på at gruppere relationerne.

## 15.3.1.1 Disjoint

De to involverede geometriske objekters kanter eller indre skærer ikke hinanden.

$X.Disjoint(Y)$

$$\Leftrightarrow (\circ X \cap \circ Y = \emptyset) \wedge (\circ X \cap \partial Y = \emptyset) \wedge (\partial X \cap \circ Y = \emptyset) \wedge (\partial X \cap \partial Y = \emptyset)$$

$$\Leftrightarrow X.Relate(Y, 'FF*FF****')$$

## 15.3.1.2 Touches

Denne relation kan forekomme mellem alle objektpar, undtagen mellem to punkter. Kanten af det ene eller den andet objekt rører det andet objekts kant eller indre, mens de to objekters indre ikke rører hinanden. Dette udtrykkes som:

$X.Touches(Y)$

$$\Leftrightarrow (\circ X \cap \circ Y = \emptyset) \wedge ((\partial X \cap \circ Y \neq \emptyset) \vee (\circ X \cap \partial Y \neq \emptyset) \vee (\partial X \cap \partial Y \neq \emptyset))$$

$$\Leftrightarrow X.Relate(Y, 'FT*****') \vee X.Relate(Y, 'F*T*****')$$

$$\vee X.Relate(Y, 'F***T*****')$$

## 15.3.1.3 Crosses

Denne relation kan forekomme mellem en linie og en linie eller en linie og en polygon. For de tilfælde hvor X er en linie og Y er et polygon udtrykker denne relation de tilfælde, hvor det indre af X både krydser det indre og det ydre af Y. Dette gives som:

$$X.Crosses(Y) \Leftrightarrow (\circ X \cap \circ Y \neq \emptyset) \wedge (\circ X \cap -Y \neq \emptyset)$$

$$\Leftrightarrow X.Relate(Y, 'T*T*****')$$

Hvis begge objekter er linier, er det de tilfælde, hvor de to liniers indre netop krydser i et punkt. Relationen udtrykkes da som:

$$X.Crosses(Y) \Leftrightarrow \dim(\circ X \cap \circ Y) = 0$$

$$\Leftrightarrow X.Relate(Y, '0*****')$$

## 15.3.1.4 Within

Denne relation kan forekomme mellem to geometriske objekter, hvor X har en udstrækning, der er den samme som eller større end dimensionen af Y. Dette omfatter de objekter, der ligger helt inde i et

andet objekt, dvs. at det indre af X krydser det indre af Y, men hverken det indre eller kanten af X krydser det ydre af Y:

$$\begin{aligned} X.Within(Y) \\ \Leftrightarrow (\textcircled{X} \cap \textcircled{Y} \neq \emptyset) \wedge (\textcircled{X} \cap -Y = \emptyset) \wedge (\partial X \cap -Y = \emptyset) \\ \Leftrightarrow X.Relate(Y, 'T * F ** F ***') \end{aligned}$$

#### 15.3.1.5 Overlaps

Denne relation kan kun forekomme mellem to geometriske objekter der har samme udstrækning, dvs. eksempelvis en linie og en linie. To punkter eller to polygoner overlapper hvis begge objekters indre skærer det andet objekts indre og ydre. Relationen er da defineret som:

$$\begin{aligned} X.Overlaps(Y) \\ \Leftrightarrow (\textcircled{X} \cap \textcircled{Y} \neq \emptyset) \wedge (\textcircled{X} \cap -Y \neq \emptyset) \wedge (-X \cap \textcircled{Y} \neq \emptyset) \\ \Leftrightarrow X.Relate(Y, 'T * T *** T ***') \end{aligned}$$

For to linier skal de to indres fællesgeometri have en udstrækning i én dimension (være en linie), i modsætning til crosses, men ellers er relationen som ovenfor, hvilket vil sige:

$$\begin{aligned} X.Overlaps(Y) \\ \Leftrightarrow (\dim(\textcircled{X} \cap \textcircled{Y}) = 1) \wedge (\textcircled{X} \cap -Y \neq \emptyset) \wedge (-X \cap \textcircled{Y} \neq \emptyset) \\ \Leftrightarrow X.Relate(Y, '1 * T *** T ***') \end{aligned}$$

#### 15.3.1.6 Intersects

Denne relation betegner en generel skæring mellem objekterne X og Y, dvs. at der forekommer en eller anden skæring mellem de to objekters indre eller kant:

$$X.Intersects(Y) \Leftrightarrow !X.Disjoint(Y)$$

#### 15.3.1.7 Contains

Her undersøges det, om X indeholder Y. Det svarer til, at Y ligger inden i X:

$$X.Contains(Y) \Leftrightarrow Y.Within(X)$$

En anden nyttig funktion, der er beskrevet af OpenGIS konsortiet, er de binære funktioner for geometri; *Intersect*, *Difference*, *Union*, *SymDifference*, *Buffer* og *ConvexHull*. Disse kan også benyttes til at undersøge topologiske relationer mellem objekter. Desuden er der

beskrevet en funktion *Distance*, der bestemmer afstanden mellem to objekter. Dette kan bruges, hvis der er defineret en minimumsafstand mellem objekter i data. I Top10dk skal objekter, der ligger mindre end 1 meter fra hinanden registreres med fællesgeometri, dvs. de to objekters geometri skal være sammenfaldende, dér hvor de løber tæt på hinanden.

I tabellen herunder er det angivet hvilke af de funktioner, som netop er beskrevet, der dækker hvilke topologiske relationer fra Appendiks 1. Objekterne er hhv. punkt (Pu), linie (Li) eller polygon (Po).

Funktion	Objekter	Relationsmatrice	Topologiske relationer
disjoint	Pu/Pu	'FF*FF*****'	79
	Pu/Li	'FF*FF*****'	79
	Pu/Po	'FF*FF*****'	79
	Li/Li	'FF*FF*****'	79
	Li/Po	'FF*FF*****'	79
	Po/Po	'FF*FF*****'	79
touches	Pu/Pu	-	-
	Pu/Li	'*FT*****' v 'F**T*****' v 'F***T*****'	151
	Pu/Po	'*FT*****' v 'F**T*****' v 'F***T*****'	151
	Li/Li	'*FT*****' v 'F**T*****' v 'F***T*****'	85, 95, 103, 111, 119, 205, 207, 221, 229, 231, 237, 239, 245
	Li/Po	'*FT*****' v 'F**T*****' v 'F***T*****'	87, 95, 151, 207, 215, 223
	Po/Po	'*FT*****' v 'F**T*****' v 'F***T*****'	95

Funktion	Objekter	Relationsmatrice	Topologiske relationer
crosses	Pu/Pu	-	-
	Pu/Li	'T*T*****'	295
	Pu/Po	'T*T*****'	295
	Li/Li	'O*****'	335, 341, 351, 359, 367, 375, 461, 463, 477, 485, 487, 493, 495, 501
	Li/Po	'T*T*****'	463, 471, 479, 487, 495, 503
	Po/Po	-	-
within	Pu/Pu	'T*F**F***'	273
	Pu/Li	'T*F**F***'	295
	Pu/Po	'T*F**F***'	295
	Li/Li	'T*F**F***'	273, 295, 311
	Li/Po	'T*F**F***'	279, 295, 311, 407, 423, 439
	Po/Po	'T*F**F***'	273, 295, 439
overlaps	Pu/Pu	'T*T***T**'	-
	Pu/Li	-	-
	Pu/Po	-	-
	Li/Li	'I*T***T**'	335, 341, 351, 359, 367, 375, 461, 463, 477, 485, 487, 493, 495, 501
	Li/Po	-	-
	Po/Po	'T*T***T**'	511
contains	Pu/Pu		273
	Pu/Li	-	-
	Pu/Po	-	-
	Li/Li		273, 457, 473
	Li/Po	-	-
	Po/Po		273, 457, 505
intersects	Pu/Pu	'T*****' v '*T*****' v '***T*****' v '*T*****'	alle undtagen 79
	Pu/Li	'T*****' v '*T*****' v '***T*****' v '*T*****'	alle undtagen 79
	Pu/Po	'T*****' v '*T*****' v '***T*****' v '*T*****'	alle undtagen 79

	Li/Li	'T*****' v '*T*****' v '***T*****' v '***T*****'	alle undtagen 79
	Li/Po	'T*****' v '*T*****' v '***T*****' v '***T*****'	alle undtagen 79
	Po/Po	'T*****' v '*T*****' v '***T*****' v '***T*****'	alle undtagen 79

Funktion	Objekter	Relationsmatrice	Topologiske relationer
equal	Pu/Pu	'TFFFTFFFT'	273
	Pu/Li	-	-
	Pu/Po	-	-
	Li/Li	'TFFFTFFFT'	273
	Li/Po	-	-
	Po/Po	'TFFFTFFFT'	273

Tabel 15-3 Funktionerne fra OpenGIS, deres relationsmatrice, og de topologiske relationer, der findes vha. funktionerne.

De typer relationer, hvor der indgår et punkt, er stort set de samme, uanset hvilken type det andet objekt har. For punkt-punkt-relationen, er der kun to muligheder i relationsskemaet, men alligevel er der 4 funktioner, hvor relation 273 indgår. For de to andre relationsgrupper punkt-linie og punkt-polygon er der ligeledes tre funktioner, hvor den samme relation, 295, indgår. Det er uhensigtsmæssigt, da det kan skabe forvirring om, hvilken funktion der skal benyttes til at undersøge relationen. Derfor ville det være fordelagtigt at se på disse objektpar for sig selv.


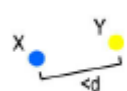

Der er mange forskellige relationstyper mellem to linier. Alligevel er der to af funktionerne herover, *crosses* og *overlaps*, hvor der indgår de samme relationer. Den eneste forskel er, at for *crosses* er fællesgeometrien et punkt, og for *overlaps* er fællesgeometrien en linie. Det er ikke muligt at undersøge om linien rent faktisk krydser, eller om den kun rører den anden linie i begge funktionerne. Her er der igen uhensigtsmæssigheder i funktionsmodellen. Det er selvfølgelig vigtigt at overveje, om det er nødvendigt at teste for alle de relationer, der er blevet gennemgået i det tidligere kapitel, da mange af dem minder meget om hinanden, men det bør være muligt at foretage en entydig topologisk undersøgelse vha. de definerede funktioner.

For relationen mellem en linie og en polygon, er modellen ganske hensigtsmæssig, idet funktionerne *touches*, *crosses* og *within* deler relationerne op i tre grupper, hvor relationerne minder om hinanden. Dette gør det let at benytte funktionerne til at bestemme relationen mellem to objekter med denne geometritype.

Mellem to polygoner er der otte mulige relationer, ifølge relationskemaet, der beskrives i Appendiks 1. Der er dog nogle vigtige ting, der ikke kan opdages med funktionerne, såsom om en polygon bliver delt i to dele af en anden polygon, dvs. at den ene polygon vil blive til to, hvis den anden bliver trukket fra vha. funktionen *Difference*. Det vil være mere hensigtsmæssigt at se på skæringerne mellem den ene polygon og den anden polygones indre, kant og ydre.

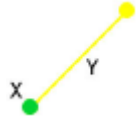
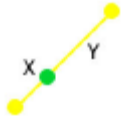
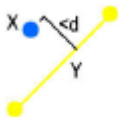
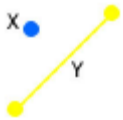
### 15.3.2 Alternative funktioner

På baggrund af kritikpunkterne ovenfor, vil det være aktuelt at forsøge at foretage en anderledes opdeling af relationerne. Dette gøres ved ikke at generalisere på samme måde som ovenfor, hvor alle objekter bruger samme funktioner, men hvor der i stedet laves individuelle funktioner for hvert enkelt objektpar. I det følgende beskrives en sådan opdeling. Disse funktioner skal ses som et forsøg på at gøre processen, med at definere topologiske regler, mere intuitiv, og beskrives i de følgende seks tabeller:

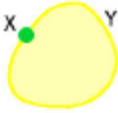
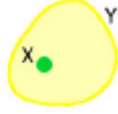
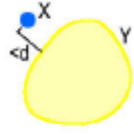

Punkt-punkt			
Navn	Funktion	Relationer	Eksempel
Equal	$X.Relate(Y, 'TFFFTFFFT')$	79	
Near(distance d)	$X.Relate(Y, 'FF * FF ****')$ $\wedge Distance(X, Y) < d$	79	
Disjoint	$X.Relate(Y, 'FF * FF ****')$	79	

Tabel 15-4 Funktioner for relationer mellem to punkter.



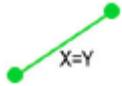


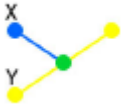
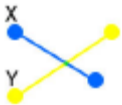

Punkt-linie			
Navn	Funktion	Relationer	Eksempel
Touches	$X.Relate(Y, '*0**0*****')$	<b>151</b>	
Within	$X.Relate(Y, '0**0*****')$	<b>295</b>	
Near(distance d)	$X.Relate(Y, 'FF * FF *****')$ $\wedge Distance(X, Y) < d$	<b>79</b>	
Disjoint	$X.Relate(Y, 'FF * FF *****')$	<b>79</b>	


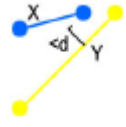


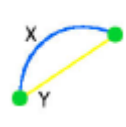


Tabel 15-5 Funktioner for relationer mellem et punkt og en linie.

Punkt-polygon			
Navn	Funktion	Relationer	Eksempel
Touches	$X.Relate(Y, '*0**0****')$	<b>151</b>	
Within	$X.Relate(Y, '0**0****')$	<b>295</b>	
Near(distance d)	$X.Relate(Y, 'FF * FF ****')$ $\wedge Distance(X, Y) < d$	<b>79</b>	
Disjoint	$X.Relate(Y, 'FF * FF ****')$	<b>79</b>	

Tabel 15-6 Funktioner for relationer mellem et punkt og en polygon.

Når de to objekter, der indgår i en relation begge er linier, er det nødvendigt at benytte en anden tilgang, end den, der hidtil er blevet brugt. Her deles funktionerne i to grupper, de normale, baseret på *Relate*, samt dimensionen af fællesgeometrien, der findes vha. *Intersect*. To objektklassers topologiske relation defineres da som en kombination af de to grupper. F.eks. kan det angives, at to objekter kun må have relationen møde hinanden, og have en fællesgeometri, der er et punkt. I dette tilfælde er den eneste tilladte relation 95, der er kan ses i Appendix 1. Det er altså meningen, at det angives, hvilke af de nedenstående funktioner, der er tilladte, og hvilke der ikke er tilladte, og derved er kombinationerne af funktionerne på samme måde enten tilladte eller ej.

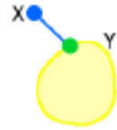
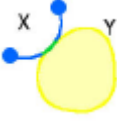
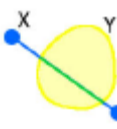
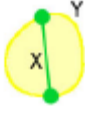
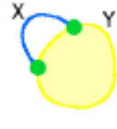
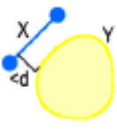
Linie-linie			
Navn	Funktion	Relationer	Eksempel
Equal	$X.Relate(Y, 'TFFF TFFFT')$	273	
Meets	$X.Relate(Y, '**** T ****')$	85, 95, 119, 221, 245, 273, 311, 341, 351, 367, 375, 463, 473, 477, 501	
Touches	$X.Relate(Y, '*** T ****')$	103, 111, 119, 229, 231, 237, 239, 245, 295, 311, 359, 367, 375, 485, 487, 493, 495, 501	
IsTouched	$Y.Relate(X, '* T ****')$	205, 207, 221, 229, 231, 237, 239, 245, 457, 461, 463, 473, 477, 485, 487, 493, 495, 501	
Crosses	$X.Relate(Y, 'T ****')$	273, 295, 311, 335, 341, 351, 359, 367, 375, 457, 461, 463, 473, 477, 485, 487, 493, 495, 501	
Contains	$X.Relate(Y, 'T * F ** F **')$	273, 295, 311	

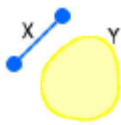
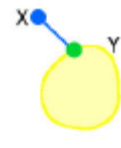
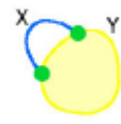

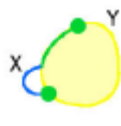
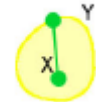
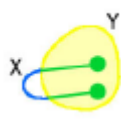
Navn	Funktion	Relationer	Eksempel
IsContained	$Y.Relate(X, 'T ***** FF*')$	273, 457, 473	
Near(distance d)	$Y.Relate(X, 'FF * FF *****')$ $\wedge Distance(X, Y) < d$	79	
Disjoint	$Y.Relate(X, 'FF * FF *****')$	79	
ComGeom0D (maxIsects I)	$GeomCount(X.Intersect(Y))$ $= I \wedge Max(Dim(X.Intersect(Y)))$ $= 0$	I = 1: 95, 111, 207, 335	
		I > 1: 85, 103, 119, 205, 221, 229, 231, 237, 239, 245, 341, 335, 351, 359, 367, 375, 461, 463, 477, 485, 487, 493, 495, 501	
ComGeom1D (maxIsects I)	$GeomCount(X.Intersect(Y))$ $= I \wedge Max(Dim(X.Intersect(Y)))$ $= 1$	I = 1: 273, 295, 311, 335, 351, 367, 457, 463, 473, 495	
		I > 1: 335, 341, 351, 359, 367, 375, 461, 463, 477, 485, 487, 493, 495, 501	

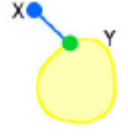
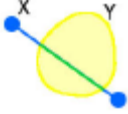
Tabel 15-7 Funktioner for relationer mellem to linier.

De sidste to funktioner  $\text{ComGeom0D}(\text{maxIsects } I)$  og  $\text{ComGeom1D}(\text{maxIsects } I)$ , bestemmer, om der er fællesgeometri hhv. som punkt og linie, og det angives hvor mange gange de to linier må skære hinanden vha. værdien  $\text{maxIsects}$ .

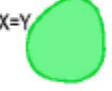

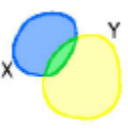
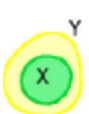
På samme måde som før defineres de tilladte relationer, med to grupper af funktioner. Denne gang differentieres der dog imellem fællesgeometri med polygonens kant, indre og ydre.


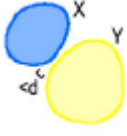
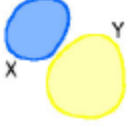
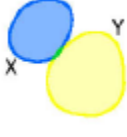
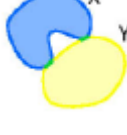


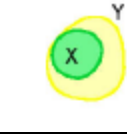
Linie-polygon			
Navn	Funktion	Relationer	Eksempel
Meets	$X.Relate(Y, '****T****')$	87, 95, 151, 215, 223, 279, 311, 407, 439, 471, 479, 503	
Touches	$X.Relate(Y, '*T*****')$	151, 207, 215, 223, 407, 423, 439, 463, 471, 479, 487, 495, 503	
CrossesEdge	$X.Relate(Y, 'TTT*****')$	463, 471, 479, 487, 495, 503	
IsInside	$X.Relate(Y, 'T*F*****')$	279, 295, 311, 407, 423, 439	
IsOutside	$X.Relate(Y, 'F*T*****')$	79, 87, 95, 207, 215, 223	
Near(distance d)	$Y.Relate(X, 'FF*FF*****')$ $\wedge \text{Distance}(X, Y) < d$	79	

Navn	Funktion	Relationer	Eksempel
Disjoint	$Y.Relate(X, 'FF * FF ****')$	79	
EdgeComGeom0D (maxIsects I)	$GeomCount(X.Intersect(Y.Edge))$ $= I \wedge$ $Max(Dim(X.Intersect(Y.Edge)))$ $= 0$	I = 1: 95, 207, 311, 423, 495	
		I > 1: 87, 207, 215, 223, 279, 407, 423, 439, 463, 471, 479, 487, 503	
EdgeComGeom1D (maxIsects I)	$GeomCount(X.Intersect(Y.Edge))$ $= I \wedge$ $Max(Dim(X.Intersect(Y.Edge))) = 1$	I = 1: 151, 207, 223, 423, 439, 495	
		I > 1: 207, 215, 223, 407, 423, 439, 463, 471, 479, 487, 495, 503	
InComGeom1D (maxIsects I)	$GeomCount(X.Intersect(Y.Edge))$ $= I \wedge$ $Max(Dim(X.Intersect(Y.Edge))) = 1$	I = 1: 279, 295, 311, 407, 439, 463, 471, 479, 495, 503	
		I > 1: 407, 423, 439, 463, 471, 479, 487, 495, 503	

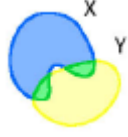
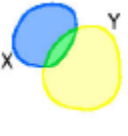
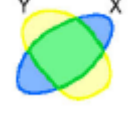
Navn	Funktion	Relationer	Eksempel
OutComGeom1D (maxIsects I)	$GeomCount(X.Intersect(Y.Outside)) = I \wedge Max(Dim(X.Intersect(Y.Outside))) = I$	<b>I = 1:</b> 79, 87, 95, 215, 223, 471, 479, 487, 495, 503	
		<b>I &gt; 1:</b> 207, 215, 223, 463, 471, 479, 487, 495, 503	

Tabel 15-8 Funktioner for relationer mellem en linie og polygon.

Polygon-polygon			
Navn	Funktion	Relationer	Eksempel
Equal	$X.Relate(Y, 'T***T***T')$	<b>79</b>	
Touches	$X.Relate(Y, 'F***T***')$	<b>95</b>	
Overlaps	$X.Relate(Y, 'TTTTTTTT')$	<b>511</b>	
Inside	$X.Relate(Y, 'T*F**F***')$	<b>273, 295, 439</b>	

Navn	Funktion	Relationer	Eksempel
Contains	$Y.Relate(X, 'T*****FF*')$	273, 457, 505	
Near(distance d)	$Y.Relate(X, 'FF*FF*****')$ $\wedge Distance(X, Y) < d$	79	
Disjoint	$Y.Relate(X, 'FF*FF*****')$	79	
EdgeComGeom0D (maxIsects I)	$GeomCount(X.Intersect(Y.Edge))$ $= I \wedge$ $Max(Dim(X.Intersect(Y.Edge))) = 0$	I = 1: 95, 439, 505	
		I > 1: 95, 439, 505	
EdgeComGeom1D (maxIsects I)	$GeomCount(X.Intersect(Y.Edge))$ $= I \wedge$ $Max(Dim(X.Intersect(Y.Edge))) = 1$	I = 1: 95, 439, 505, 511	
		I > 1: 95, 439, 505, 511	
InComGeom2D (maxIsects I)	$GeomCount(X.Intersect(Y.Edge))$ $= I \wedge$ $Max(Dim(X.Intersect(Y.Edge))) = 2$	I = 1: 273, 295, 439, 457, 505, 511	



		<b>I &gt; 1: 511</b>	
OutComGeom2D (maxIsects I)	$GeomCount(X.Intersect(Y.Outside)) = I \wedge Max(Dim(X.Intersect(Y.Outside))) = 2$	<b>I = 1: 79, 95, 511</b>	
		<b>I &gt; 1: 511</b>	

Tabel 15-9 Funktioner for relationer mellem to polygoner.

For to polygoner, ses det, at alle relationer kan realiseres ved at undersøge skæringen mellem de to objekter, og da DU9SM-modellen netop gør dette, kan denne med fordel benyttes.

Der findes også relationer mellem andre objektpar, såsom polygon-punkt, men de funktioner, der skal bruges til at undersøge relationer, svarer til dem, der benyttes til at undersøge relationer mellem det modsatte objektpar, punkt-polygon. Den skæringsmatrice der benyttes, skal blot transponeres.

Når der skal defineres en topologisk relation mellem to objekter, er det hensigten, at en af de funktioner, der er beskrevet ovenfor, vælges. Herefter er det muligt at definere de tilladte skæringer, mellem de forskellige dele af objekterne; det indre, kanten og det ydre, i en matrice. På den måde fungerer funktionen som skabelon for 9-skæringmatricen, der herefter kan tilpasses til den relevante relation.

### 15.3.3 Regler i Top10dk

For at illustrere, hvordan de topologiske funktioner kan bruges, beskrives de topologiske regler fra Top10dk, og det gennemgås, hvordan disse kan defineres vha. både funktionen *Relate*, fra OpenGIS, og vha. de alternative funktioner, der er beskrevet ovenfor.

For en mere udførlig beskrivelse af de termer, der benyttes her, fra Top10dk, samt hvilke topologiske regler der forekommer, henvises

der til specifikationen for Top10dk [Kort & Matrikelstyrelsen, 2001]]. Herunder beskrives de enkelte topologiske regler i Top10dk. Disse topologiske regler defineres to forskellige steder i specifikationen til Top10dk. Det ene sted er i topologiskemaerne, der er oversigter over, hvordan et objekt fra en objektklasse relaterer til alle andre. Det andet sted er i afsnittet om datamodellen.

### 15.3.3.1 Snap

Hvis et objekt  $a$  kan have fælles punkt med et objekt  $b$ , altså at de to objekter kan krydse eller røre hinanden i netop ét punkt, står der 3d/2d i topologiskemaet [Kort & Matrikelstyrelsen, 2001]. Det kan bestemmes om to objekter har denne relation vha. funktionerne, der er vist i tabel 15-10 og 15-11.

Kriterier	Funktion
Hvis $a$ og $b$ begge er punkter:	$a.Relate(b, '0***0****')$
Hvis $a$ er et punkt og $b$ er en linie:	$a.Relate(b, '0*0****') \vee a.Relate(b, '*0*0****')$
Hvis $a$ og $b$ begge er linier:	$a.Relate(b, '0F*FF****') \vee a.Relate(b, 'F0*FF****') \vee a.Relate(b, 'FF*0F****') \vee a.Relate(b, 'FF*F0****')$
Hvis $a$ er en linie og $b$ er en polygon:	$a.Relate(b, 'F0*FF****') \vee a.Relate(b, 'FF*F0****')$
Hvis $a$ og $b$ begge er polygoner:	$a.Relate(b, 'FF*F0****')$

Tabel 15-10 Metoder til at bestemme relationen "Snap" vha. skæringsmatricer.

I stedet for det overstående, kan relationen også bestemmes ved:

$$Dim(a.Intersect(b)) = 0$$

Dvs. at den geometri, der er fælles for de to objekter skal være et punkt (0-dimensional).

Hvis  $a$  er et punkt og  $b$  en polygon, vil det ovenstående ikke gælde, idet skæringen mellem  $a$  og  $b$  også vil være et punkt, hvis  $a$  ligger inden i  $b$ :

$$a.Relate(b, '0**0****') \vee a.Relate(b, '*0**0****')$$

Hvis de alternative funktioner benyttes, fås:

Kriterier	Funktion
Hvis $a$ og $b$ begge er punkter:	$a.Equal(b)$
Hvis $a$ er et punkt og $b$ er en linie:	$a.Touches(b) \vee a.Within(b)$
Hvis $a$ og $b$ begge er linier:	$(a.Meets(b) \vee a.Touches(b) \vee a.IsTouched(b) \vee a.Crosses(b)) \wedge a.CommonGeom0D(b,1)$
Hvis $a$ er en linie og $b$ er en polygon:	$(a.Meets(b) \vee a.Touches(b) \vee a.IsTouched(b) \vee a.Crosses(b)) \wedge a.CommonGeom0D(b,1)$
Hvis $a$ og $b$ begge er polygoner:	$a.Touches(b) \wedge a.EdgeCommonGeom0D(b,1)$

Tabel 15-11 Metoder til at bestemme relationen "Snap" vha. alternative funktioner.

Hvis de to objekter rører hinanden i et punkt, og denne relation er tilladt, skal de begge have et fællespunkt i skæringspunktet. Hvis  $z$ -koordinaterne i skæringspunktet er ens for de to objekter, skal det være et 3d fællespunkt, mens der skal være 2d-fællespunkt, hvis de har forskellige  $z$ -koordinater.

Hvis relationen ikke er tilladt, er der opstået en fejl, der skal rettes på en eller anden måde. En bygning kan f.eks. ikke snappe med vejmidte i niveau, men kan godt snappe med vejmidte i 2d.

### 15.3.3.2 Bryder

Det er kun objekter, med geometritypen linie, der kan brydes af andre objekter. Hvis et objekt  $a$  har relationen bryder med et andet objekt  $b$ , betyder det at  $b$  skal have endepunkt i fællespunktet, og dermed at

objektet skal deles op, hvis det ikke har endepunkt i fællespunktet. Det vil sige, linien skal deles op, hvis de to objekter snapper til hinanden,  $a$  bryder  $b$ , og relationen er:

$$a.Relate(b, '0F * FF ****') \vee a.Relate(b, 'FF * 0F ****')$$

Dette kan også bestemmes vha. de alternative funktioner:

$$(a.Touches(b) \vee a.Crosses(b)) \wedge a.CommonGeom0D(b,1)$$

Hvis det først undersøges hvilke objekter, der snapper til hinanden, vil de allerede have fællespunkt, dér hvor de krydser eller rører hinanden. Derfor er det kun nødvendigt at undersøge, om der, i nogle af disse fællespunkter, skal foretages en opdeling af linierne. To objekter fra klassen Vejmidte bryder f.eks. hinanden i 3d, men ikke i 2d, således at Vejmidte-objekterne udgør et netværk.

### 15.3.3.3 Fællesgeometri

Fællesgeometri afgør om det er tilladt for to objekter, at have geometrisammenfald. For objekter, med punkter som geometri, er det det samme som snap, og det er derfor ikke aktuelt at undersøge punktobjekter. For objekter, der har linier som geometri, vil fællesgeometri sige, at det er tilladt for to objekter, at have et eller flere liniestykker, der er sammenfaldende. For objekter, med polygon-geometri, er det dele af polygonens grænser, der kan være sammenfaldende. For at teste for denne relation skal disse tests benyttes:

Kriterier	Funktion
Hvis $a$ og $b$ begge er linier:	$a.Relate(b, '1 * * * * * * *')$
Hvis $a$ er en linie og $b$ er en polygon:	$a.Relate(b, '*1 * * * * * *')$
Hvis $a$ og $b$ begge er polygoner:	$a.Relate(b, '* * * * 1 * * * *')$

Tabel 15-12 Metoder til at bestemme relationen "Fællesgeometri" vha. skæringsmatricer.

Dette kan også beskrives ved:

$$dim(a.Intersect(b)) = 1$$

Med de alternative funktioner, benyttes følgende:

Kriterier	Funktion
Hvis $a$ og $b$ begge er linier:	$a.Crosses(b) \wedge a.CommonGeomID(b,1)$
Hvis $a$ er en linie og $b$ er en polygon:	$a.Touches(b) \wedge a.IsOutside(b) \wedge a.EdgeCommonGeomID(b,1)$
Hvis $a$ og $b$ begge er polygoner:	$a.Touches(b) \wedge a.EdgeCommonGeomID(b,1)$

Tabel 15-13 Metoder til at bestemme relationen "Fællesgeometri" vha. alternative funktioner.

Derudover skal det undersøges, om der er tale om 3d eller 2d fællesgeometri, og om det er tilladt. Et objekt fra klassen Bygning kan ikke have fællesgeometri med et objekt fra klassen Vejmidte, men den kan godt have 2d-fællesgeometri med et andet objekt fra klassen Bygning.

#### 15.3.3.4 Kan ligge inden i

Relationen Kan ligge inden i afgør om et objekt  $b$  kan ligge inden et andet objekt  $a$ , der har geometritypen polygon. Relationen betyder, at objektet skal ligge inden i polygonen uden at røre kanten, hvis det andet objekt,  $b$ , har punkt eller polygon-geometri. Hvis  $b$  er et objekt med liniegeometri kan det ligge både inden i og udenfor, med mindre  $a$  bryder  $b$ . En metode til at undersøge, om der er denne relation mellem  $a$  og  $b$ , kan findes ved at benytte følgende:

Kriterier	Funktion
Hvis $a$ er en polygon og $b$ er et punkt eller en polygon:	$b.Relate(a, 'TFFTF * * *')$
Hvis $a$ er en polygon og $b$ er en linie:	$b.Relate(a, 'T * * * * * * *')$

Tabel 14-14 Metoder til at bestemme relationen "Kan ligge inden i" vha. skæringsmatricer.

Hvis de alternative funktioner anvendes, benyttes følgende:

Kriterier	Funktion
Hvis $a$ er en polygon og $b$ er et punkt:	$a.Within(b)$
Hvis $a$ er en polygon og $b$ er en linie:	$!a.IsOutside(b)$
Hvis $a$ er en polygon og $b$ er en polygon:	$a.Inside(b)$

Tabel 15-15 Metoder til at bestemme relationen "Kan ligge inden i" vha. alternative funktioner.

Det er muligt, at der skal foretages *cut out* hvis et objekt  $b$  ligger inden i et objekt  $a$ . Det vil sige, at der skal være hul i  $a$ 's geometri, dér hvor  $b$  ligger. Et eksempel på dette er at der skal være *Cut out* hvis et objekt, fra klassen Hede ligger inde i et objekt, fra klassen Skov.

#### 15.3.3.5 Identiske objekter må ikke forekomme

For alle objektklasser gælder det, at to objekter fra samme objektklasse, som har samme form og placering ikke må forekomme. Det kan undersøges om de er ens vha. relationen:

$a.Relate(b, 'TFFFTFFFT')$

Dette kan også udtrykkes vha. de alternative funktioner:

$a.Equal(b)$

#### 15.3.3.6 To arealobjekter må ikke overlappe

Det er ikke tilladt at et objekt, der har en polygon som geometri, ligger både indenfor og udenfor et andet tilsvarende objekt, eller at det ligger inden i og på kanten af et objekt. For eksempel må et objekt fra klassen Skov ikke overlappe et objekt fra klassen Hede. Hvis det er tilfældet, skal grænsen følge det objekt, der er defineret bedst. Relationen for dette er:

$a.Relate(b, 'TT * TT * TTT')$

De alternative funktioner udtrykker dette ved:

*a.Overlaps(b)*

Der kan eventuelt udføres en klassificering af, hvilke objekter der er bedst defineret, og dermed kan denne slags fejl automatisk rettes.

## 15.3.3.7 Minimum en meter mellem to objekter

Hvis to objekter er mindre end en meter fra hinanden, skal de have sammenfaldende geometri, hvis det er tilladt. Fællesgeometrien skal igen følge det objekt, der er bedst defineret. Her kan følgende benyttes, til at finde objekter, der har en afstand mindre end 1 meter:

*a.Distance(b) < 1*

eller, udtrykt vha. de alternative funktioner:

*a.Near(b,1)*

Denne regel gælder for alle objekter, både i planen og i koten.

De netop beskrevne regler, er de topologiske regler, der indgår i top10dk. Disse regler kan implementeres med begge metoder. I det følgende beskrives et eksempeldatasæt, der minder om Top10dk, og derefter forsøges det at indbygge en række topologiske regler for dette datasæt i ArcGIS.

## 15.4 Implementering

For at formalisere de topologiske regler, er det nødvendigt med et værktøj, hvori det er muligt at definere de topologiske regler fra specifikationen. Ved hjælp af et sådan værktøj kan topologireglerne genereres for de objekter, der indgår i en given datamodel. Opbygningen af et sådan værktøj illustreres i det følgende. Som eksempel benyttes en simpel datamodel, kaldet Testdata, der indeholder objektklasserne Bygning og Matrikel, der har geometrien polygon, Vej, der har geometrien linie, samt Standsningssted, der har geometrien punkt. Denne model har en del fællestræk med Top10dk.

Objektklasserne skal overholde følgende topologiske regler:

Objektpar	Regler
Bygning-Bygning	<p><b>Bygninger må røre hinanden i ét punkt. Ud over det, er der ikke andre tilladte relationer.</b></p> <p><b>To bygninger må ikke være identiske.</b></p>
Bygning-Matrikel	<p><b>Bygning skal ligge inden i en matrikel.</b></p> <p><b>Bygning skal være indeholdt i matriklen.</b></p> <p><b>Bygningen må gerne røre kanten af matriklen både i et punkt, i en linie eller en kombination.</b></p> <p><b>Bygningen må ikke skære kanten af matriklen.</b></p> <p><b>Bygningen må gerne fylde hele matriklen.</b></p>
Bygning-Vej	<p><b>En bygning må ikke røre en vej.</b></p>
Bygning-Standsningssted	<p><b>En bygning må ikke røre eller indeholde et standsningssted.</b></p>
Matrikel-Matrikel	<p><b>Matrikler må ikke overlappe.</b></p> <p><b>Matrikler må gerne røre hinanden.</b></p> <p><b>To matrikler må ikke være identiske.</b></p> <p><b>Må ikke have huller.</b></p>
Matrikel-Vej	<p><b>Matrikler må gerne indeholde veje.</b></p> <p><b>Matrikler må gerne skære veje.</b></p> <p><b>Veje må gerne ligge på kanten af matrikler.</b></p>
Matrikel-Standsningssted	<p><b>Matrikel må gerne indeholde standsningssted.</b></p> <p><b>Standsningssted må gerne ligge på kanten af matrikel.</b></p>
Vej-Vej	<p><b>Vej må gerne krydse en anden vej i et punkt, hvis der i punktet er en koteforskel på 3 meter eller mere.</b></p> <p><b>Vej skal møde en anden vej.</b></p> <p><b>Vej må ikke røre en anden vej.</b></p> <p><b>To veje må ikke være identiske.</b></p>



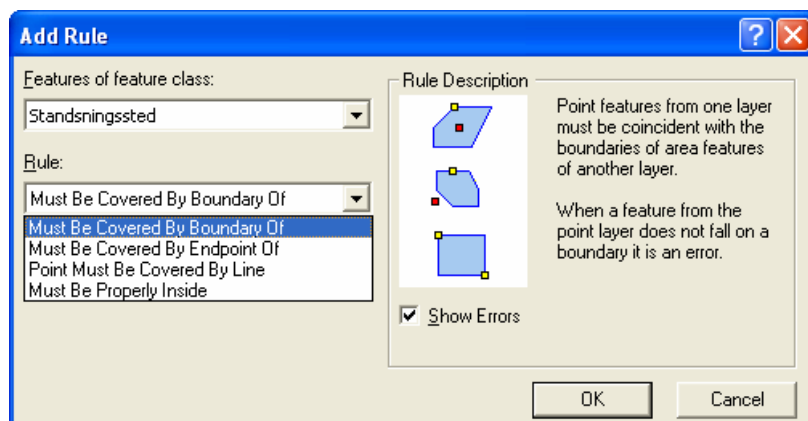
	<b>To veje må ikke have fællesgeometrien linie.</b>
Vej-Standsningssted	<b>Et standsningssted skal ligge på en vej, enten i enden eller på kanten.</b>
Standsningssted-Standsningssted	<b>To standsningssteder må ikke røre hinanden.</b>

Tabel 15-16 Topologiskema for Testdata

### 15.4.1 Topologi i ArcGIS

I det følgende gives en række eksempler på, hvordan nogle af de topologiske regler fra eksemplet ovenfor kan implementeres simpelt i ArcGIS. I ArcGIS gemmes disse regler sammen med data i databasen.

Den metode, der bruges til at definere topologiske regler i ArcGIS, er beskrevet i Appendiks 4. På figur 15-2 ses et eksempel på, hvordan en topologisk regel for klassen standsningssted defineres.



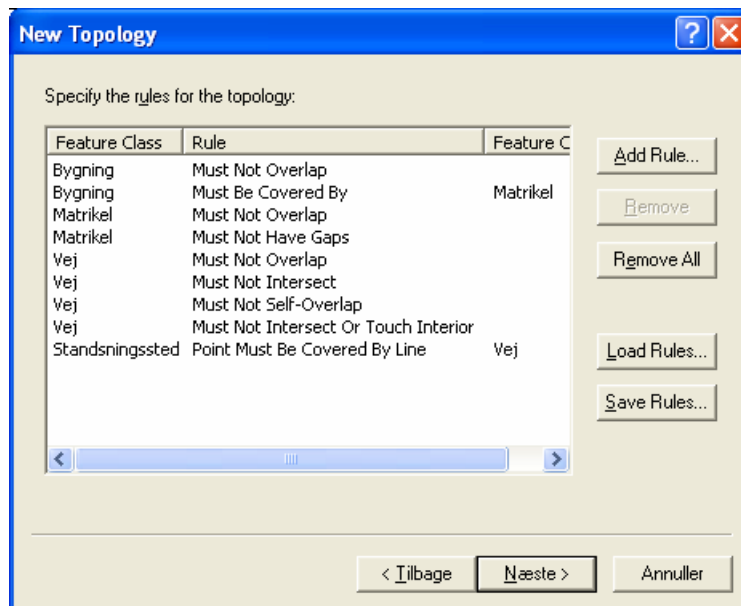
Figur 15-2 De mulige topologiske regler for klassen Standsningssted.

Figur 15.3 viser en liste over de topologiske regler i det aktuelle datasæt. Hvis der sammenlignes med topologiskemaet for datasættet fra sidste afsnit, er der en del regler, der ikke kan tjekkes:

- To objekter fra klassen Bygning må kun røre hinanden i netop et punkt. I ArcGIS er det kun muligt at definere, om en classes objekter må overlappe hinanden eller ej. Dvs. at der ikke vil blive registreret en fejl, hvis to bygninger har en fælles linie.

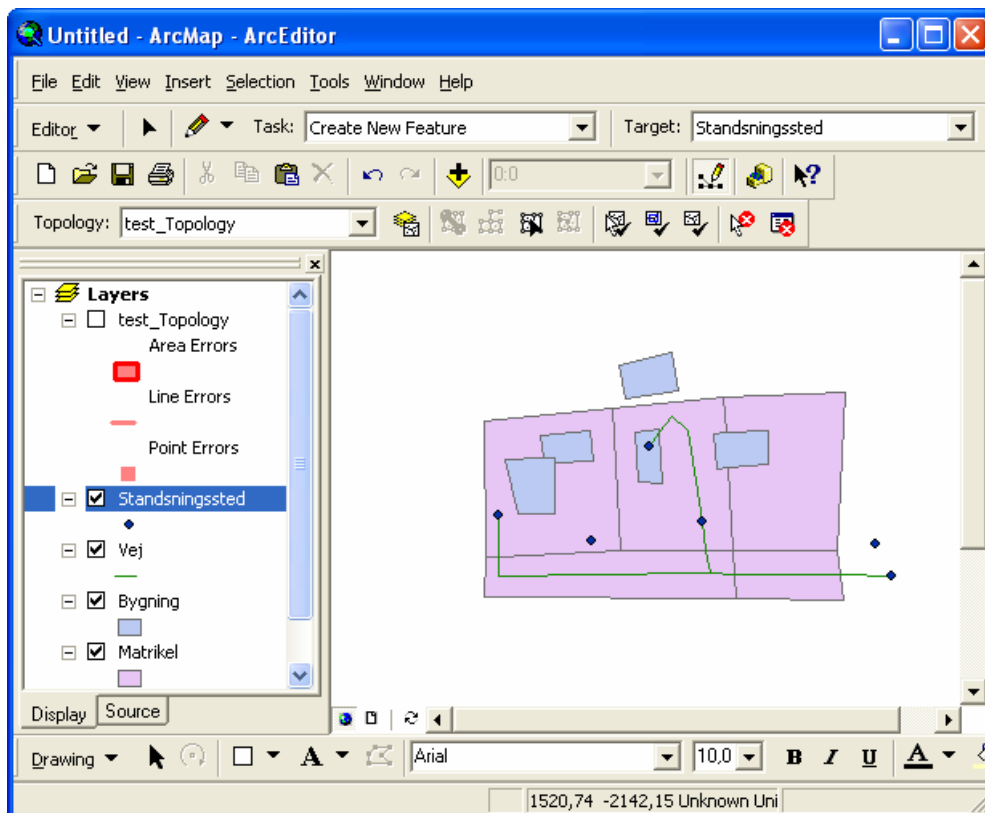
- En bygning må ikke røre en vej. Det er ikke muligt at oprette en regel, der undersøger om linier ligger inden i eller rører polygoner. Derfor bliver denne regel ikke undersøgt.
- En bygning må ikke røre eller indeholde et standsningssted. ArcGIS giver mulighed for at lave en regel om at et punkt skal ligge inden i en polygon, men det er ikke muligt at lave en regel, der forhindrer, at et punkt ligger inden i en polygon. Derfor kan denne regel heller ikke oprettes.
- To standsningssteder må ikke røre hinanden. Det er ikke muligt lave topologiske regler mellem to punkter.
- Vej må gerne krydse en anden vej i et punkt, hvis der i punktet er en koteaforskul på 3 meter eller mere. Det er ikke muligt at benytte andre af objekternes attributter end geometrien i de topologiske regler.

De resterende regler fra tabel 15-16 er mulige at realisere, vha. af de indbyggede regler i ArcGIS. Der er især mange muligheder for regler for objekter, med geometrien polygon, f.eks. at der ikke må være huller imellem objekterne og at én klasses polygoner, skal udfylde en anden klasses polygoner. De oprettede regler kan ses på figur 15-3.



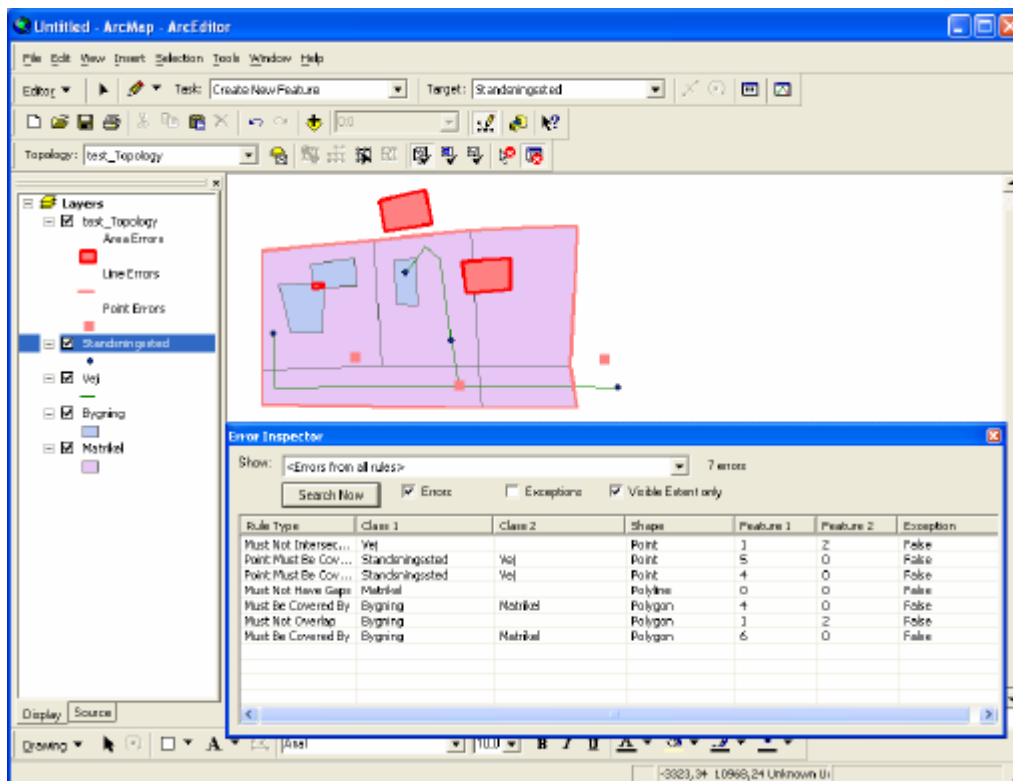
Figur 15-3 De topologiske regler, der er tilføjet til datasættet.

Efter at topologien er opbygget, genereres der testdata, som ses på figur 15-5. Der er indlagt forskellige fejl, for at undersøge hvilke der bliver fundet, og hvilke der overses, samt hvordan det er muligt at identificere disse.



Figur 15-4 Eksempeldatasæt.

Derved tjekkes de valgte objekter, vha. af de regler, der er defineret tidligere. De steder, hvor der er fejl, markeres, og det er muligt at se en liste over samtlige fejl vha. en såkaldt "error inspector", hvilket er illustreret på figur 15-5. Ved at udarbejde en liste over fejlene, kan der opnås et overblik over disse, og det er muligt at finde og rette de forskellige fejl på en enkelt måde.



Figur 15-5 Fejlene i datasættet er markeret med rød, og vises på listen i Error Inspector.

Værktøjet i ArcGIS giver en del muligheder for at arbejde med topologi, men der er stadig en del ting, der savnes. Det er ikke muligt at lave "hjemmelavede" regler vha. af det topologiske værktøj, og der mangler en del af de topologiske relationer, der ofte ses. Det er f.eks. ikke muligt at undersøge, om en linie skærer en polygon.

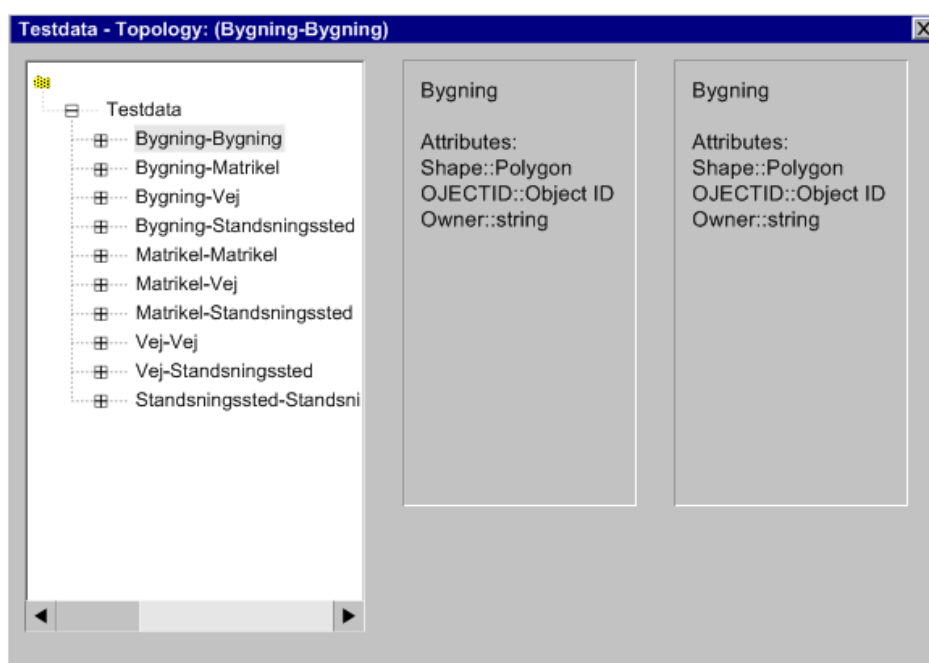
I det følgende beskrives en alternativ metode til at oprette topologiske regler for et datasæt.

#### 15.4.2 Topologiværktøj

De topologiske regler for datasættet skal defineres vha. et værktøj, der f.eks. kan udvikles som et plug-in til ArcGIS, eller endnu bedre, i et selvstændigt program, der kan bygge på COM-teknologi, der er beskrevet i Appendiks 3. Værktøjet kan da læse datasættets struktur fra en struktureret database, som f.eks. Microsoft Repository, og derefter er det muligt at definere de topologiske regler, implementere dem i et COM-objekt, og tilknytte COM-objektet til data. I det følgende illustreres det, hvordan processen med at definere reglerne

kan foregå, samt hvordan det omtalte værktøj kan se ud. Dette er et tænkt eksempel, og er ikke noget, der er implementeret, men det illustrerer hvilke muligheder, der er for at oprette topologiske regler. De viste skærbilleder konstrueret vha. Microsoft Visio, og er altså fiktive.

Når en bruger vælger at anvende dette værktøj til at oprette topologiske regler i sin database, bliver der genereret en liste over alle de mulige objektpar, der er i databasen. Herefter åbnes en dialog, som illustreret på figur 15-6, hvor det er muligt at definere de topologiske regler for hvert enkelt objektpar. I dialogen præsenteres en træstruktur, der viser datasættet og de forskellige objektpar. Hvis et af objektparrene vælges, kan de to objektklassers attributter ses.

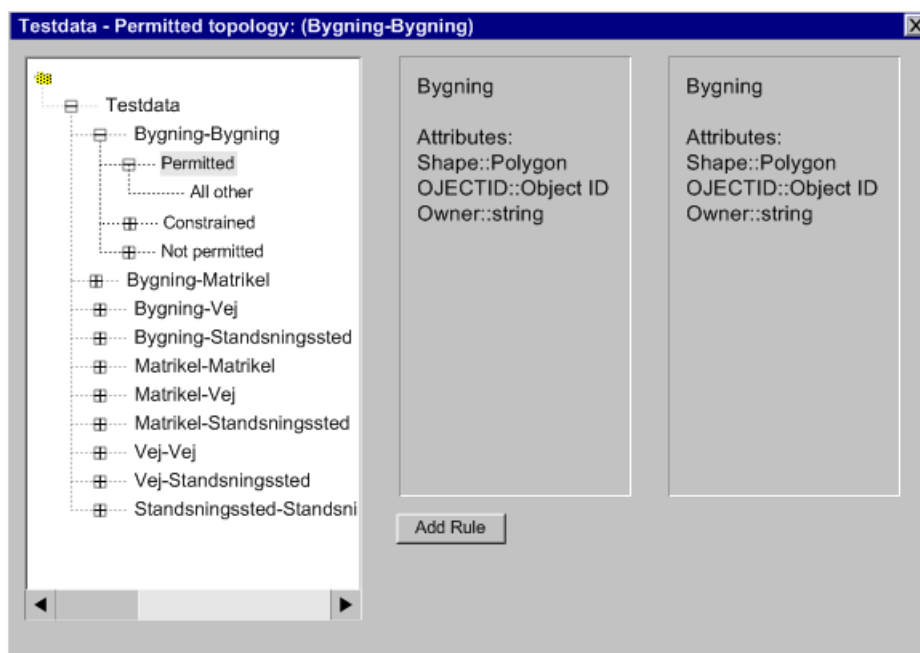


Figur 15-6 Værktøjet til at oprette topologiske regler.

Hvis et af objektparrene i træstrukturen åbnes, vises de tre mapper, *permitted* (tilladt), *constrained* (begrænset) og *not permitted* (ikke tilladt), hvilket er illustreret på figur 15-7. Disse mapper indeholder alle de mulige relationer, der kan være mellem de to objekter. Fra starten ligger relationen *All other* med skæringsmatricen '\*\*\*\*\*' i *permitted*. Denne relation betegner de resterende relationer. Alle relationer er altså tilladte fra starten, og er det så længe, der ikke foretages ændringer i mapperne, dvs. tilføjet eller flyttet regler. Hvis der er defineret en regel i specifikationen om, at det er tilladt for to

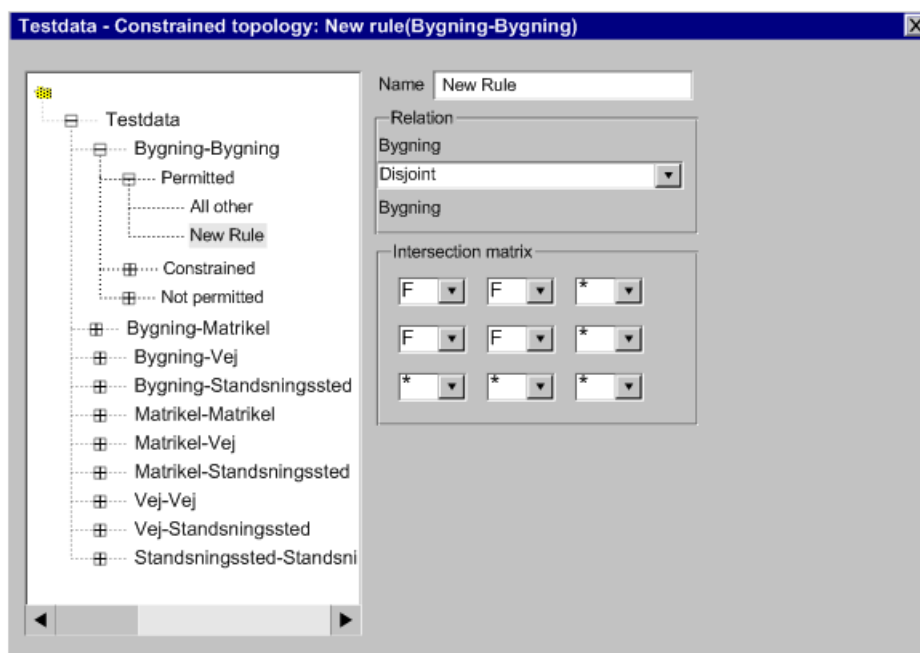
objekter at røre hinanden, og de resterende relationer ikke er tilladte, så kan relationen *All other* flyttes over i *not permitted*, og relationen *touch* oprettes i *permitted*.

Det er altså muligt at markere en af folderne f.eks. *permitted*, og her vælge *add rule* (tilføj regel). Trykkes der her, kan denne relation defineres. Skærmen, hvor dette sker, er vist på figur 15-8.



Figur 15-7 I folderen *Permitted* er det muligt at tilføje regler.

For relationen bygning-bygning er de eneste relationer, der er tilladte, *disjoint* og *touch* i et punkt. Derfor flyttes alle relationer fra *permitted* til *not permitted*. Herefter tilføjes relationen *disjoint* til *permitted*, som vist på figur 15-8.



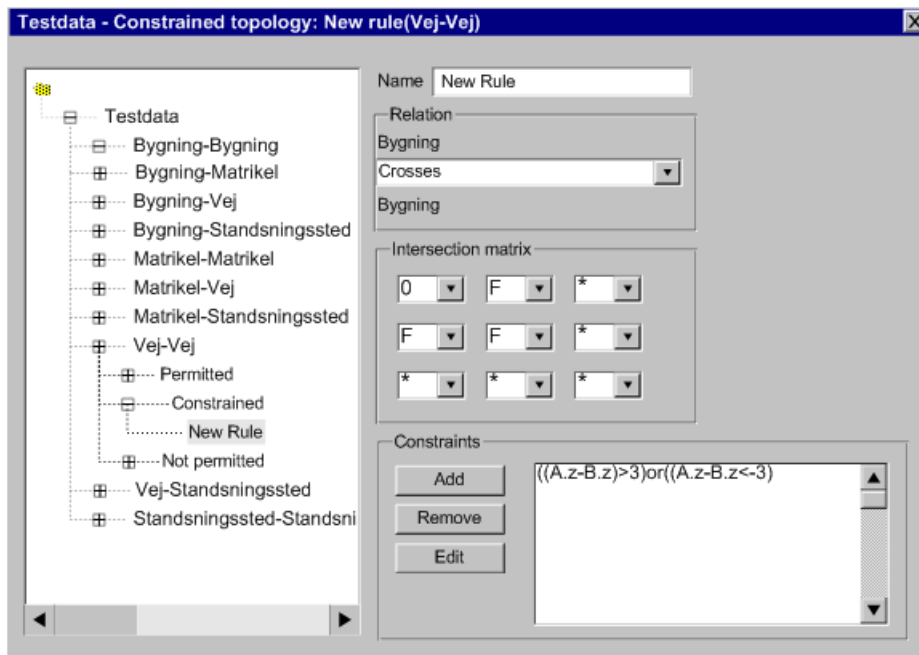
Figur 15-8 En ny regel defineres i forderen *permitted*.

Som det ses på figuren, er det muligt at vælge en af de alternative funktioner, der er beskrevet tidligere, fra drop-down menuen for oven. Denne funktion vises herefter som en skæringsmatrice nedenunder. Her er det muligt at modificere relationen, så den svarer præcist til det ønskede. For eksempel er det muligt at vælge hvilken type fællesgeometri, der er tilladt. Det er kun de mulige fællesgeometrier der vises, eksempelvis kan fællesgeometrien mellem kanten af to linier kun kan være 0 eller -1. De funktioner der kan vælges, er dem, der er gennemgået tidligere i dette kapitel. Programmet undersøger, hvilken type geometri de to objekter har, og angiver derudfra, de funktioner, der kan vælges i drop-down menuen. I matricen er det derefter muligt at definere hvilken type fællesgeometri, de to objekter kan have. Der kan evt. vises et eksempel på topologireglen til højre for dette, ligesom i ArcGIS. Relationen *touch* defineres på samme måde som relationen *disjoint*.

Ved hjælp af værktøjet er det også muligt at definere funktioner, der afhænger af andre parametre. Det ses af reglerne fra topologiskemaet, at to veje må krydse hinanden, hvis koteforskellen er større end 3 meter. Denne regel defineres i *constrained*-mappen, som illustreret på figur 15-9. Her er det muligt at definere regler, der netop også tager hensyn til andre attributter. Nederst i dialogboksen ses de ting, der



skal være opfyldt, for at relationen er tilladt. Disse defineres vha. en *query builder*, der kan se ud som illustreret på figur 15-10. Denne fungerer stort set ligesom i ArcGIS, hvor det dog ikke er muligt at benytte denne i forbindelse med topologiske regler.



Figur 15-9 Det er muligt at definere regler, der afhænger af andre attributter end de rumlige.

Herved er det muligt at definere samtlige topologiske relationer mellem de forskellige klasser i datasættet.



Figur 15-10 Query builder.

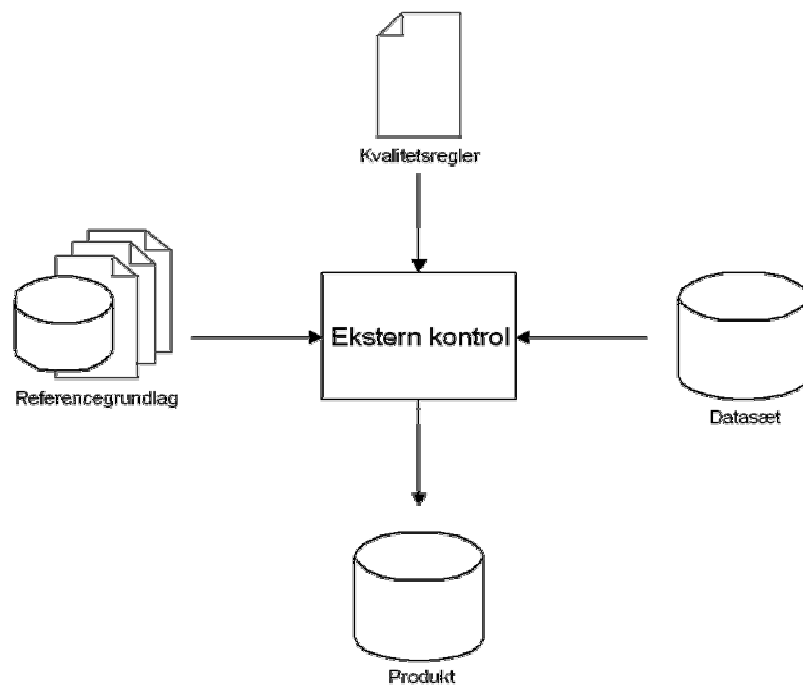
Dette værktøj kan altså bruges til, på en effektiv og intuitiv måde, at specificere en række regler, der gør, at datasættet overholder de topologiske regler, der er beskrevet i specifikationen. På den måde er det muligt at reducere antallet af topologiske fejl, og samtidig lette arbejdet for operatøren.

Når at hele datasættet er undersøgt i den interne kontrol, sendes det videre til den eksterne kontrol, hvor en række af kvalitetselementerne bestemmes og kontrolleres i forhold til de opstillede krav.

## 16 Ekstern kontrol

Den eksterne kontrol benyttes til at kontrollere, at kvalitetselementerne for det producerede datasæt overholder de krav, der stilles i specifikationen.

På figur 16-1 er det illustreret, hvordan den eksterne kontrol modtager det producerede datasæt, efter at det er kontrolleret for bl.a. topologiske fejl i den interne kontrol. Ud over dette datasæt, benytter den eksterne kontrol også et referencegrundlag, der indeholder en eller flere datakilder samt de krav til kvalitetselementerne, der er stillet i specifikationen i form af en række kvalitetsregler. Dette beskrives i de følgende afsnit.



Figur 16-1 Den eksterne kontrol.

Den eksterne kontrol består af to overordnede aktiviteter. Først skal kvalitetselementerne bestemmes ved at se på hele datasættet i forhold til en reference. Disse kvalitetselementer er med til at angive kvalitetsniveauet for datasættet. Den næste aktivitet består i at

kontrollere om det opnåede kvalitetsniveau også overholder de krav der er stillet i specifikationen.

Disse to aktiviteter beskrives i det følgende, hvor det først beskrives, hvordan kvalitetselementerne bestemmes.

## 16.1 Bestemmelse af kvalitetsniveau

En del af de kvalitetselementer, der er beskrevet tidligere i kapitlet om datakvalitet, kan først bestemmes når hele datasættet er registreret. De to kvantitative kvalitetselementer, der kan bestemmes her, er fuldstændighed og nøjagtighed, samt de tilhørende underelementer, og de skal begge bestemmes ud fra en sammenligning med et referencegrundlag. Dette referencegrundlag kan ligesom datagrundlaget, der blev beskrevet kapitel 14, bestå af forskellige former for geografisk relaterede informationer.

Det er ikke sikkert at det er nødvendigt med en fuldstændig kontrol, men at det er tilstrækkeligt med en kontrol af en mindre del af datasættet. Ved at foretage en kontrol vha. stikprøver fra et repræsentativt område, er det muligt at bestemme et bud på nøjagtigheden og fuldstændigheden for de forskellige klasser i datasættet. Ud fra en statistisk analyse, er det muligt at finde ud af, hvor godt disse kvalitetselementer er bestemt. Dette beskrives ikke nærmere her, men yderligere information om statistiske analyser kan bl.a. findes i [Petruccelli, 1999].

### 16.1.1 Referencegrundlag

Referencegrundlaget kan være de samme kilder, der er benyttet som datagrundlag til at registrere datasættet ud fra. Herved er det muligt at finde registreringsfejl ved at foretage en direkte sammenligning mellem datasættet og den kilde, der ligger til grund for registreringen.

Der kan dog også benyttes andre datakilder til at foretage kontrollen af datasættet. Det kan f.eks. være andre flyfotos, andre objektbaserede datakilder eller nogle helt andre data, der kan relateres til datasættet vha. andre geografiske data. Afhængig af hvilke kvalitetselementer, der skal kontrolleres, kan forskellige datakilder benyttes.

Flyfotos foretaget fra stor højde (større end datagrundlaget, hvis der er benyttet flyfotos), dvs. med en mindre nøjagtighed, kan måske godt benyttes til at bestemme fuldstændigheden samt den tematiske nøjagtighed for de forskellige objektklasser i datasættet. Derimod kan den geometriske nøjagtighed ikke nødvendigvis kontrolleres ved at benytte en datakilde, der har en dårligere detaljeringsgrad. Fordelen ved at benytte flyfotos, der er optaget fra stor højde, er at det er muligt

at dække et stort område med færre fotos. Herved bliver det billigere at foretage en komplet kontrol.

Hvis der derimod benyttes flyfotos foretaget fra en lavere flyvehøjde end det oprindelige datagrundlag, er det igen muligt at kontrollere fuldstændigheden, men det er samtidig også muligt at kontrollere den geometriske nøjagtighed. Derudover vil denne metode ligeledes medføre en bedre kontrol af den tematiske nøjagtighed da det er lettere at vurdere om et objekt er klassificeret korrekt når detaljeringsgraden er stor. At udføre kontrollen ud fra flyfotos foretaget fra en lavere flyvehøjde, er dog en noget dyrere metode, da den kræver en del flere flyfotos, for at kontrollere hele datasættet. Derfor er det her muligt f.eks. at foretage en stikprøve frem for en fuldstændig kontrol.

Der er fordele og ulemper, både ved at benytte de samme datakilder, som blev benyttet som datagrundlag, eller andre. Fordelen ved at benytte de samme datakilder er at det kan være billigere, da det ikke er nødvendigt at anskaffe sig yderligere datakilder i form af f.eks. nye billeder eller andet materiale. Derudover er der også den fordel, at det er muligt, at kontrollere den tidsmæssige nøjagtighed, da der ikke er nogen forskel i tiden mellem det oprindelige datagrundlag og referencegrundlaget. Hvis der derimod benyttes et referencegrundlag, der ikke er fra det samme tidspunkt som datagrundlaget, er det ikke muligt at kontrollere den tidsmæssige nøjagtighed ligeså godt. I det tidsrum, der er mellem de to datakilder, kan der være sket ændringer i de entiteter, der er registreret, og derved kan registreringer fra de to datakilder ikke direkte sammenlignes. Dette er dog en vurderingssag, der er afhængig af hvor hurtigt de ting, der observeres, ændrer sig. Inden for en relativt kort tidsperiode er det ikke sikkert, at der er sket væsentlige ændringer.

En af de største ulemper ved at benytte samme datakilder som referencegrundlag og som datagrundlag er, at der kan være fejl i disse kilder. Der kan være fejl i datakilderne, både hvis der benyttes eksisterende data, som tidligere beskrevet, eller flyfotos. Fejl i eksisterende data kan være opstået af mange grunde såsom tidligere fejlregistreringer eller fejlfortolkninger. Tilsvarende kan der også være opstået fejl når der benyttes flyfotos som datakilde. Det kan være fejl i de fikspunkter som billederne er rettet op efter, eller der kan være sket en anden fejl i denne oprettelse [Wolf & Dewitt, 2000]. Hvis den samme datakilde både benyttes som datagrundlag og referencegrundlag, opdages eventuelle fejl af denne type ikke, og derfor er det nødvendigt at benytte forskellige referencedatakilder.

Når kvalitetselementerne er bestemt skal de sammenlignes med de krav der er stillet i specifikationen. Dette beskrives i det følgende.

## 16.2 Kontrol af kvalitetselementer

Kvalitetsniveauet for et datasæt bedømmes samlet, ud fra de kvalitetselementer, der er beskrevet i det tidligere kapitel om datakvalitet. For at være i stand til at foretage denne bedømmelse, er det nødvendigt at alle kvalitetselementer er repræsenterede. Dette gælder både de netop bestemte elementer; fuldstændighed og nøjagtighed, men også de kvalitetselementer, der er angivet og kontrolleret tidligere under den interne kontrol.

Samtidig med kontrollen af, om de enkelte kvalitetselementer eksisterer, skal det kontrolleres, om de overholder de krav, der er stillet i specifikationen. Det kan f.eks. være at fuldstændigheden for objekter fra objektklassen Bygning skal være 99 %, eller at den geometriske nøjagtighed skal være bedre end 1 meter.

Når de forskellige kvalitetselementer er bestemt skal de angives i data, hvilket beskrives i det følgende.

### 16.2.1 Håndtering af kvalitetselementer

Kvalitetselementerne kan angives på tre niveauer; datasætniveau, objektklasseniveau eller objektniveau. På datasætniveau tilføjes kvalitetselementer, der er gældende for hele datasættet, som f.eks. fuldstændighed. På objektklasseniveau tilføjes kvalitetselementer, der er gældende for alle objekter, i en bestemt objektklasse. Det kan f.eks. være geometrisk nøjagtighed, men kan også være fuldstændighed for objekter fra den pågældende klasse. Kvalitetselementer, der knyttes til data på objektniveau, beskriver hvert enkelt objekt for sig selv, såsom præcision. De anførte eksempler beskrives nærmere i det følgende.

Kvalitetselementer kan f.eks. opbevares i en selvstændig fil, som et objekt i data, eller som attributter i de enkelte objekter. I ArcGIS er det muligt at håndtere metadata til at beskrive de data, der ligger i et datasæt eller i en objektklasse [Vienneau, 2001]. Her gemmes metadata i formatet XML, enten i en fil ved siden af Geodatabasen, eller i selve Geodatabasen, hvilket gør denne information læsbar for mange forskellige applikationer.

Den måde metadata vises på, i ArcGIS er meget overskuelig, men samtidig også meget omfattende. Det er muligt at indtaste en lang række informationer, der kan gemmes sammen med data, og det gør, at de altid er tilgængelige, når data benyttes. De enkelte

kvalitetslementer fra kapitlet om datakvalitet, kan føjes til data vha. den skabelon, der er opstillet i ArcGIS.

De forskellige kvalitetslementer under fuldstændighed og nøjagtighed registreres alle under den eksterne kontrol, og kræver, at hele datasættet er samlet, for at de er mulige at bestemme, og de skal angives på datasætniveau og objektklasseniveau. Disse kvalitetslementer er ikke kun med til at beskrive kvaliteten for det enkelte objekt i en objektklasse, men også for samtlige objekter i et datasæt. En af grundene til at kvalitetslementerne knyttes til data på begge niveauer er, at de er et udtryk for enten hele datasættet eller hele klassen og ikke kun beskriver ét enkelt objekt. Det samme er gældende for kvalitetslementet logisk konsistens, der ligeledes er en beskrivelse af hvor rigtigt data er i forhold til hele datasættet eller hele objektklassen.

Præcision og oprindelse hænger i høj grad sammen, som beskrevet i det tidligere i kapitel 11 om datakvalitet. Da disse to kvalitetslementer ikke er optaget på baggrund af hele datasættet, men derimod er en beskrivelse af hvert objekt, skal de to elementer angives på objektniveau. Oprindelsesoplysningerne og præcisionen skal altså knyttes til det enkelte objekt.

Egentlig burde samtlige punkter i datasættet have tilknyttet oplysninger omkring alle kvalitetslementer, der hører til det pågældende datasæt. Med punkter menes ikke kun geometritypen punkt, men også samtlige registrerede punkter i forbindelse med geometrityperne linier og polygoner. Ved at relatere kvalitetslementerne til hvert enkelt punkt, er det muligt at foretage små rettelser i et objekt, som f.eks. en bygning eller en vej, og herved ændres kun på kvalitetslementerne for de punkter, der er ændret. På denne måde er det muligt altid at holde styr på hvilken baggrund data stammer fra og derved vurdere om disse data er anvendelige. Der skal dog også være en samlet beskrivelse af hele datasættet eller objektklasserne, for at give et overblik, så det er muligt at vurdere den samlede kvalitet. Dette giver store datamængder, og er dermed pladskrævende, og gør det svært at skabe et overblik over alle informationer.

Om det er nødvendigt at have disse informationer alle steder i data, er dog meget afhængig af hvilket produkt der er tale om, hvad det skal bruges til, samt hvilke krav, der stilles til f.eks. ajourføring. Hvis ajourføringen foregår ved at foretage en ny registrering af hele objektet når der er sket en ændring, er det ikke nødvendigt med informationer om kvaliteten i hvert enkelt punkt i en polygon eller linie. Da denne information er den samme for samtlige punkter i

objektet, er det kun nødvendigt at knytte kvalitetselementerne til objektet og ikke til punkterne. Hvis ajourføringen derimod foregår ved at rette enkelte punkter i et objekt, kan det være relevant at knytte kvalitetselementerne til hvert enkelt punkt. Herved er det f.eks. muligt at finde ud af hvornår og hvem der har udført rettelserne, samt hvordan de er foretaget. Det kan f.eks. være at rettelserne ikke er foretaget med samme metode som de resterende punkter i objektet, og der derved ikke er de samme forventninger til kvaliteten af denne del af objektet.

### 16.3 Afslutning

Når de sidste kvalitetselementer er bestemt, sammenlignes de som beskrevet, med de kvalitetskrav, der er stillet i specifikationen. Hvis data ikke lever op til de stillede krav, skal det vurderes hvilke konsekvenser det skal have.

Konsekvenserne af at kvalitetskravene ikke overholdes, er afhængig af hvilke krav, der ikke overholdes, samt i hvor høj grad de ikke overholdes. Hvis f.eks. den tidsmæssige nøjagtighed ikke er overholdt, skal det sammenholdes med betydningen af denne fejl. Det kan være, at det ikke har så stor betydning for det endelige produkt, at der er denne unøjagtighed. Det afhænger igen af, hvilke krav der stilles til data.

Hvis det vurderes, at det er nødvendigt, må datasættet eller objekterne fra den pågældende klasse registreres på ny. Det kan være, at fejlene er opstået under fortolkningen af det anvendte datagrundlag, men det kan også være, at der er fejl i datagrundlaget. Derfor er det nødvendigt, at der foretages en vurdering af, hvor fejlen er opstået, og på baggrund af dette, kan der foretages en revision af data. Dette kan gøres ved enten at benytte samme metode og datagrundlag, hvis der er tale om sjusk, eller ved at skifte fejkilderne eller de fejlagtige metoder, hvis datagrundlaget eller metoderne ikke er tilstrækkelige.

Når alle krav er overholdt, kan datasættet overføres til det endelige produkt, der nu er klar til at blive anvendt.



# Del 4 – Afslutning

De spørgsmål, der er stillet i problemformuleringen, er alle besvarede i den foregående del af rapporten. I denne afsluttende del foretages der en opsamling af de opnåede resultater. Dette gøres i den følgende konklusion. Dernæst opstilles en række forslag til hvad det arbejde, der er udført i forbindelse med dette projekt, kan udvides til. Disse udvidelsesmuligheder beskrives i den efterfølgende perspektivering.



## 17 Konklusion

Ved produktion af geografisk data er et af hovedproblemerne, at det skal sikres, at produktet overholder de krav, der er stillet i specifikationen. Specifikationen er en vigtig del af produktionen, og det er nødvendigt at denne specifikation er klart og tydeligt formuleret, så der ikke opstår problemer når den benyttes. Dette gøres ved at strukturere de krav, der stilles i specifikationen, i forskellige grupper af regler. De enkelte regelgrupper benyttes forskellige steder i produktionen.

Produktionen kan inddrages i fire overordnede processer. Den første proces består i at udarbejde en specifikation for produktet. Dernæst genereres data ud fra de opstillede fortolkningsregler og et givet datagrundlag. Herefter kontrolleres de registrerede objekter for fejl i den interne kontrol, hvor de regler, der kan formaliseres, benyttes. Den sidste proces består i at sammenligne det samlede datasæt med et referencegrundlag, og derved bestemme kvalitetselementerne, og undersøge, om de overholder de krav, der stilles i specifikationen.

Det er vist, at en del af reglerne fra specifikationen kan formaliseres, og det derved er muligt at opstille funktioner, der kan foretage kontrollen af disse regler automatisk. På den måde gøres arbejdet lettere for en operatør, der skal foretage registreringerne ud fra et givet datagrundlag. Det sikrer også, at data overholder disse regler, uanset hvilken datakilde der benyttes. Grunden til at det letter arbejdet for operatøren er, at det ikke er nødvendigt koncentrere sig om at overholde denne gruppe af regler, og fokus i stedet kan flyttes til fortolkningen af datagrundlaget.

Det er hovedsageligt gruppen af topologiske regler, der kan formaliseres. For at kontrollere de topologiske regler, er det nødvendigt at identificere hvilke topologiske relationer, hver enkelt regel omhandler. Ved hjælp af en metode, der kaldes den dimensionalt udvidede 9-skæringsmatricemodel, er det muligt at identificere samtlige topologiske relationer mellem to geometriske objekter i to dimensioner. Herved kan de topologiske relationer formaliseres vha. værdier i en 3x3 matrice. Denne matrice kan derefter benyttes i en funktion, til at bestemme om relationen er gældende for de to objekter, hvorpå eventuelle topologiske fejl kan identificeres.

Når den automatiske kontrol af et objekt er foretaget, og der er identificeret en fejl, skal denne fejl rettes. Det kan i nogle tilfælde

gøres automatisk, men det kræver, ofte at der foretages en ny fortolkning af det relevante objekt. Det kan skyldes en fejl under fortolkningen, men det kan også være, at fejlen skal betegnes som en undtagelse.

Det er vigtigt at dokumentere, at produktet overholder specifikationen. Det kan gøres ved at angive nogle elementer, der gør det muligt at vurdere kvaliteten af det samlede produkt. Kvaliteten af data kan beskrives på tre forskellige niveauer; datasæt-, objektklasse- og objektniveau. På hvert niveau beskrives kvaliteten af kvalitetselementerne, hvorved det er muligt at benytte en mindre del af produktet, og stadig have information om kvaliteten.

I den efterfølgende perspektivering beskrives en række fremtidige muligheder, der kan være interessante at arbejde videre med, i forbindelse med automatisk kontrol og sikring af, at kvaliteten af data dokumenteres.

## 18 Perspektivering

Der er mange muligheder for at arbejde videre ud fra dette projekt. F.eks. kan hele produktionsprocessen integreres i en programpakke, der indeholder alle fire dele af produktionen. Det giver en ensartet struktur for forskellige produkter, og letter hele produktionen.

Specifikationen kan genereres i et program, hvor det er muligt at opbygge datamodellen i et objektorienteret modelleringsværktøj. Herefter er det muligt at definere alle typer regler vha. værktøjer som det, der er beskrevet i rapporten, og som benyttes til topologiske regler.

Specifikationen gemmes herefter i en database, hvorfra der kan realiseres en række forskellige versioner af den. Der kan f.eks. udtrækkes en specifikation på papirform, ligesom specifikationen til Top10dk. Der er også en matematisk version, som indeholder de funktioner, der skal benyttes i de applikationer, hvor registreringen foregår. Registreringsreglerne, der ikke kan formaliseres, vises i applikationen som en liste, så operatøren har et bedre overblik over dem.

Data bliver gemt, enten i en objektorienteret database, hvor reglerne er implementeret i funktioner, eller i en objekt-relational database, hvor reglerne er indbygget i COM-objekter, som i ArcGIS.

Først opbygges datamodellen i modelleringsværktøjet. Dette inkluderer formatregler for attributterne, geometriske regler og kvalitetsregler. Herefter defineres registreringsreglerne for hver enkelt klasse i et værktøj. Her indtastes de forskellige regler, der ikke kan formaliseres, såsom at bygninger skal registreres ved tagudhæng. De topologiske regler defineres i et andet værktøj, der fungerer som det, der er beskrevet i kapitel 15. Når alle regler er defineret, bliver specifikationen gemt, og der genereres en datamodel, og funktionerne, der realiserer reglerne, bliver automatisk implementeret i COM-objekter.

Det er nu muligt at tilføje data til modellen. Da, det er, en objektorienteret model, er det muligt at anvende flere forskellige applikationer til dette. Det kan f.eks. være en applikation til en fotogrammetrisk arbejdsstation, hvor data bliver tilføjet til databasen ét objekt af gangen, hvorefter det kontrolleres med reglerne. Det kan også være et program, som ArcCatalog, hvor en samling af objekter

kan lægges i databasen. Herefter kan disse objekter valideres vha. reglerne.

På baggrund af kvalitetsreglerne lagres kvalitetslementerne enten på datasæt-, objektklasse- eller objektniveau. Der kan f.eks. opbygges et hierarki af klasser, der repræsenterer de forskellige kvalitetslementer, og objekter, objektklasser eller datasæt kan da relatere til objekter fra kvalitetsklasserne. På den måde er det muligt at tilføje data med forskellig kvalitet til det samme datasæt, og stadig have information om dette i forskellige kvalitetsobjekter, som data kan relatere til. På den måde kan nogle bygninger i et datasæt være registreret med fotogrammetrisk opmåling af flyfotos, og henvise til ét kvalitetsobjekt, mens andre bygninger kan være målt op med GPS, og henvise til et andet kvalitetsobjekt.

Dette system kræver følgende:

- Der skal udvikles en datamodel, der kan indeholde regler.
- Reglerne skal grupperes og der skal udvikles metoder til at formalisere dem.
- Der skal udvikles et værktøj til at opbygge specifikationen, og dermed datamodellen og reglerne.
- Det skal undersøges hvordan specifikationen skal gemmes i databasen.
- Der skal udvikles et værktøj, der kan omsætte reglerne til kode, og implementere det i COM-objekter.
- Der skal opbygges et hierarki af kvalitetsklasser.
- De applikationer, der benyttes til at føje data til databasen skal understøtte denne model.

I denne rapport er der især fokuseret på de to første punkter, men der er også givet et eksempel på hvordan værktøjet, til at opbygge topologiske regler kan se ud, samt hvilke funktioner, der kan benyttes til de topologiske undersøgelser. Det er dog stadig nødvendigt at udvikle et program, der benytter disse ting.

Andre rumlige databaser kan også benytte et sådan system, men også databaser der ikke har rumlige attributter. Jo mere kompleks data er, jo mere er der brug for en overordnet kontrol af det, der tilføjes.

## 19 Litteraturliste

- Jens Michael Carstensen (2000), Digital image processing
- Jesper Vinther Christensen (1997), Objekt-Orientering og Kvalitet af Geodata
- John Kamper, Jesper Vinther Christensen (2002), System til ajourføring af 3D kortdatabaser
- Jesper Vinther Christensen, Anders Friis-Christensen (2003), A Framework for Modeling Quality of Geographic Data
- Jesper Kihlberg Christiansen, Karsten Pihl Olsen (2002), Modellering af tagflader fra laserscannede data
- Jesper Kihlberg Christiansen, Karsten Pihl Olsen (2003), Kvalitetssikring af Geografiske data, En forundersøgelse
- Sophie Cockcroft (1997), A Taxonomy of Spatial Data Integrity Constraints
- Sophie Cockcroft (1998), User defined spatial business rules: storage, management and implementation – a pipe network case study
- C. J. Date (2000), An introduction to database systems, 7th international edition
- Steve DeMartino, Eric Hrnicek (2001), Object-oriented GIS 101: New GIS Data Model Features Intelligent Objects
- Dr. Gui (2003), <http://www.microsoft.com/com/news/drgui.asp>, juli 2003
- M. J. Egenhofer (1994), Definitions of Line-Line Relations for Geographic Databases

- M. J. Egenhofer, A. R. B. M. Shariff og D. M. Mark (1998), Natural-Language Spatial Relations Between Linear and Areal Objects: The Topology and Metric of English-Language Terms
- ESRI (2003a), [www.esri.com](http://www.esri.com), juli 2003
- ESRI (2003b), [web.utk.edu/~sshaw/ESRICASETtools.pdf](http://web.utk.edu/~sshaw/ESRICASETtools.pdf), juli 2003
- M. Fowler (2000), UML distilled: A brief guide to the standard object modeling language
- Anders Friis-Christensen, Nectaria Tryfona Christian S. Jensen (2001), Requirements and Research Issues in Geographic Data Modeling
- Tom Gilb (2002), Competitive Engineering
- Jens Gottlieb (2003), KMS, møde d. 28.5.2003
- Stephen. C. Guptill, Joel. L. Morrison (1995), Elements of spatial data quality
- ISO (2001), Geographic information – Quality principles, ISO/DIS 19113
- ISO (2003), Geographic information – Data product specification, ISO/DIS 19131
- Kommunalteknisk chefforening (2000), Specifikation for Teknisk Kort TK99 2. udgave
- Kort & Matrikelstyrelsen (2001), Top10dk Geometrisk registrering specifikation udgave 3.2.0
- Neal Leavitt (2000), Whatever Happened to Object-Oriented Databases
- Andrew MacDonald (2001), Building a Geodatabase
- Dara Obasanjo (2001), An Exploration of Object Oriented Database Management Systems



- 
- Karsten Pihl Olsen, Annemarie Arnvig Hansen og Erik Norström Andersen (2000), Laserscanning og bymodeller
  - OpenGIS Consortium (1999), OpenGIS Simple Features Specification for SQL, Rev. 1.1
  - Thomas René Lindegaard Pedersen (2000), Udvikling af objektorienteret trafik GIS
  - Jesper Kaae Petersen (1992), Objektorienteret digital kortlægning
  - Joseph Petruccelli (1999), Applied Statistics for Engineers and Scientists
  - Sylvie Servigne, Thierry Ubeda, Alain Puricelli og Robert Laurini (1999), A Methodology for Spatial Consistency Improvement of Geographic Databases
  - Shashi Shekhar, Sanjay Chawla (2003), Spatial Databases: A Tour
  - Aleta Vienneau (2001), Using ArcCatalog, ESRI Press
  - Poul R. Wolf, Bon A. Dewitt (2000), Elements of Photogrammetry, 3rd edition
  - Peter Woodsford (1995), The Significance of Object-Orientation for GIS
  - Michael F. Worboys (1995), GIS—A Computing Perspective
  - Michael Zeiler (2000), Modeling Our World
  - Michael Zeiler (2001), Exploring ArcObjects, vol. 2, app. B



# Appendiks

De følgende appendikser indeholder en række forskellige ting, der er med til at uddybe konkrete emner i rapporten. Det første giver en detaljeret oversigt over mulige skæringsmatricer mellem de geometriske objekter; punkt, linie og polygon. Dernæst beskrives forskellige fremgangsmåder til etablering af en Geodatabase, samt hvordan regler kan defineres, i ArcGIS.

De enkelte appendikser findes på følgende sider:

<b>APPENDIKS 1 – TOPOLOGISKE RELATIONER</b>	<b>145</b>
<b>APPENDIKS 2 - KONSTRUKTION AF DATAMODEL</b>	<b>151</b>
<b>APPENDIKS 3 - CASE-TOOL</b>	<b>163</b>
<b>APPENDIKS 4 - TOPOLOGI I ARCGIS</b>	<b>177</b>





## Appendiks 1 – Topologiske relationer


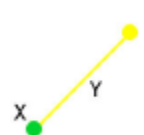
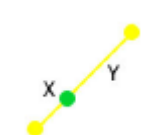
Dette appendiks indeholder en beskrivelse af de mulige skæringsmatricer mellem forskellige geometriske objekter. Dette gøres ved først at beskrive skæringen mellem et punkt og hhv. et andet punkt, en linie og en polygon. Dernæst beskrives skæringen mellem en linie og en anden linie samt en linie og en polygon efterfulgt af skæringen mellem to polygoner.

### 1.1 Skæringsmatricer med et punkt

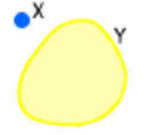
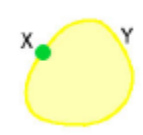
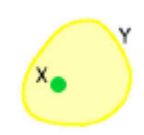
For to punkter er der følgende mulige skæringsmatricer:

79	273
	
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

For et punkt og en linie er der følgende skæringsmatricer:

79	151	295
		
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$



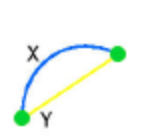




For et punkt og et polygon kan de samme skæringsmatricer forekomme:

79	151	295
		
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$

## 1.2 Skæringsmatricer for en linie

Der er 33 mulige relationer mellem to linier. Disse 33 relationer kan deles op i to typer, symmetriske og asymmetriske. Der er 13 symmetriske, hvor skæringsmatricen er symmetrisk omkring diagonalen, og 10 par asymmetriske, hvor skæringsmatricen er asymmetrisk, men for hvert par kan matricen transponeres for at få den modsvarende matrice [Egenhofer, 1994].

Symmetriske:

79	95	85	341	273	495	335
						
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

351	239	245	501	229	485
$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$

Herunder ses et eksempel for hvert par af de asymmetriske relationer:

111/207	367/463	295/457	311/473	359/461
$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$

103/205	221/119	477/375	237/231	493/487
$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$

APPENDIKS 1

For en linie og en polygon er der 19 mulige skæringer [Egenhofer et al., 1998]:

79	87	95	151	207	215	223
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

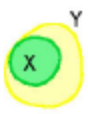

279	295	311	407	423	439	463
$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

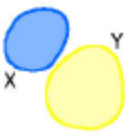
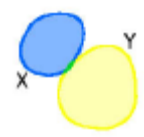


471	479	487	495	503
$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$



### 1.3 Skæringsmatricer for en polygon

For to polygoner kan der forekomme 8 forskellige skæringer, hvoraf de fire er symmetriske. Derudover er der 2 par asymmetriske skæringsmatricer. De symmetriske vises øverst i nedenstående tabel. Et eksempel på hvert par af de asymmetriske er vist under de symmetriske:

439/505	295/457
	
$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$

79	95	511	273
			
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

De mulige skæringsmatricer er nu beskrevet, men yderligere gives i [Egenhofer et al., 1998] og [Egenhofer, 1994].

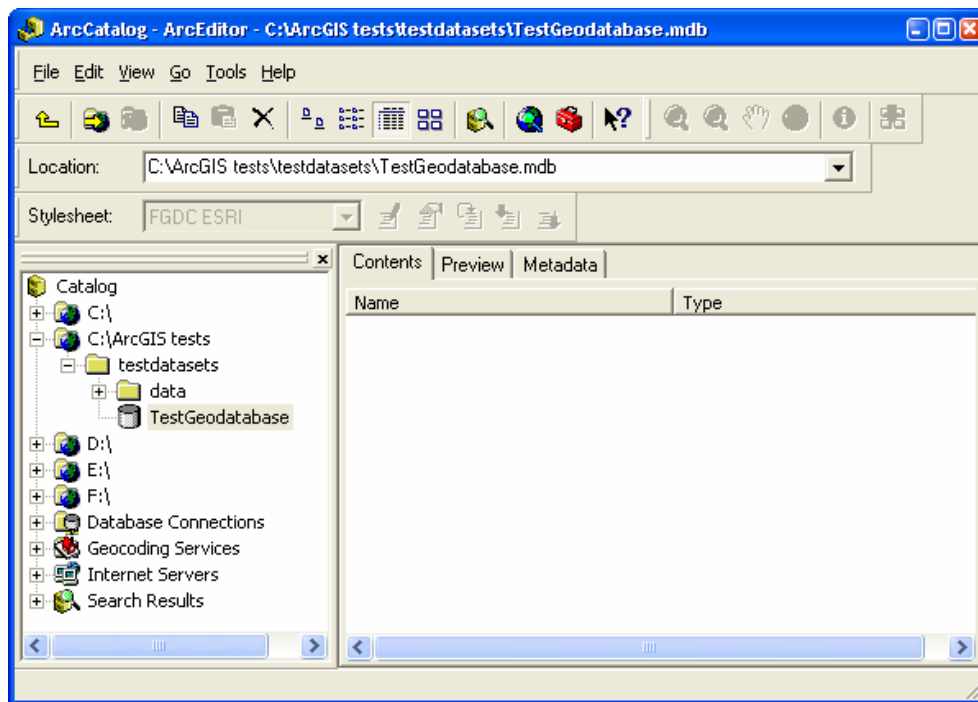


## Appendiks 2 - Konstruktion af datamodel

I dette appendiks beskrives det, hvordan en Geodatabase oprettes i ArcGIS, hvordan der tilføjes et datasæt og hvordan de forskellige formatregler i ArcGIS opbygges. Som eksempel benyttes datasættet, der er beskrevet i kapitel 15 om topologi.

Det værktøj, der benyttes til at opbygge en Geodatabase, er ArcCatalog, der benyttes til at organisere og håndtere geografiske data. ArcCatalog indeholder bl.a. værktøjer til at indtaste og se metadata, se eksempler fra datasættet samt definere strukturen for geografiske data.

På figur 1 ses ArcCatalogs layout. I venstre side er der en træstruktur, der viser de tilkoblede foldere. Det er muligt at forbinde til lokale foldere, foldere på et netværk, samt foldere på internettet. I højre side er der muligheder for at se en oversigt over indholdet af den valgte folder, datasæt eller fil, det er muligt at se et preview af data, eller at se metadata for det valgte. For oven er en række værktøjslinier, der giver adgang til forskellige funktioner.

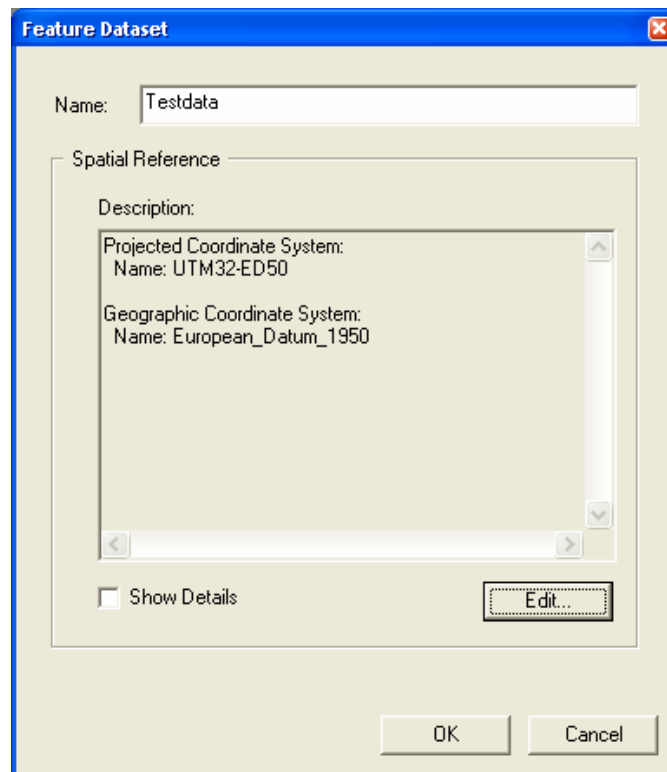


Figur 1 Layout i ArcCatalogs

## 2.1 Opret ny Geodatabase

For at oprette en ny Geodatabase, navigeres der hen til den folder, hvor den ønskes oprettet, og herefter højreklikkes der og vælges "New -> Personal Geodatabase". Geodatabasen bruges til at opbevare feature datasæt og feature klasser.

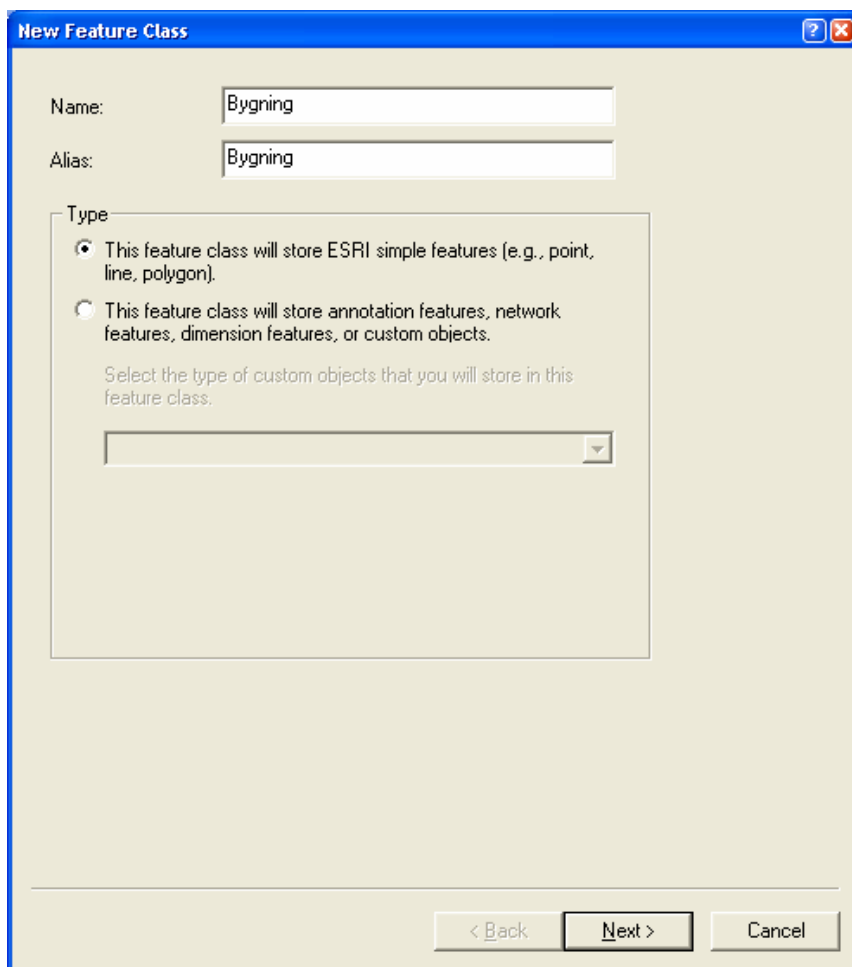
For at oprette et datasæt højreklikkes der på Geodatabasen, og vælges "New -> Feature Dataset". Herefter fremkommer dialogen, der er vist på figur 2. Her er det muligt at definere navn og koordinatsystem for feature datasættet.



Figur 2 Dialogen "New Feature Dataset"

I dette tilfælde er det valgt at kalde feature datasættet Testdata, og det benytter UTM32-ED50 som koordinatsystem.

Efter at have oprette datasættet skal der tilføjes featureklasser. Dette gøres ved igen at højreklikke på feature datasættet og vælge "New -> Feature Class". Herefter kommer New Feature Class-dialogen frem, hvor første skærbillede i dialogen er vist på figur 3. Her indtastes navn og eventuelt alias, og det vælges om klassen skal benytte simpel geometri, eller specielle typer, såsom netværk, kommentarer, eller *Custom Features*, der bliver nærmere beskrevet i Appendiks 3. I dette tilfælde vælges navnet Bygning og simpel geometri.

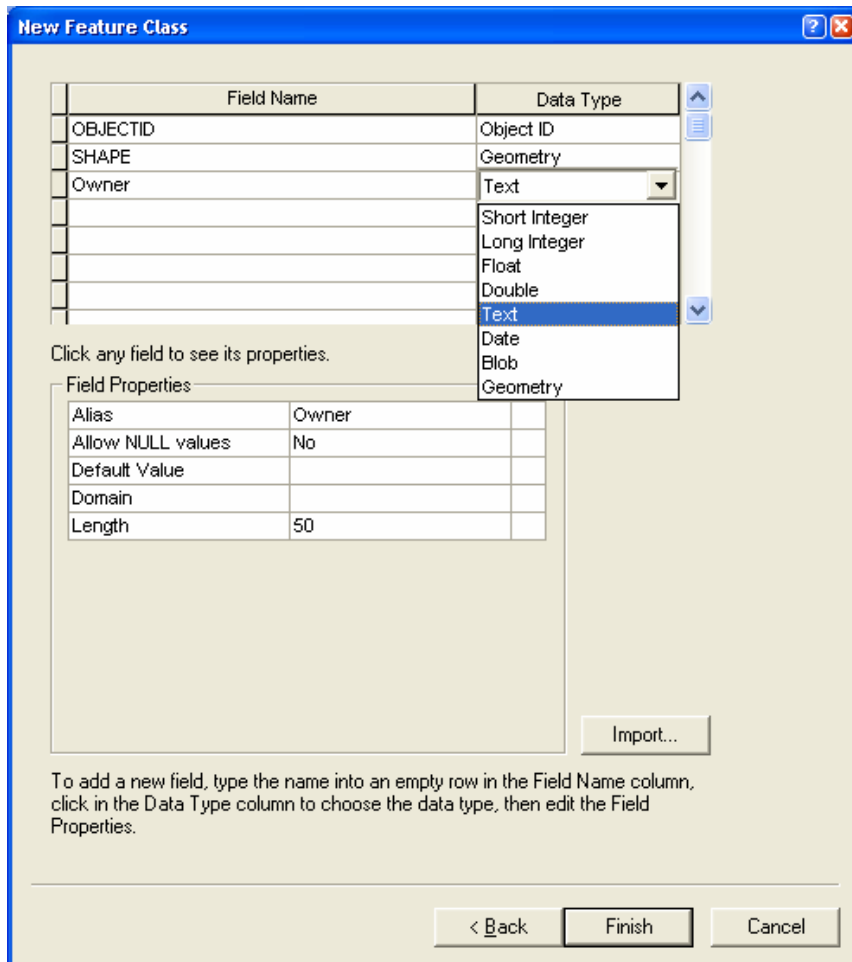


Figur 3 Første skærbillede for dialogen New Feature Class

Andet skærbillede i dialogen benyttes kun, hvis Geodatabasen understøtter ArcSDE, hvilket ikke beskrives nærmere her, men yderligere information herom kan findes i [MacDonald, 1999].

Det tredje skærbillede i dialogen giver mulighed for at definere hvilke attributter (fields), der skal være knyttet til klassen. Der er fra start tilknyttet to attributter, OBJECTID, der er af typen "Object ID", og SHAPE, der er af typen "Geometry". Disse typer har forskellige egenskaber, der kan ændres. På figur 4 vises det tredje skærbillede samt de forskellige egenskaber for typen "Geometry". Det ses at egenskaben "Spatial reference" automatisk er blevet sat til det samme som det koordinatsystem, der var blevet valgt for feature datasættet.

For at tilføje nye attributter, indtastes navnet på attributten på en ledig plads i "Field Name", og datatypen vælges herefter til højre for dette. Dette er illustreret ved at tilføje attributten Owner, der skal have typen "Text". På figur 4 ses det hvilke datatyper, der ellers er tilgængelige. Herefter trykkes der på "Finish" og Feature klassen er tilføjet til datasættet.

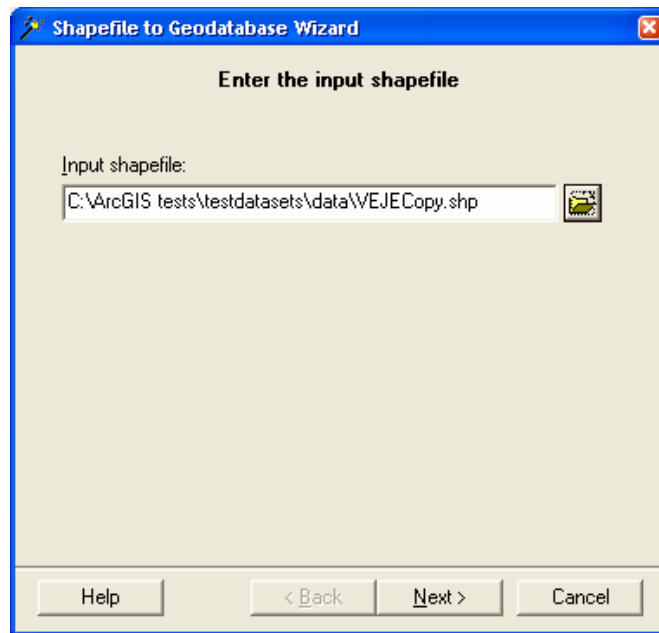


Figur 4 New Feature Class Skærbillede 3

## 2.2 Import af data

Det er også muligt at importere data til en klasse. For eksempel er der en funktion "Shapefile to Geodatabase Wizard", der guider processen med at importere en shapefil til Geodatabasen. Wizarden startes ved at

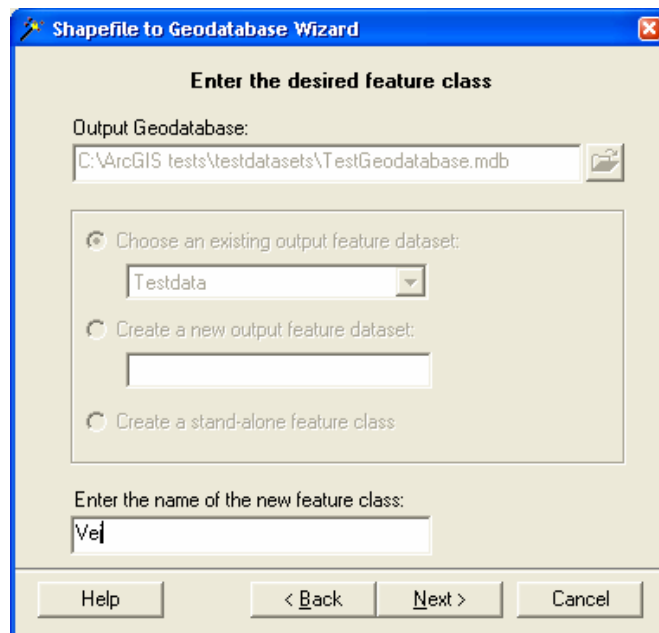
højreklikke på feature datasættet, og vælge ”Import -> Shapefile to Geodatabase Wizard”. Herved fremkommer dialogen, der ses på figur 5. Den shapefil, der skal importeres, vælges. I dette tilfælde vælges et udsnit fra Top10dk i området omkring Hundested.



Figur 5 Første skærmbillede for dialogen Shapefile to Geodatabase Wizard

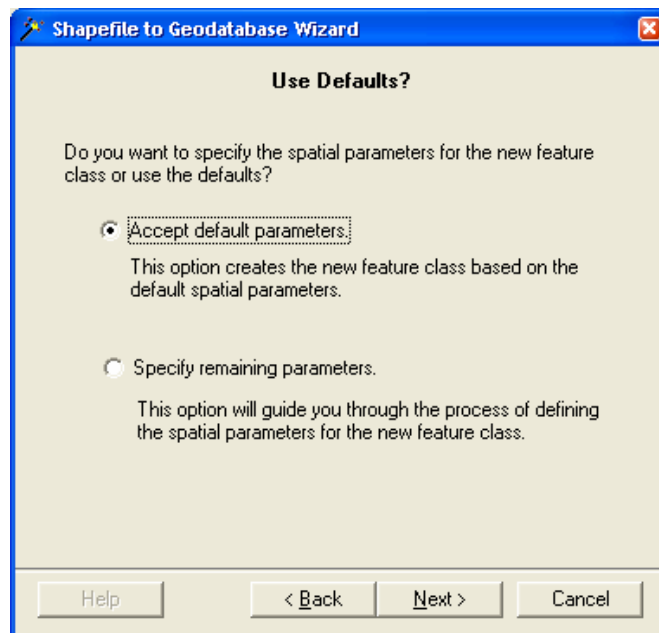
På andet skærmbillede, der er vist på Figur 6, er det muligt at vælge hvilken Geodatabase og feature datasæt, der skal eksporteres til. Både Geodatabasen og feature datasættet er allerede defineret, da der blev klikket på et feature datasæt da ”Import” blev valgt, og det er derfor ikke nødvendigt at gøre her. Det er kun nødvendigt at angive hvad navnet på den nye feature class skal være, og her kaldes den nye klasse for ”Vej”.





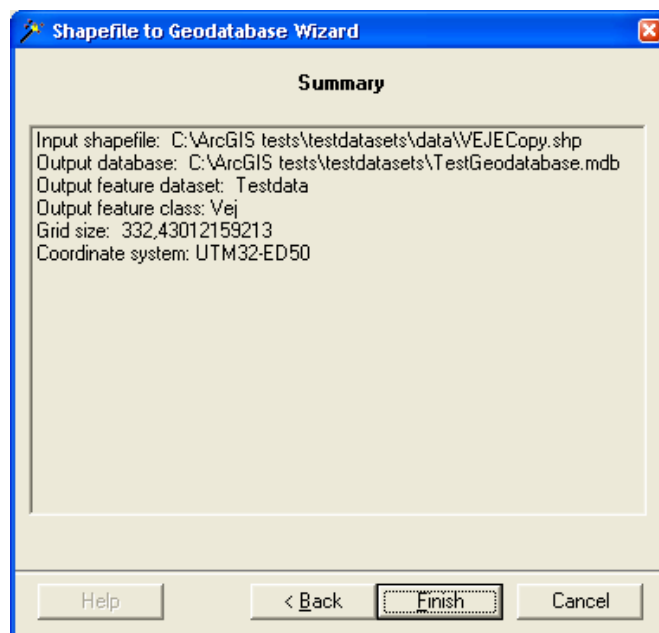
Figur 6 Andet skærbillede for dialogen Shapefile to Geodatabase Wizard

Det næste, der skal defineres, er de rumlige parametre, såsom koordinatsystem, samt hvilke attributter, der ønskes importeret fra Shapefilen. Her vælges "Accept default parameters", da koordinatsystemet er det samme som i feature datasættet, og alle attributter skal tages med. Dette gøres på dialogens tredje skærbillede, der ses på figur 7.



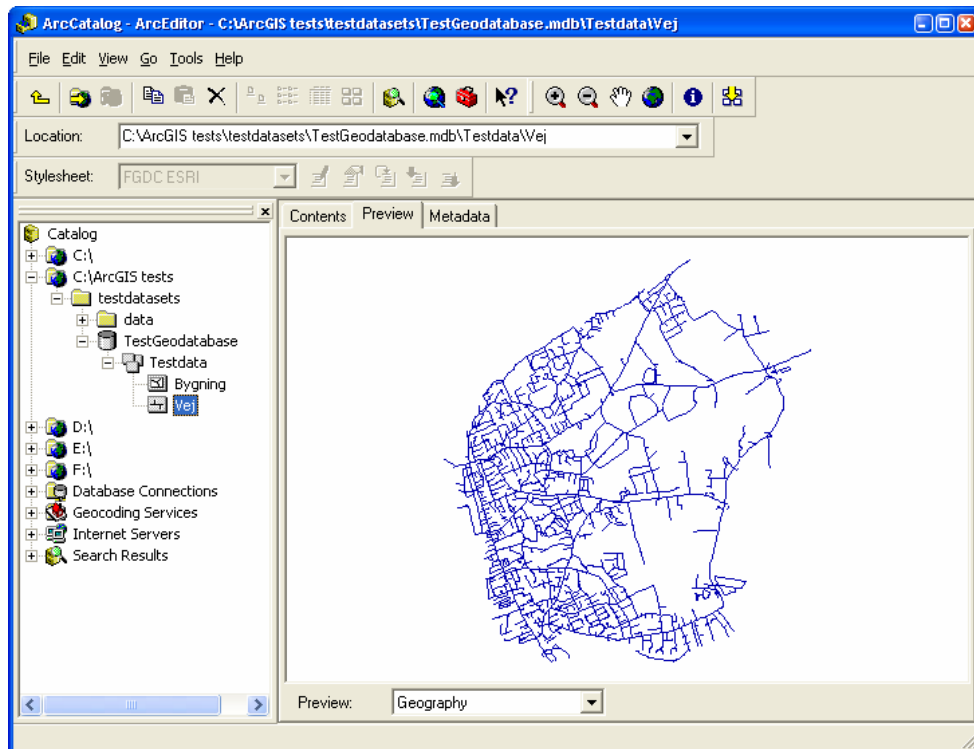
Figur 7 Det tredje skærmbillede i dialogen Shapefile to Geodatabase Wizard

Figur 8 viser det sidste skærmbillede i Shapefile to Geodatabase Wizarden. Den giver en oversigt over de valg, der er foretaget. Herefter trykkes der på "Finish", og den valgte Shapefile bliver oversat til en Feature class.



Figur 8 Sidste skærmbillede i dialogen Shapefile to Geodatabase Wizard

Efter oversættelsen er det muligt at se et eksempel på hvordan det importerede data ser ud. Til venstre ses Geodatabasen, feature datasættet Testdata og de to featureklasser Bygning og Vej. Til højre ses et eksempel på en repræsentation af klassen Vejs geometri.



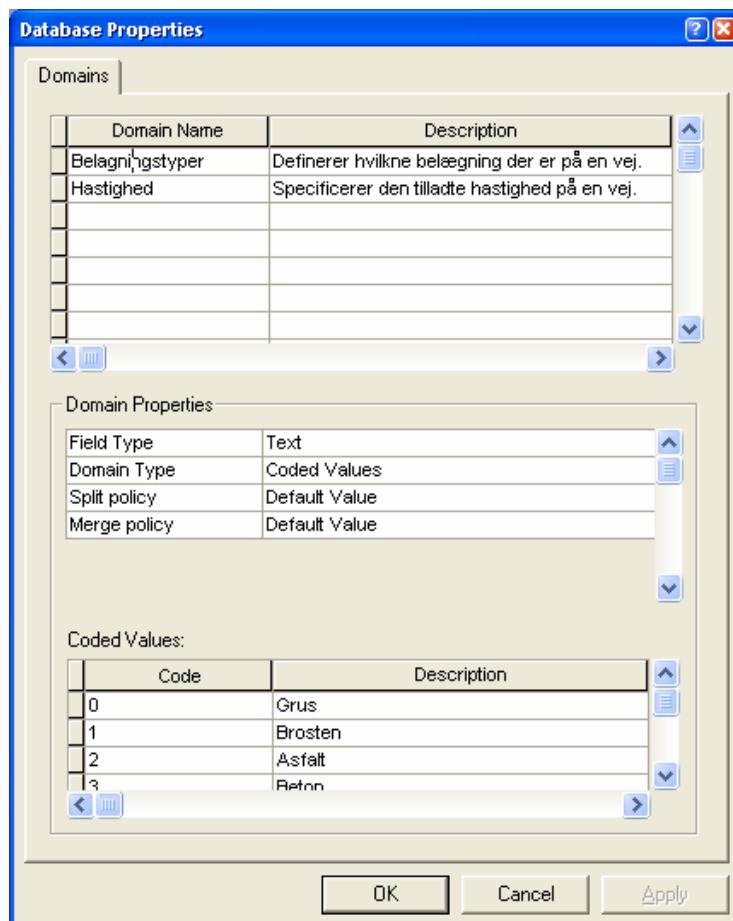
Figur 9 Det importerede data kan ses i ArcCatalog

I kapitel 15 er det beskrevet hvordan det er muligt at bruge kodet værdidomæne (Coded Value Domains) og værdimængdedomæne (Range Domains) til bl.a. at reducere antallet af indtastningsfejl. I det følgende beskrives det, hvordan disse ting oprettes og bruges i ArcGIS.

## 2.3 Domænereregler

For at en klasses attribut kan benytte en domænerregel, skal domænerne først defineres i Geodatabasen. Dette gøres ved at højreklikke på databasen og vælge "Properties", hvorefter databasens egenskaber vises. Her er det muligt at definere de to forskellige domænetyper.

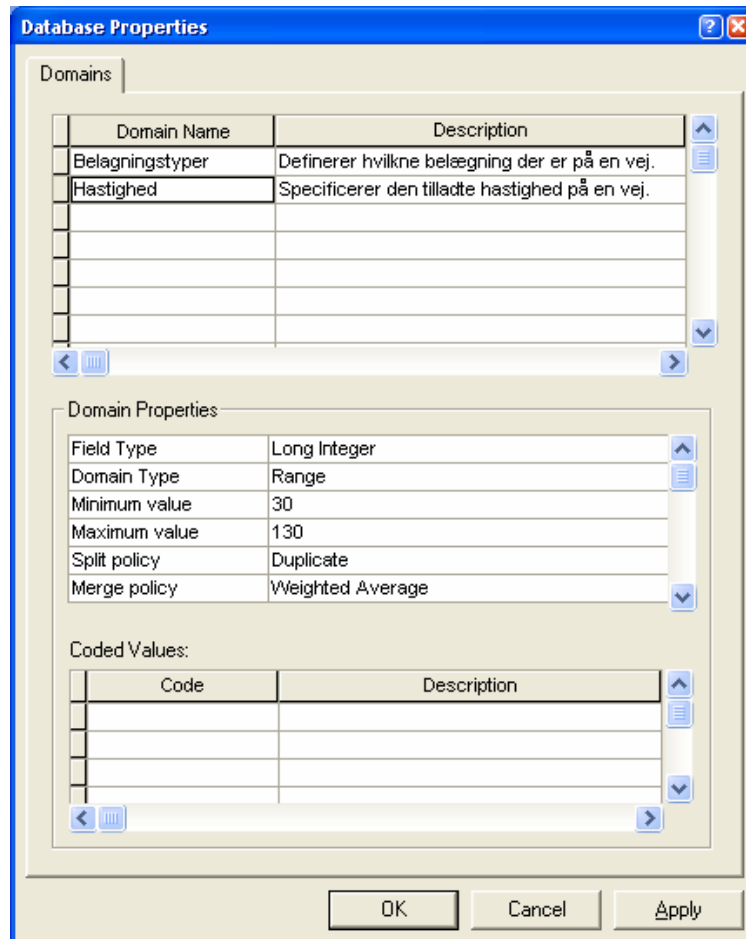
På figur 10 vises dialogen, hvor domænerne defineres. Øverst er en liste over de domæner, der er i databasen. Her er det øverste domæne "Belagningstyper" valgt. Ud over navnet står der en kort beskrivelse af domænet. Nedenunder kan domænets egenskaber indtastes. Det er muligt at vælge om domænet skal være et kodet værdidomæne eller et værdimængdedomæne. I dette tilfælde er et kodet værdidomæne valgt. Derudover er det muligt at vælge, hvilken type attribut (Field type), domænet skal repræsentere, her er "Text" valgt. Det er desuden muligt at definere hvilke værdier, der skal gælde, hvis to objekter bliver slået sammen, eller hvis et objekt bliver splittet op. Under domænets egenskaber, er det muligt at indtaste de forskellige værdier og beskrivelser. Her er indtastet fire muligheder.



Figur 10 Kodet værdidomæne

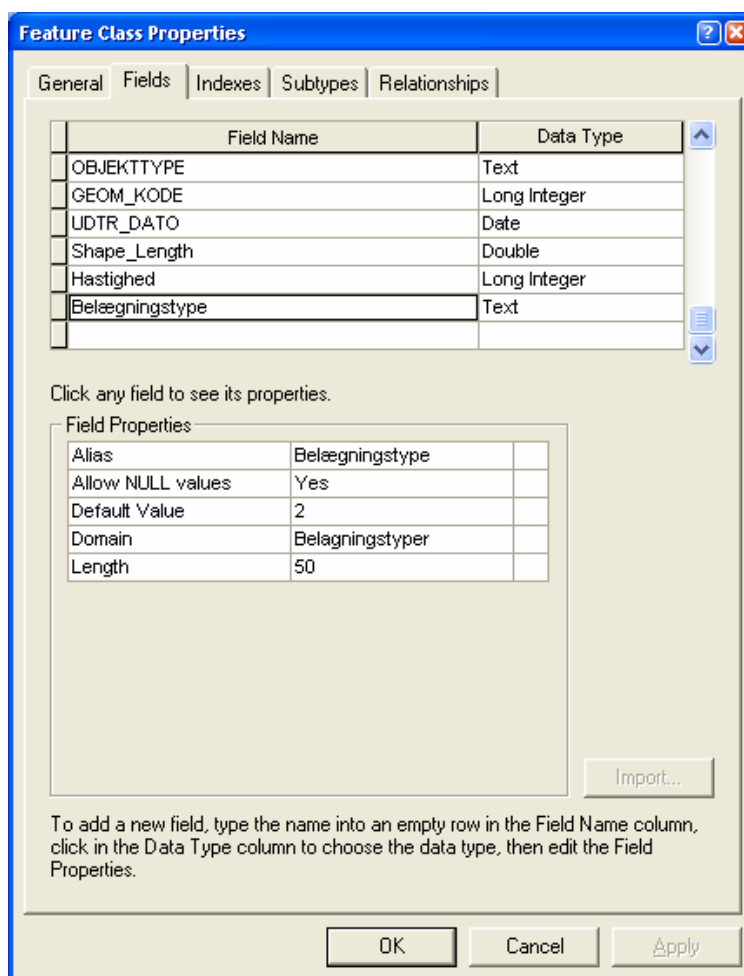
Det er som nævnt også muligt at angive et værdimængdedomæne. På figur 11 vises et eksempel på dette. Her indtastes, ud over valget af

domænetype og attribut, minimums- og maksimumsværdien for domænet. Desuden er det muligt at angive, hvad der skal ske, hvis et objekt bliver splittet op, eller hvis to objekter bliver slået sammen.



Figur 10 Værdimængdedomæne

For at tilføje et domæne til en attributværdi f.eks. i feature klassen "Vej", dobbeltklikkes der på klassen, hvorved klassens egenskaber (Feature Class Properties) vises som illustreret på figur 12. Her vælges fanen "Fields", hvor det er muligt at se, tilføje og ændre attributter og deres egenskaber. Øverst i dialogboksen ses en liste over klassens attributter, og nedenunder er den valgte attributs egenskaber vist. På figuren er attributten "Belægningstype" valgt, og herunder ses det, at det er muligt at angive et domæne. Her er domænet "Belægningstyper" valgt. Dette gør, at det kun er muligt at tilføje værdier fra domænet til attributten "Belægningstype".



Figur 11 attributter i klassen "Vej"

Dette er én måde at etablere en Geodatabase på i ArcGIS, og er en forholdsvis simpel metode. En mere kompleks metode er beskrevet i det efterfølgende Appendiks 3.

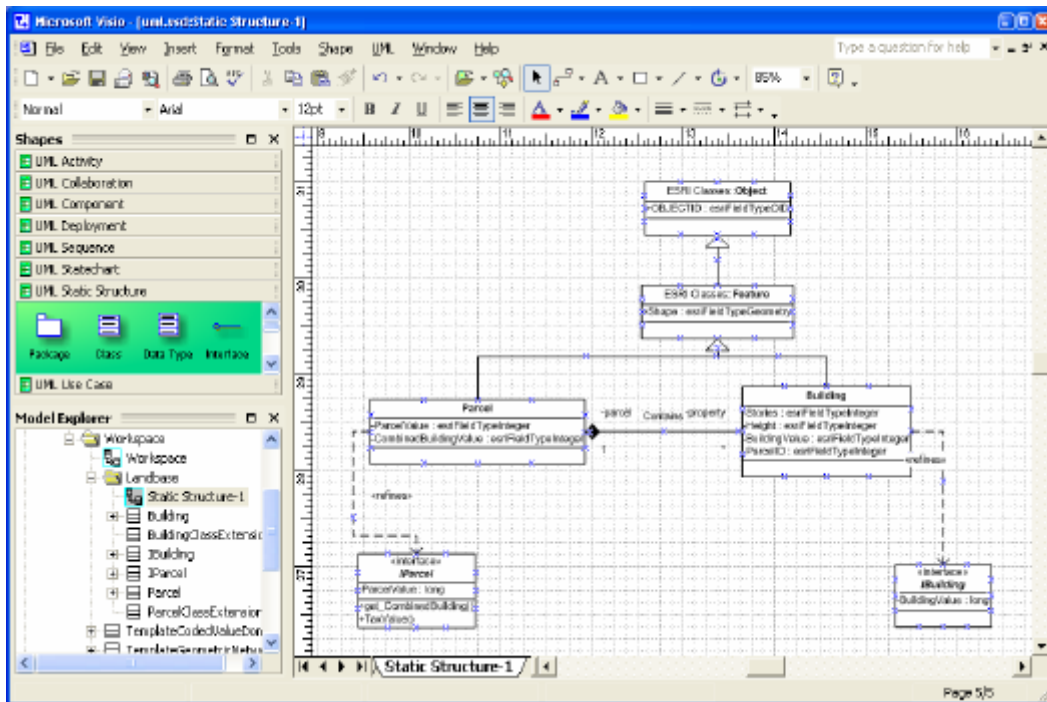
## Appendiks 3 - Konstruktion af datamodel vha. CASE-tool

I Appendiks 2 beskrives det hvordan feature datasæt og Feature klasser kan oprettes i ArcGIS. Der er også en alternativ metode til at gøre dette, hvor der benyttes et CASE-tool (Computer Aided Software Engineering). Det er et grafisk værktøj, der benyttes til at generere en objektorienteret datamodel vha. UML. Derefter er det muligt at generere kode til metoder, der giver klasserne brugerdefineret funktionalitet. Nogle af mulighederne ved denne metode illustreres i dette appendiks vha. et eksempel. Metoden samt det tilhørende eksempel bygger på de ting, der er beskrevet [ESRI, 2003b] og derfor kan de enkelte typer af regler afvige fra det, der står i resten af denne rapport.

I det følgende opbygges der et feature datasæt kaldet "Landbase", der indeholder to feature klasser, "Building" (Bygning) og "Parcel" (Matrikel). Klassen "Building" har attributterne "Stories" (etager), "Height" (højde), "BuildingValue" (bygningens værdi) og "ParcelID" (matrikel id), der alle er heltal. Klassen "Parcel" har attributterne "ParcelValue" (matrikel værdi) og "CombinedBuildingValue" (kombineret bygningens værdi), der begge er heltal. Det er meningen, at de bygninger, der digitaliseres inden for en matrikel, bliver tilknyttet matriklen, og matriklens attribut "CombinedBuildingValue" opdateres automatisk, så den angiver summen af de tilknyttede bygningens værdi. Desuden skal bygningens højde mindst være 2,5m gange antallet af etager. Denne attributregel indbygges i en funktion, der er knyttet til data.

### 3.1 Datamodel

Først opbygges modellen vha. UML i programmet Microsoft Visio. En illustration af dette program er vist på figur 1. Til venstre på figuren ses de elementer, der kan benyttes i UML-modellen (Shapes) samt en oversigt over de elementer modellen består af (Model Explorer). Til højre ses det arbejdsområde, hvor modelleringen foregår.



Figur 1 UML-model af data

Modellen opbygges ved at trække en figur, der repræsenterer en klasse, ind på arbejdsområdet, og definere dens attributter og tilknyttede funktioner. Dette gøres for alle de klasser modellen skal bestå af. Klasserne nedarver fra ESRI-klassen "Feature", der er en prædefineret klasse. På den måde får de nye klasser den samme funktionalitet som denne generelle klasse. På figur 1 ses UML-modellen af datasættet. Modellen bygger på ArcGIS' objektmodel, og er i realiteten blot en udvidelse til denne, og denne objektmodel er også repræsenteret i oversigten til venstre.

Klasserne "Parcel" og "Building" er kædet sammen af relationen "Contains", der er en komposit aggregering, med en-til-mange multiplicitet, dvs. at et objekt fra klassen "Parcel" kan være relateret til nul eller flere objekter fra klassen "Building". Denne relation er defineret som en klasse i databasen.

Da ArcGIS benytter COM-objekter til at implementere brugerdefinerede funktioner, skal de to klasser have hver deres interface. Interfaces er den eneste måde at kommunikere med klasserne på, men da de også har arvet nogle interfaces fra de objekter, klasserne har nedarvet fra, er der også andre interfaces end dem der er



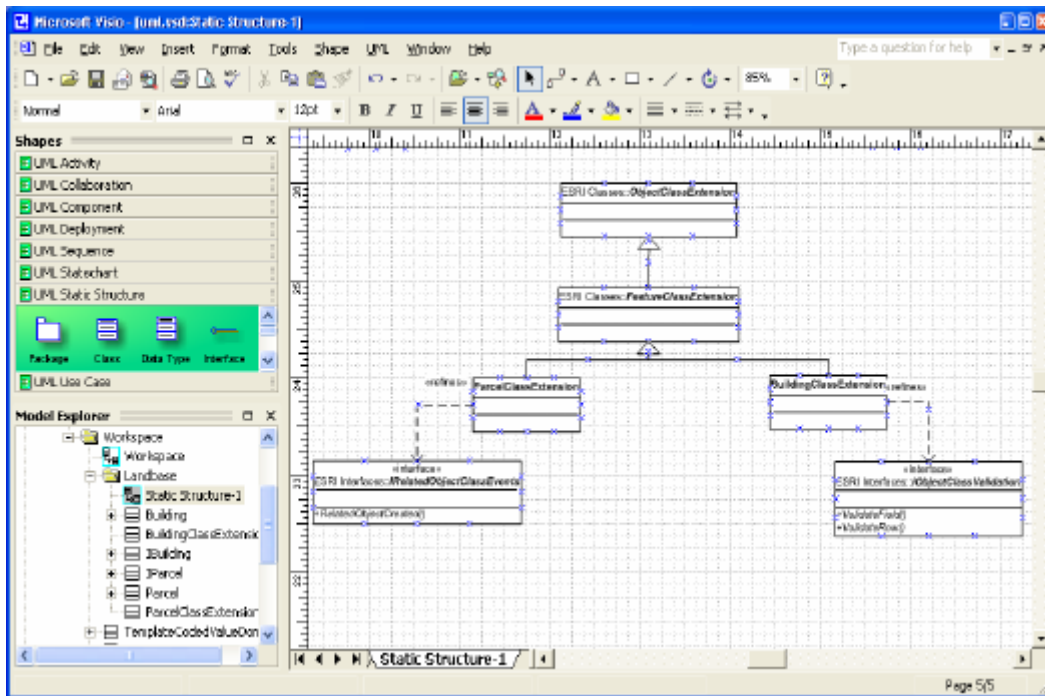
vist på figuren. For yderligere information om COM-objekter henvises der til [drgui, 2003].

I "IParcel", der er interface for "Parcel", findes funktionerne "get\_CombinedBuilding()", der beregner de tilhørende bygningers samlede værdi, når der sker ændringer, og "taxValue()", der beregner hvor meget der skal betales i skat, ud fra matriklens og bygningernes værdi. Desuden ses det, at "ParcelValue" er angivet som attribut i "IParcel". Dette gør, at det er muligt at hente og sætte værdien af denne attribut vha. interfacet.

I "Building"s interface, "IBuilding", er attributten "BuildingValue" tilgængelig, da den skal bruges til beregningen af matriklers samlede bygningsværdi.

På figur 2 ses den anden del af modellen. Denne del modellerer "Class Extensions" (klasse udvidelser), der er en del af COM-terminologien. Hvor de almindelige klasser definerer de funktioner, der bruges af det enkelte objekt. En "Class Extension" kan definere funktioner, der undersøger alle objekter i featureklassen. Der er et godt eksempel på dette i det datasæt, der beskrives her. Klassen "Parcel" har udvidelsen "ParcelClassExtension", der har interfacet "IParcelClassExtension", som implementerer funktionen "RelatedObjectCreated()". Denne funktion kaldes, når et objekt, som klassen "Parcel" relaterer til, oprettes. I dette tilfælde relaterer "Parcel" til "Building". Herefter undersøges det, om den oprettede bygning er indeholdt i en af matriklerne. Alle matriklerne søges altså igennem, og hvis der er en af dem, der indeholder den netop oprettede bygning, tilføjes der en relation mellem dem, og matriklens attribut "CombinedParcelValue" opdateres.

"BuildingClassExtention" har et interface, "IObjectClassValidation" der implementerer funktionerne "ValidateField()" og "ValidateRow()", der kaldes, når brugeren vælger at validere en bygning. Hvis brugeren gør dette, undersøger funktionerne, om højden af alle bygninger er mere end 2,5 gang så stor som antallet af etager i meter.



Figur 2 FeatureClassExtension klasse udvidelser.

Efter modellen er færdigdesignet, eksporteres den til Microsoft Repository, der er et udviklingsværktøj, der kan bruges til at lagre modeller af software. Modellen kan nu importeres i både Microsoft Visual Studio (MVS) og ArcGIS. ESRI har udviklet et værktøj til MVS, der gør det muligt at generere stubkode til modellen, så det kun er nødvendigt at skrive selve indholdet af funktionerne. Programmeringen af funktionerne i modellen gennemgås ikke her, men der henvises til [ESRI, 2003b].

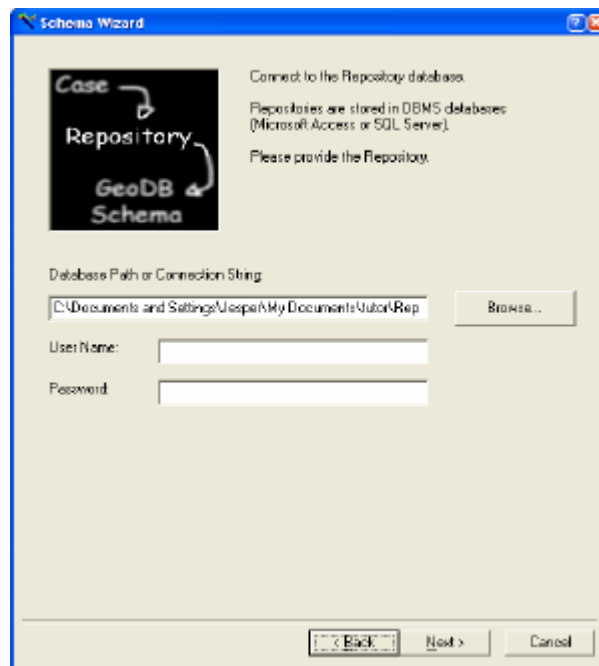
Når funktionerne er programmeret, bliver de kompileret til et COM-objekt i en .dll-fil. Denne fil bliver automatisk registreret i Windows' registreringsdatabase, og kan derfor bruges af alle programmer, heriblandt ArcGIS.

Datamodellen importeres i ArcGIS vha. et værktøj, der er illustreret på figur 3. Første skærbillede i værktøjet viser noget generel information om dette. Her trykkes der blot på "Next".



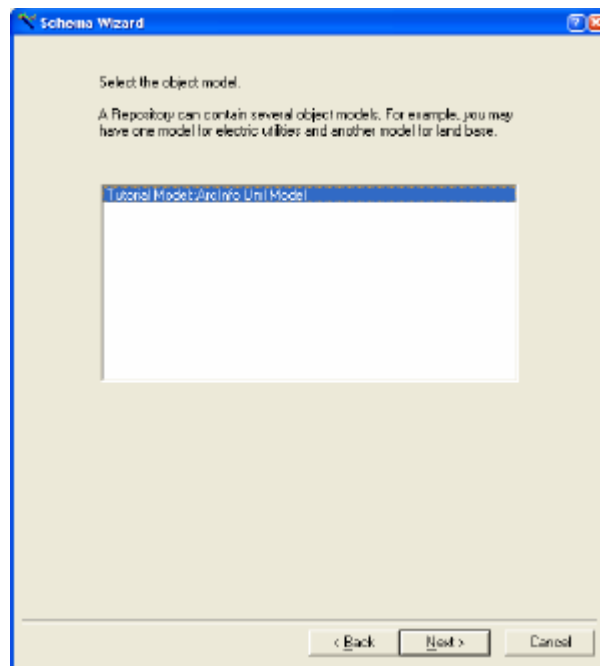
Figur 3 Schema Wizard, importerer datamodellen til ArcGIS.

På figur 4 ses det andet skærmbillede i værktøjet, hvor der forbindes til den Repository database, der indeholder modellen. Databasen kan være beskyttet af password, og derfor er det muligt at indtaste dette. Efter databasen er valgt trykkes der "Next".



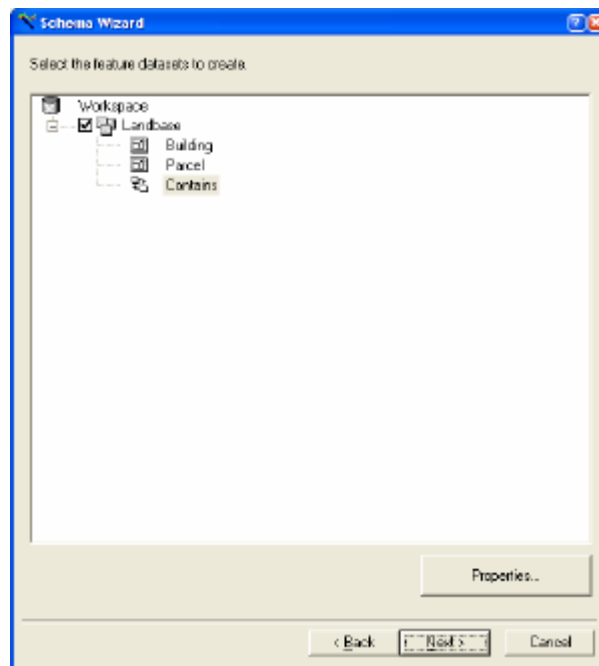
Figur 4 Andet skærmbillede i værktøjet, hvor databasen vælges.

En Repository database kan indeholde flere forskellige modeller. På tredje skærmbillede kan den ønskede model vælges. På figur 5 vises dette skærmbillede, men i dette tilfælde indeholder Repository databasen kun én model.



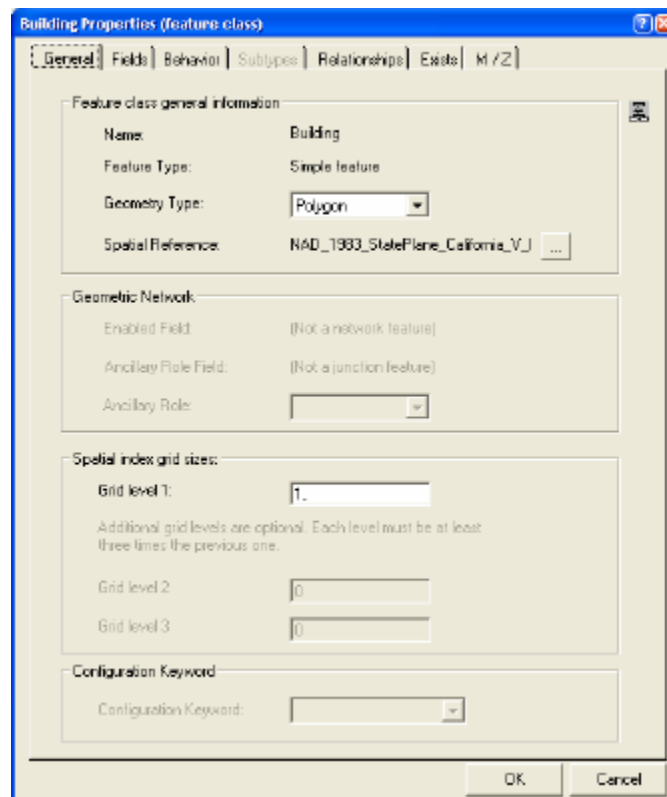
Figur 5 Tredje skærbillede i værktøjet, hvor modellen vælges.

Det næste skærbillede, der er illustreret på figur 6, viser en oversigt over modellen, hvor klasser og relationer kan ses. Denne model er simpel, og har kun to objektklasser og en relation. Ved at vælge en af objektklasserne i modellen og trykke på "Properties" kan objektklassens egenskaber ses.



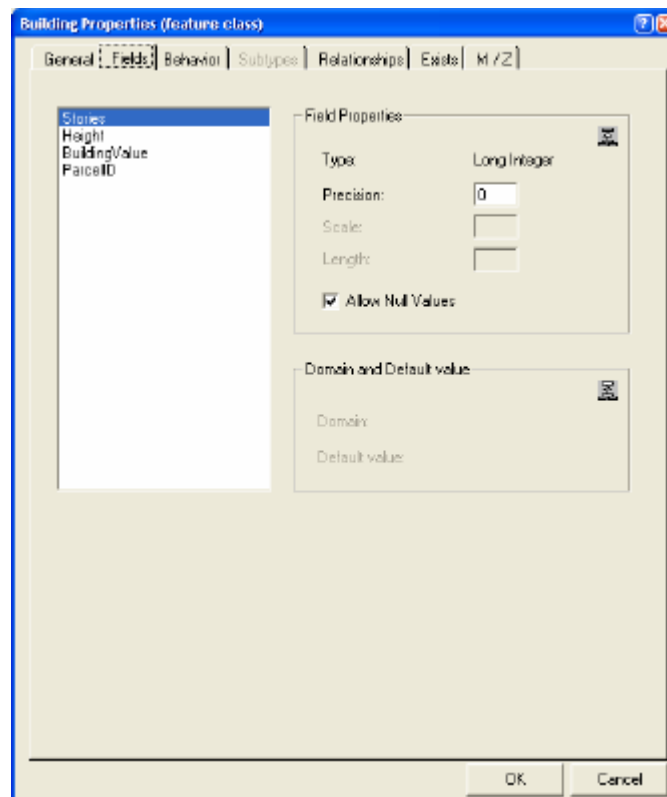
Figur 6 Det fjerde skærbillede i værktøjet der indeholder en oversigt over objektklasserne i modellen.

På figur 7 ses de generelle egenskaber for objektklassen "Building". I dette tilfælde er der tale om simpel feature, der har geometritypen "Polygon". Det er her muligt at ændre dette, samt at specificere referencekoordinatsystemet.



Figur 7 Generelle egenskaber for klassen "Building".

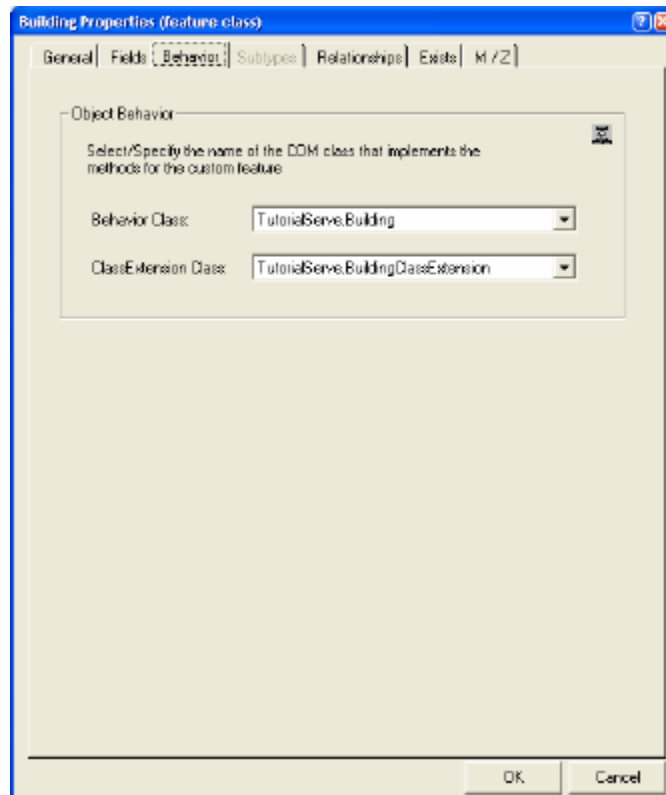
Objektklassens attributter ses under fanen "Fields". På figur 8 ses Objektklassen "Building"s attributter til venstre, og den valgte attributs egenskaber ses til højre. Det er muligt at sætte antallet af betydende cifre (Precision), og om det er tilladt at feltet ikke er tildelt en værdi (Allow Null Values).



Figur 8 Attributter for klassen "Building".

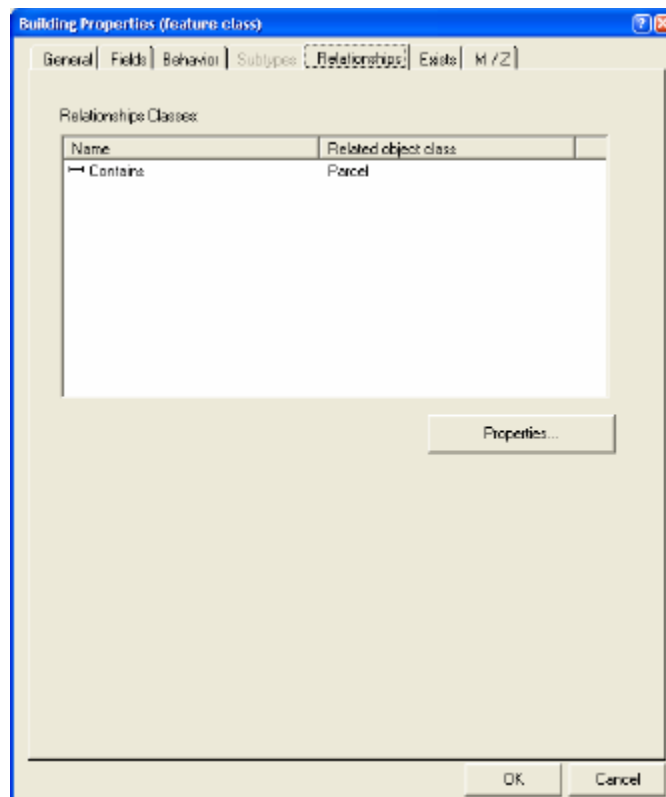
Under "Behavior" er det muligt at linke til et COM-objekt, der indeholder de funktioner, der er blevet programmeret i MVS. ArcGIS undersøger, om der er nogle funktioner knyttet til data, og om der findes et COM-objekt, der implementerer funktionerne, hvorefter klassen og ClassExtensionen blot skal vælges. Dette er illustreret på figur 9.





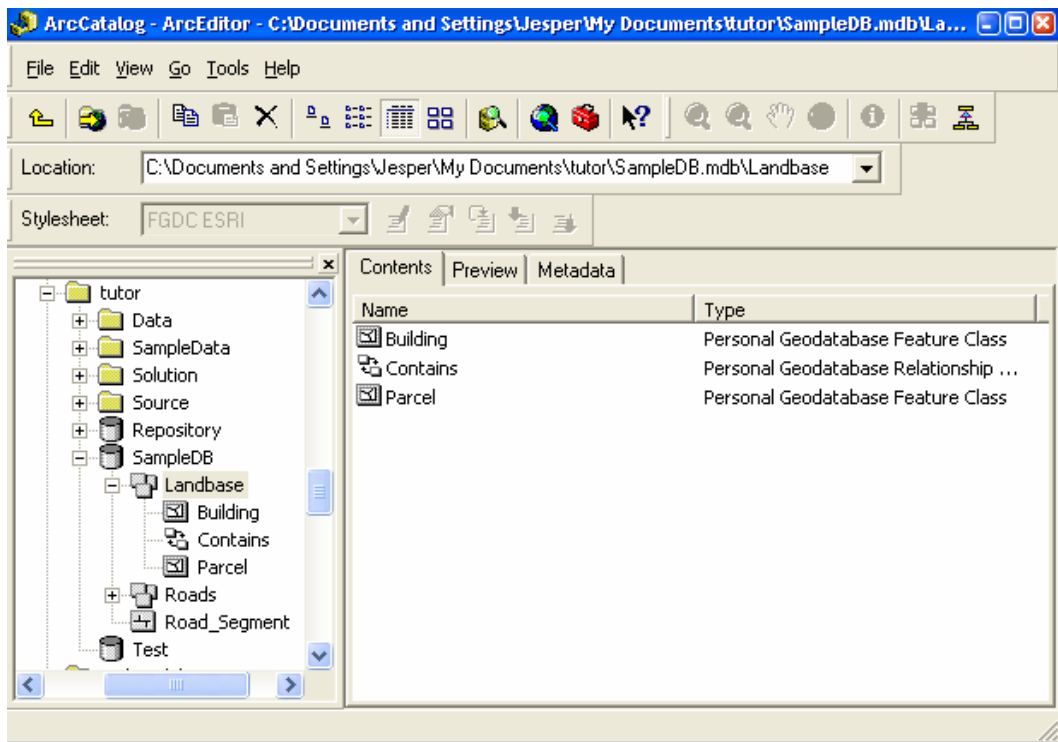
Figur 9 Objektet knyttes til et COM-objekt.

Det er desuden muligt at se hvilke relationer, objektklassen knytter, og til hvilke objekter. Klassen "Building" har relationen "Contains" til klassen "Parcel", hvilket ses på figur 10.



Figur 10 Relationer.

Efter at have undersøgt om alle klasser og relationer er defineret korrekt, afsluttes værktøjet, og ArcGIS tilføjer Feature Datasættet til den valgte Geodatabase. Herefter kan der tilføjes data til datasættet. På figur 11 ses Geodatabasen "SampleDB", der indeholder Feature Datasættet "Landbase", der igen indeholder de to Feature klasser "Building" og "Parcel", samt relationen "Contains". Det er nu muligt at tilføje data i f.eks. ArcMAP. Hvis en bygning bliver tilføjet, og den ligger inden i en matrikel, bliver de to kædet sammen af "Contains" og matriklens attribut "CombinedBuildingValue" bliver opdateret. Det er også muligt at foretage en validering af de indtastede data, og det bliver da undersøgt, om bygningernes attribut "Height" er mere end 2,5 gange så stor som "Stories". Hvis dette ikke er tilfældet bliver der vist en advarsel på skærmen.



Figur 11 Feature datasættet Landbase ses i Geodatabasen SampleDB.

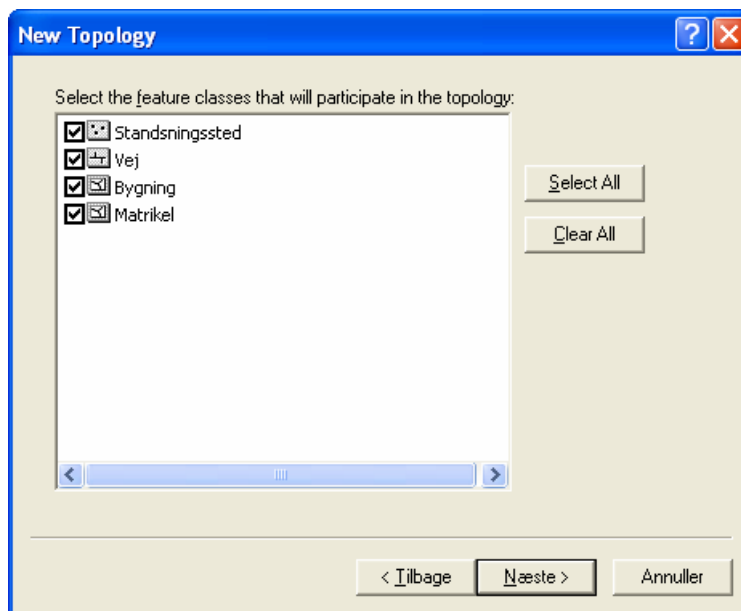
Det er altså muligt at etablere en række brugerdefinerede klasser og funktioner i ArcGIS. Dette kan være med til at sikre at indtastede data er korrekte, og at der oprettes relationer mellem objekter, der relaterer til hinanden.



## Appendiks 4 - Topologi i ArcGIS

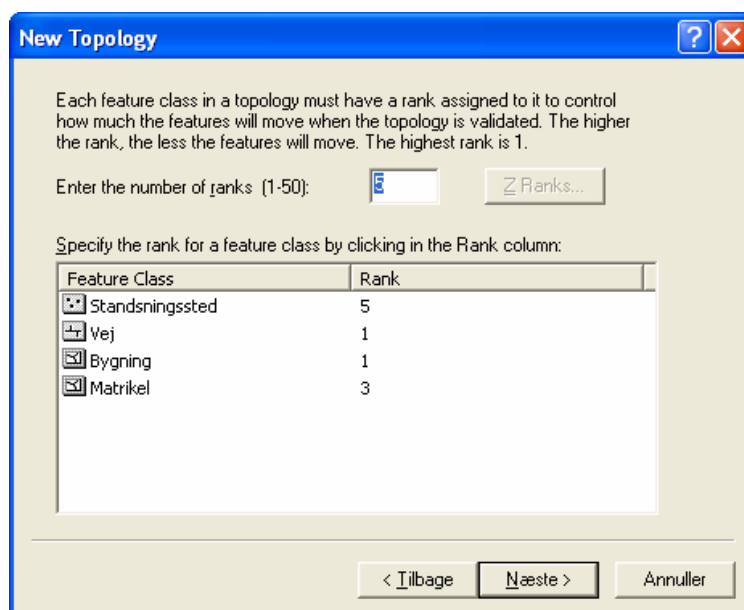
Det er muligt at opbygge et topologisk regelsæt i ArcGIS 8.3, og i dette appendiks beskrives denne proces.

Først skal selve datasættet opbygges, hvilket gøres vha. de metoder, der er beskrevet i Appendiks 2 og 3. Herefter vælges New -> Topology ved at højreklikke på datasættet i ArcCatalog. Efter en introduktionsskærm, er det muligt at udvælge hvilke klasser, der skal indgå i topologien. Alle klasserne fra datasættet bliver vist, og det er så muligt at hakke de klasser af, der skal indgå. I dette tilfælde vælges alle klasser, som illustreret på figur 1.



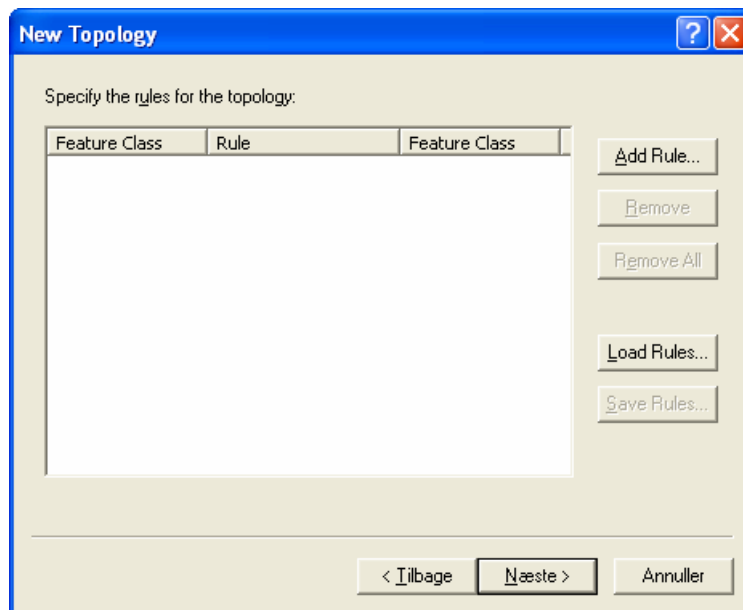
Figur 1 De klasser, som denne topologi skal omfatte.

Herefter er det muligt at vægte eller rangordne de forskellige klasser, for at kontrollere, hvilke objekter, der skal flyttes, når topologien bliver valideret. Dette er vist på figur 2. Det er muligt at definere, hvor mange forskellige range der skal være, og hvilke klasser, der skal tildeles hvilken rang.



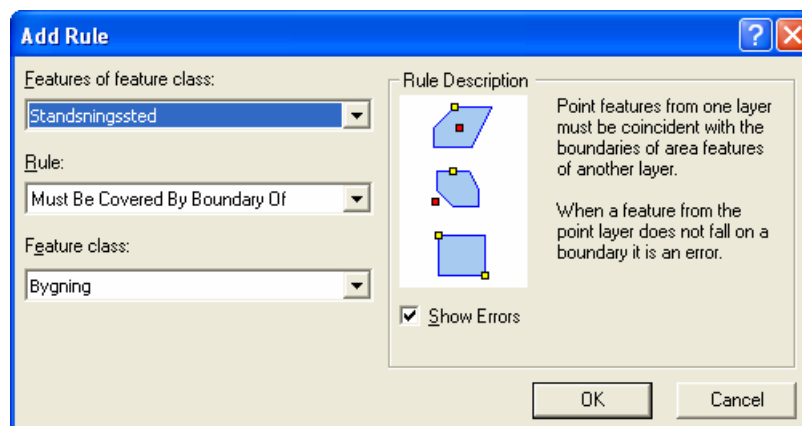
Figur 2 Rangering af klasserne.

Næste skridt er at tilføje regler til topologien. Det er muligt at hente eller gemme tidligere definerede regler vha. hhv. "Load Rules" og "Save Rules". Desuden er det muligt at tilføje en regel vha. "Add Rule", samt fjerne en eller alle oprettede regler med "Remove" og "Remove All". I vinduet i midten af dialogboksen, bliver de tilføjede regler vist. Dialogboksen, hvor dette er muligt, er illustreret på figur 3.



Figur 3 Liste med topologiske regler.

Hvis der trykkes på "Add Rule" kommer dialogen "Add Rule" frem, som vist på figur 4. I denne dialog er det muligt at vælge en feature klasse, som reglen skal baseres på. Herefter vælges den aktuelle regel fra drop-down menuen nedenunder. Til sidst vælges en eventuel relateret klasse. Til Højre vises en beskrivelse af reglen. Når reglen er defineret tilføjes den til topologilisten ved at trykke på "OK".

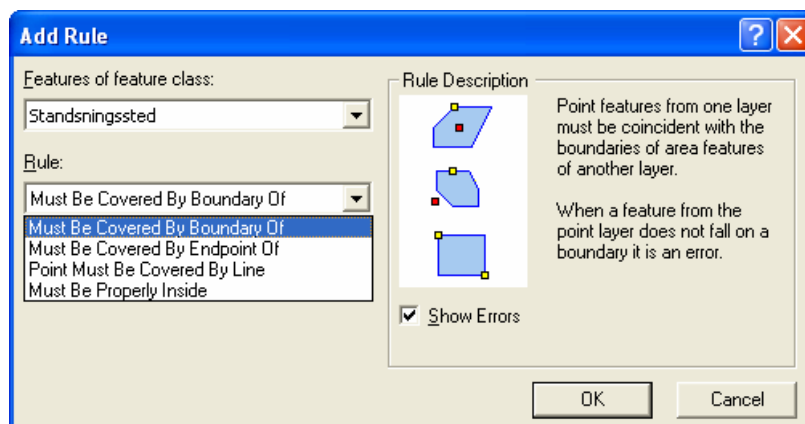


Figur 4 Tilføj en ny regel.

## 4.1 Topologiske relationer for et punkt

På figur 5 ses de mulige topologiske relationsregler for et punkt, og er beskrevet nærmere her:

- **Must Be Covered Boundary By:** Objektet skal ligge på grænsen af et objekt fra den anden klasse. Den anden klasse skal have geometrien polygon.
- **Must Be Covered Endpoint Of:** Objektet skal ligge i endepunktet af et objekt fra den anden klasse. Den anden klasse skal have geometrien linie.
- **Point Must Be Covered By Line:** Objektet skal ligge på et objekt fra den anden klasse. Den anden klasse skal have geometrien linie.
- **Must Be Properly Inside:** Objektet skal ligge inden i og ikke på kanten af et objekt fra den anden klasse. Den anden skal have geometrien polygon.



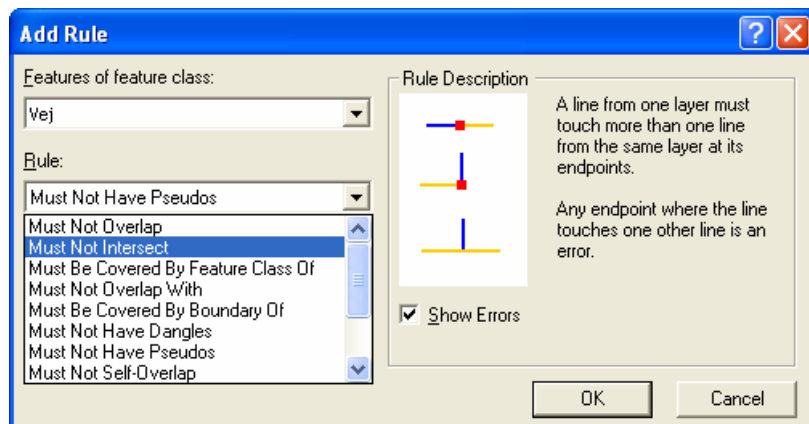
Figur 5 Topologiske relationer for et punkt.

## 4.2 Topologiske relationer for en linie

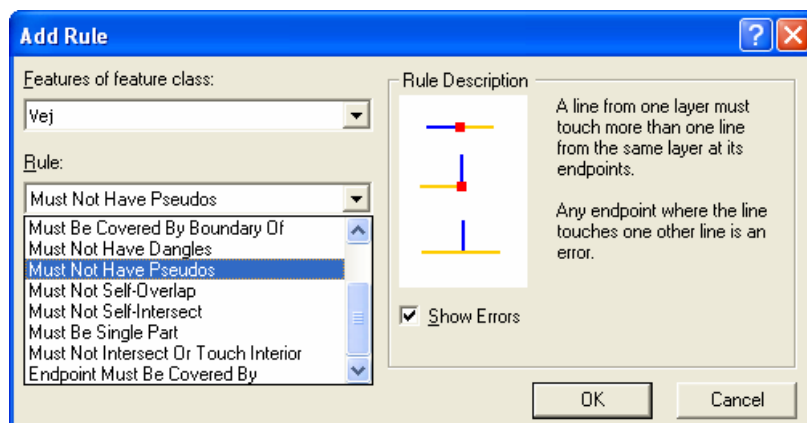
På figur 6 og 7 ses de mulige topologiske relationer for objekter fra klassen Vej, der har geometrien linie. En liste med disse relationer kan ses her:



- Must Not Overlap: Objekter indenfor den valgte klasse må ikke overlappe hinanden.
- Must Not Intersect: Objekter indenfor den valgte klasse må ikke skære hinanden.
- Must Be Covered By Feature Class Of: Objekter fra den første klasse skal ligge inden i objekter fra den anden. Begge klasser skal have geometrien linie.
- Must Not Overlap With: Objekter fra den første klasse må ikke overlappe objekter fra den anden. Begge klasser skal have geometrien linie.
- Must Be Covered By Boundary Of: Objektet skal ligge på kanten af den anden klasses geometri. Den anden klasses geometri kan kun være polygon.
- Must Not Have Dangles: Objekter fra klassen, skal i begge endepunkter røre andre objekter fra denne klasse.
- Must Not Have Pseudos: Objekter fra klassen skal i begge endepunkter røre mindst to andre objekter fra samme klasse.
- Must Not Self-Overlap: Objekter fra klassen må ikke overlappe sig selv.
- Must Not Self-Intersect: Objekter fra klassen må ikke skære sig selv.
- Must Be Single Part: Objekter fra klassen må kun bestå af en del (må ikke have en kompleks geometri).
- Must Not Intersect Or Touch Interior: Objekter fra klassen må ikke skære eller overlappe det indre af andre objekter fra klassen.
- Endpoint Must Be Covered By: Endepunkter for objekter fra klassen, skal være sammenfaldende med et punkt fra den anden klasse.



Figur 6 Topologiske relationer for en linie (1)



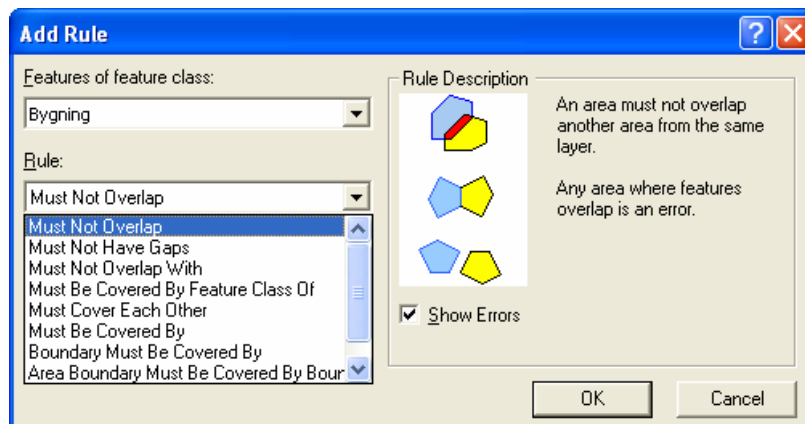
Figur 7 Topologiske relationer for en linie (2)

### 4.3 Topologiske relationer for en polygon

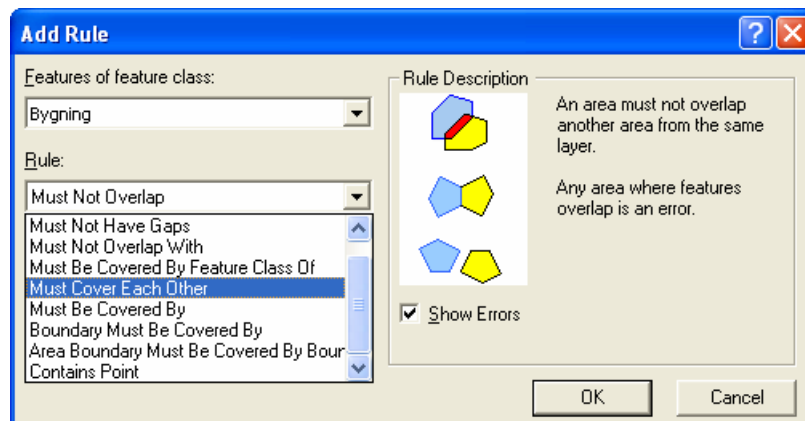
For objekter med geometritypen Polygon, som f.eks. Bygning er der følgende mulige topologiske relationsregler, som ses på figur 8 og 9:

- **Must Not Overlap:** Objekter indenfor den valgte klasse må ikke overlappe hinanden.
- **Must Not Have Gaps:** Der må ikke være mellemrum eller huller mellem objekter fra klassen.

- **Must Not Overlap With:** Må ikke overlappe objekter fra den anden klasse. Den anden klasse skal have geometrien polygon.
- **Must Be Covered By Feature Class Of:** Arealet af objekter fra den første klasse skal dække objekter fra den anden klasse. Den anden klasse skal have geometrien polygon.
- **Must Cover Each Other:** Objekterne fra de to klasser skal dække hinanden. Den anden klasse skal have geometrien polygon.
- **Must Be Covered By:** Arealet af objekter fra den første klasse skal være dækket af objekter fra den anden klasse. Den anden klasse skal have geometrien polygon.
- **Boundary Must Be Covered By:** Objektets grænse skal være dækket af det andet objekt. Det andet objekt skal have geometrien linie.
- **Area Boundary Must Be Covered By Boundary:** Objektets grænse skal være dækket af grænsen for objekter fra den anden klasse. Den anden klasse skal have geometrien polygon.
- **Contains Point:** Objektet fra klassen skal indeholde et punkt fra den anden klasse. Objekter fra den anden klasse skal have geometrien punkt.



Figur 8 Topologiske relationer for en polygon (1)



Figur 9 Topologiske relationer for en polygon (2)

Disse topologiske relationer er simple at oprette og bruge, og det er muligt at identificere en del forskellige fejl vha. dem. Der er dog en række begrænsninger, der gør at det ikke er muligt at bestemme alle relevante typer relationer.