

# SYSTEMARKITEKTUR FOR ROMLIG-TEMPORAL DATABASE MED INFORMASJON OM JORDBRUKSAREALER OG DYRKINGSJORD I NORGE

SYSTEM ARCHITECTURE FOR SPATIO-TEMPORAL DATABASE WITH  
INFORMATION ON CULTIVATED AND CULTIVABLE LAND IN NORWAY

INGVILD NYSTUEN

UNIVERSITETET FOR MILJØ- OG BIOVITENSKAP  
INSTITUTT FOR MATEMATISKE REALFAG OG TEKNOLOGI  
MASTEROPPGAVE 30 STP. 2011





# Forord

## Valg av oppgave

Jeg har lenge vært fascinert av tid i databaser, eller rettere sagt manglende tid i databaser. Hos min arbeidsgiver Norsk institutt for skog og landskap (Skog og landskap) har vi lenge tenkt på at vi burde håndtere historikk, og dermed endringer, i våre geografiske databaser på en bedre måte. Jeg innser nå, etter å ha brukt noen månedersverk på å studere tid i databaser, at vi også har mye å hente på hvordan tid håndteres i våre ikke-romlige databaser.

Jeg har et datateknisk perspektiv på oppgaven. Andre innfallsvinkler kunne for eksempel vært bruk av fjernmålingsteknikker, modellering av arealdekkedata, klassifikasjonssystemer for arealdekkedata, eller estimering av usikkerhet i endringsanalyser basert på arealdekkedata. Fra før har jeg utdanning i informatikk innenfor databehandling, med blant annet databaseteori og algoritmer. Jeg har arbeidet i en del år ved Skog og landskap med forvaltning og distribusjon av geodata. Min primære interesse er i skjæringspunktet mellom informatikk og geografisk informasjonsvitenskap, med fokus på bruk av datateknologi for å løse oppgaver knyttet til kart og geodata. Det er derfor naturlig å avgrense oppgaven til å lage et informasjonssystem der en romlig-temporal database står i sentrum. Klassifikasjonssystemer for arealdekke og sofistikerte fjernmålingsteknikker er ikke et tema. Jeg forholder meg til det klassifikasjonssystemet, de data og de rutiner for oppdatering av data som allerede finnes.

Litteraturstudier har tatt en del tid og krefter. Det er skrevet og gjort mye innenfor databaser og databasemodellering for romlige, temporale og romlig-temporale data i flere tiår, samtidig som det har gått tregt å få satt teorien om romlig-temporale databaser ut i livet utenfor akademien. Jeg ønsket å skaffe meg en oversikt over status på området. Mye av teorien lar seg ikke implementere ved hjelp av dagens allment tilgjengelige teknologi uten stor egeninnsats. Forskningen ligger langt forut for teknologien. Jeg mener likevel det er viktig å følge med på forskningen, både for å kunne plukke ut og benytte aktuelle elementer og for å kunne påvirke programvareleverandørene når de skal velge hvilke elementer de skal implementere.

Arbeidet med oppgaven har gitt meg mulighet til å fordype meg i et emne, både vitenskapelig, men også mer praktisk ved å gjøre eksperimenter og se hvilke utfordringer som oppstår. Jeg gleder meg til å virkelig sette ut i live min nyervervede kunnskap om romlig-temporale databaser og endringshåndtering.

## Takksigelser

Det er mange som må takkes. Særlig mine to veiledere, Håvard Tveite (førsteamanuensis ved UMB) og Knut Bjørkelo (sjefingeniør ved Skog og landskap). Uten veiledere, ingen masteroppgave.

En stor takk til også til min arbeidsgiver, og min nærmeste overordnede Hildegunn Norheim i særdeleshet, som lot meg få bruke en del arbeidstid på et masterstudium i geomatikk.

Kolleger ved Skog og landskap har bidratt med verdifull innsikt og gode diskusjoner. Geir-Harald Strand oppmuntret meg i årevis til å komme i gang med masterstudiene. Han har også vært en god diskusjonspartner underveis. Vidar Svendsen har bidratt med sin enorme innsikt i forvaltningsrutinene rundt datasettet AR5. Mange flere har også bidratt med sin kunnskap.

Til vanlig er jeg seksjonsleder for ni særdeles dyktige og produktive medarbeidere ved Skog og landskap. De har vist stor forståelse og medfølelse for en til tider svært (både fysisk og psykisk) fraværende seksjonsleder. Dessuten har de motivert meg til å stå på når jeg har vært lei og sliten. Tusen takk!

Til slutt må jeg takke min tålmodige ektemann som har lagd mat, vasket hus og handlet julegaver mens jeg har sittet med nesa i bøker, artikler og tastatur. Helger og kvelder i stor grad stått i de romlige-temporale databasers tegn de siste månedene.

Langhus, 11. desember 2011

Ingvild Nystuen

# Sammendrag

Formålet med denne oppgaven er å skissere et informasjonssystem for å følge utviklingen av tilgjengelig jordbruksjord i Norge. Oppgaven er avgrenset til å lage en romlig-temporal datamodell og database for et nytt datasett som jeg kaller TilgjengeligJord. TilgjengeligJord inneholder alle områder som enten er, eller har vært, jordbruksareal eller dyrkingsjord i Digitalt markslagskart (DMK) eller jordbruksareal i AR5 (AR5). Informasjonssystemet skal oppdateres årlig fra ajourført AR5. Det skal kunne svare på en rekke skisserte brukstilfeller.

Teorien om romlige-temporale databaser går flere tiår tilbake i tid. Det finnes internasjonalt anerkjente standarder på området. Allikevel er det mer unntaket enn regelen at romlige databaser er temporale. Slik mister vi viktig informasjon.

Det finnes flere modeller for temporale relasjonsdatabaser. Øyeblikksbilder er den enkleste, og versjoner hele relasjonen ved å ta en kopi. Kopien dokumenterer relasjonen slik den så ut på kopitidspunktet. Sekvensielle kopier representerer utvikling. Tupler kan også versjoneres. Ved å innføre en bitemporal modell med to ortogonale tidsakser, vil også retting av feil kunne ivaretas på en dokumenterbar måte.

Modeller for romlige data deles gjerne inn i feltmodeller og objektmodeller. DMK, AR5 og TilgjengeligJord er såkalte klassifiserte arealdekker av type diskrete polygondekker. Slike data faller ikke naturlig inn i en av modellene, men er en slags mellomting som kan modelleres som begge deler.

Det inngår ulike datoer som attributter i DMK og AR5. Disse datoene er viktige i informasjonssystemet. Er de ikke brukt etter intensjonen, vil det skape problemer med temporale gap og overlapp i mer avanserte temporale modeller enn øyeblikksbilder. Det er en del feil i bruk av datoer i DMK og AR5.

Tre ulike romlig-temporale modeller for TilgjengeligJord skisseres: øyeblikksbilder, rom-tid sammensetning og objektmodellen. De to førstnevnte testes ut på et lite testdatasett for deler av Hobøl kommune. Testen gjøres i PostGIS med simple feature-geometrimodell. Oppdatering av datasettet ga en del utfordringer med geometriproblemer. Neste versjon av PostGIS kommer med ny funksjonalitet og ny topologimodell, og bør testes ut.

Øyeblikksbildene er den enkleste modellen å etablere og vedlikeholde, men klarer ikke svare på alle brukstilfellene. Rom-tid sammensetning og objektmodellen svarer godt på brukstilfellene, men lider under temporale gap grunnet feil bruk av datoer i AR5 og DMK. I tillegg er de mer komplekse å etablere og vedlikeholde. Gitt dagens kvalitet på datoer i AR5 og DMK, er det øyeblikksbilder som er mest hensiktsmessig. Imidlertid vil informasjonssystemet gi mer relevant informasjon om utviklingen av jordbruksjord dersom man klarer å kvalitetssikre dato-egenskapene og bygger informasjonssystemet i en mer avansert romlig-temporal modell.

# Abstract

The objective of this thesis is to outline an information system being able to follow the development of cultivated and cultivable land in Norway. The task is limited to making a spatio-temporal data model and database for a new dataset named «TilgjengeligJord» (available farm land). The new dataset is composed from all areas that are, or have ever been, cultivated or cultivable land in Digitalt markslagskart (DMK) or cultivated land in AR5, and is to be updated annually from updated AR5. A range of use cases for the information system is presented.

The literature on spatial and temporal databases goes several decades back in time. International standards on spatio-temporal modelling exist. Still, we are losing valuable information by not making relevant spatial databases temporal.

Temporal relational databases can be modelled in several ways. Snapshots are the simplest model. A snapshot is a copy of the whole relation. It documents the relation at the time of the copy. Sequent snapshots will represent the development. Tuples can also be versioned. By introducing a bitemporal model with orthogonal time lines, corrections of errors in the data can be documented for the future.

Spatial modelling and models are divided into two main groups: field models and object models. DMK, AR5 and the new dataset are categorical coverages or discrete surface coverages. They are not easily placed in either the field model or the object model.

AR5 and DMK have date attributes that are important parts of the information system. I found quite a few examples where these dates are not used as intended, which again will cause problems with temporal gaps and overlaps in more advanced spatio-temporal models.

Three different spatio-temporal models for the new dataset is outlined: snapshots, space-time composite and the object model. The snapshot model and the space-time composite model are tested on a small test area in the municipality of Hobøl. The test is implemented in PostGIS with the simple feature geometry type. The actual updating of the data set in PostGIS caused quite a few geometry problems. The next version of PostGIS comes with new functionality and a topology model, and should be tested.

The snapshot model is the easiest to establish and maintain, but does not support all the use cases. The space-time composite and the object model support the use cases, but suffer from temporal gaps due to wrong use of date attributes in AR5 and DMK. They are also more complex to establish and maintain. Given the actual quality of the date attributes in AR5 and DMK, the snapshot model is the most appropriate. However, the information system will give more valuable information on the development of available farm land if it is made with a more advanced spatio-temporal model.

# Innhold

<b>1</b>	<b>Innledning</b>	<b>1</b>
1.1	Problemformulering . . . . .	1
1.2	Hvorfor et slikt informasjonssystem? . . . . .	1
1.2.1	Hvorfor ta med tidsdimensjonen? . . . . .	2
1.3	Hvorfor nå og ikke for fem år siden eller om fem år? . . . . .	3
1.4	Oppbygging av oppgaven . . . . .	4
<b>2</b>	<b>Definisjoner</b>	<b>6</b>
2.1	Ord fra fagområdet markslag . . . . .	6
2.2	Ord fra fagområdene romlige og temporale databaser . . . . .	7
<b>3</b>	<b>Markslagskartets historie</b>	<b>9</b>
3.1	Økonomisk kartverk . . . . .	9
3.2	Markslaget digitaliseres – DMK oppstår . . . . .	11
3.3	DMK forenkles – AR5 skapes . . . . .	13
3.4	Systematisk ajourføring av AR5 blir en realitet . . . . .	14
<b>4</b>	<b>Databasesystemer</b>	<b>15</b>
4.1	Databasemodeller . . . . .	15
4.2	Relasjonsdatabaser . . . . .	15
4.3	Objektdatabaser . . . . .	17
4.4	Objekt-relasjonsdatabaser . . . . .	17
4.5	Bruksområder og transaksjonstyper . . . . .	17
<b>5</b>	<b>Geografiske databaser</b>	<b>20</b>
5.1	Konseptuell modellering av geografiske fenomener . . . . .	20
5.2	Lagring av geografiske data . . . . .	21
5.3	Representasjon av vektorobjekter . . . . .	21
5.4	Usikkerhet i geometrisk og temporal avgrensning og fastsetting av egen- skaper . . . . .	22
5.5	Spøringer . . . . .	22
<b>6</b>	<b>Temporale databaser og informasjonssystemer</b>	<b>24</b>
6.1	Tidsbegreper . . . . .	24
6.2	Tid i relasjonsdatabasen . . . . .	25
6.2.1	Bitemporale data i relasjonsdatabasen . . . . .	26
6.3	Tid i objektdatabaser . . . . .	27

<b>7</b>	<b>Romlig-temporale databaser og temporal GIS</b>	<b>28</b>
7.1	Sintons rammeverk . . . . .	29
7.2	Operasjoner i et geografisk informasjonssystem . . . . .	30
7.3	Lagring . . . . .	30
7.4	Analyse . . . . .	31
7.5	Oppdateringer . . . . .	31
7.6	Modeller for romlig-temporal informasjon . . . . .	32
7.7	Multimedia-data . . . . .	33
7.8	Egenskaper ved arealdekke-data i et romlig-temporalt perspektiv . . . . .	33
<b>8</b>	<b>OGC- og ISO-standarder</b>	<b>35</b>
8.1	ISO 19125 Geographic information – Simple feature access . . . . .	35
8.2	NS-EN ISO 19108 Geografisk informasjon - Modell for å beskrive tids- aspekter . . . . .	36
8.2.1	Temporale geometriske og topologiske objekter . . . . .	36
8.2.2	Tid og komponenter i geografisk informasjon . . . . .	38
8.3	Andre aktuelle standarder . . . . .	39
<b>9</b>	<b>Tilgjengelig teknologi</b>	<b>40</b>
9.1	Geodatabasesystemer i bruk ved Skog og landskap . . . . .	40
9.2	Desktop-GIS og datamodelleringsverktøy . . . . .	41
9.3	Støtte for tid i tilgjengelige databasesystemer . . . . .	41
9.3.1	Oracle . . . . .	41
9.3.2	PostgreSQL/PostGIS . . . . .	42
9.3.3	ArcSDE . . . . .	42
9.4	Teknologi valgt i denne oppgaven . . . . .	43
<b>10</b>	<b>Tilgjengelige databaser og datasett</b>	<b>44</b>
10.1	Økonomisk kartverk . . . . .	44
10.2	Digitalt Markslagskart . . . . .	44
10.2.1	Arealtilstand . . . . .	44
10.2.2	Jordklassifisering . . . . .	45
10.2.3	Temakartet dyrkingsjord skal inngå i TilgjengeligJord . . . . .	49
10.2.4	Datoangivelsenes kvalitet i DMK-databasen . . . . .	51
10.2.5	Generell kvalitet i DMK-databasen . . . . .	52
10.3	AR5 . . . . .	52
10.3.1	Ajourføring av AR5 . . . . .	54
10.3.2	Datoangivelsenes kvalitet i AR5-databasen . . . . .	55
10.3.3	Generell kvalitet i AR5-databasen . . . . .	57
10.4	Årsversjoner av AR5 . . . . .	57
10.5	Endringsanalyse . . . . .	57
10.6	Testdata for masteroppgaven . . . . .	61
10.6.1	Testdata satt inn i Sintons modell . . . . .	62
<b>11</b>	<b>Klassifisering av endringer i geodata</b>	<b>63</b>
11.1	Klassifisering av endringer . . . . .	63



11.2	Hva gjør andre? . . . . .	65
<b>12</b>	<b>Problemstillinger rundt håndtering av historikk i AR5 og DMK</b>	<b>66</b>
12.1	Kontinuerlig og periodisk ajourføring . . . . .	66
12.2	Identifisere ulike typer endringer . . . . .	66
12.3	Mer korrekt og konsistent bruk av datafangstdato og verifiseringsdato .	67
12.4	Datoer oppdateres ikke eller tilbakestilles . . . . .	69
12.5	Konsekvenser av gale datoer i DMK og AR5 . . . . .	73
12.6	Bruk av verifiseringsdato som «assertion time» . . . . .	73
<b>13</b>	<b>Brukstilfeller</b>	<b>74</b>
<b>14</b>	<b>Systemarkitektur for ny database for TilgjengeligJord</b>	<b>76</b>
14.1	Datamodell for TilgjengeligJord – Et nytt datasett laget fra DMK og AR5 . . . . .	76
14.1.1	TilgjengeligJord skal inneholde både jordbruksareal og dyrkingsjord . . . . .	77
14.1.2	Markslagesegenskaper i TilgjengeligJord . . . . .	77
14.1.3	Konseptuell modell og lagringsmodell . . . . .	80
14.1.4	TilgjengeligJord skal håndtere tid/historikk . . . . .	80
14.1.5	Prinsipper for lagring og oppdatering av TilgjengeligJord . . . . .	81
14.1.6	TilgjengeligJord kan implementeres med ulike romlig-temporale modeller . . . . .	82
14.2	TilgjengeligJord som øyeblikksbilder . . . . .	83
14.2.1	Arbeidets gang . . . . .	84
14.2.2	Problemer underveis . . . . .	86
14.2.3	Mulig forbedringer . . . . .	87
14.3	TilgjengeligJord som rom-tid sammensetning . . . . .	87
14.3.1	Arbeidets gang . . . . .	88
14.3.2	Skranker som burde vært implementert . . . . .	90
14.3.3	Fordeler, ulemper og problemstillinger . . . . .	90
14.3.4	Mulige forbedringer . . . . .	91
14.4	TilgjengeligJord som objekter . . . . .	92
14.5	De ulike brukstilfellene og øyeblikksbilder . . . . .	94
14.6	De ulike brukstilfellene og rom-tid sammensetning . . . . .	96
14.7	De ulike brukstilfellene og objektmodellen . . . . .	98
14.8	API og views . . . . .	98
<b>15</b>	<b>Konklusjoner</b>	<b>99</b>
<b>16</b>	<b>Videre arbeid</b>	<b>101</b>
16.1	Arbeid for Skog og landskap for å realisere informasjonssystemet . . . . .	101
16.2	Generelt . . . . .	102
	<b>Referanseliste</b>	<b>104</b>

Liste over figurer	109
Liste over tabeller	110
Vedlegg	111

# 1 Innledning

## 1.1 Problemformulering

Problemformuleringen for denne oppgaven er: Hvordan utvikle en systemarkitektur for forvaltning av geoinformasjon om jordbruksarealer og dyrkingsjord som understøtter dagens og den nærmeste fremtidens behov for informasjon om tilgangen til, og utviklingen av, tilgjengelig jordbruksjord i Norge.

Oppgaven avgrenses til å finne en hensiktsmessig romlig-temporal datamodell og databasestruktur for et nytt datasett basert på allerede eksisterende data, samt det å lage spørringer mot disse dataene i henhold til definerte brukstilfeller. I tillegg foreslås noen mulige forbedringer av de eksisterende dataene og forvaltningsregimet rundt disse for bedre å understøtte tidsdimensjonen. Det antydes noe om mulige videre forbedringer av det nye datasettet ved å ta i bruk andre eksisterende datakilder og annen metodikk i ajourføringen.

## 1.2 Hvorfor et slikt informasjonssystem?

Gjennom Jordlova [26], har jordbruksarealer, og arealer som kan settes i stand til å bli fulldyrka jord (dyrkingsjord), et vern. Slike arealer skal ikke omdisponeres til andre formål uten særskilt grunn. Omdisponering kan bare skje i henhold til Jordlova eller Plan og bygningsloven. Det er et uttalt politisk mål å styrke jordvernet i Norge [58]. Målet er at det skal omdisponeres maksimalt 5700 dekar i året. I følge tall fra Statistisk sentralbyrå for 2010, omdisponeres det mindre dyrka mark enn på 30 år, men målet er fortsatt ikke nådd. Det er også i de viktigste landbruksområdene at mesteparten av omdisponeringa skjer. Over halvparten av det omdisponerte jordbruksarealet befinner seg i de 20 kommunene som omdisponerer mest. Dette er tall fra KOSTRA (KOMmune-STat-RApportering, et informasjonssystem som driftes av Statistisk sentralbyrå). Tallene baserer seg på kommunenes egen rapportering, og er tall som hentes fra kommunenes reguleringsplaner.

Det er planlagt omdisponering som registreres i KOSTRA-statistikken, ikke hva som faktisk skjer og har skjedd. I praksis betyr omdisponeringen at arealer blir nedbygd, enten med næringseiendom, bolighus eller i samferdselsøyemed. Dette er informasjon som, etterhvert som tiltakene gjennomføres, registreres i ulike kartverk. Noen av disse kartene blir oppdatert regelmessig (det er i hvert fall intensjonen, i praksis kan det dessverre hende oppdateringen av kart ikke er helt som ønsket). Informasjon om jordbruksarealer og dyrkingsjord, er også å finne på kart. Et kartbasert system for å følge med på utviklingen av tilgjengelig jordbruksjord i Norge, vil kunne gi svar på andre spørsmål enn KOSTRA klarer. Et kartbasert system vil kunne gi mer detaljerte svar på hvor ting skjer enn i hvilken kommune. I motsetning til KOSTRA-statistikken, er

kartinformasjonen koordinatfestet. Kart er et godt visuelt verktøy, og kan gjøre det mulig å se geografiske mønstre som vi ikke kan se i statistikken. Et kartbasert system vil også være basert på hva som faktisk skjer i landskapet, ikke hva som er planlagt. KOSTRA-tall og kartbasert informasjon vil utfylle hverandre.

I tillegg vil et kartbasert system kunne gi presis nasjonal arealstatistikk dersom det bygger på nasjonale datasett som er tilstrekkelig oppdaterte. Det kartbaserte systemet vil også være et godt grunnlag for arealplanlegging, og et nyttig verktøy for tilskuddsfordeling i landbruket. Kart i tilskuddssammenheng i landbruket er ingen nyskaping i Norge. Gårdskart er allerede etablert for de aller fleste landbrukseiendommer i Norge, og arealtall herfra benyttes ved fordeling av arealtilskudd for å sikre en mer rettferdig fordeling av midlene. Gjennom gårdskartetableringen har man hatt en betydelig kvalitetsforbedring av både eidsgrensene for landbrukseiendommer og informasjonen om jordbruksarealene i kartet AR5. Oppdaterte AR5-data skal framover sikres gjennom et ajourholdsregime der både kommunene og Skog og landskap gjør sin del.

Den kartfestede informasjonen om dyrkingsjorda er ikke ivaretatt like godt som for jordbruksarealet. Informasjonen finnes, men den er ikke oppdatert siden den ble etablert. Det kan være inntil 40 år siden. Riktig nok er dette ganske statisk informasjon, men den påvirkes av nedbygging, oppdyrking og andre forhold (som en elv som finner seg nytt løp, eller en innsjø som demmes opp). Mye av endringene vil kunne fanges opp gjennom et administrativt ajourhold basert på andre datasett som inneholder jevnlig oppdatert informasjon om bebygd areal, samferdsel, vann og jordbruksarealer.

Informasjon om jordbruksarealer og dyrkingsjord, må sees i sammenheng. Denne oppgaven tar derfor for seg begge deler. For geoinformasjon om jordbruksarealene, finnes allerede et rimelig godt forvaltningsregime gjennom ajourholdet av datasettet AR5 og lagring av historiske AR5-data etterhvert som ajourføringen skjer. AR5 er et heldekkende datasett som også inneholder informasjon om samferdselsarealer, bebygd areal og vann. Informasjon om dyrkingsjord finnes i datasettet Digitalt markslagskart (DMK), som ikke ajourføres. I DMK finnes også eldre informasjon om jordbruksarealer. Denne oppgaven tar sikte på å lage et nytt datasett for tilgjengelig jordbruksjord som skal inneholde jordbruksarealer fra AR5 og jordbruksarealer og dyrkingsjord fra DMK. Datasettet skal ajourføres administrativt fra AR5 en gang i året. For å kunne svare på spørsmål om utviklingen av den tilgjengelige jorda, må det nye datasettet ikke bare ha en romlig dimensjon, men også en temporal.

Det nye arealressursdatasettet gi navnet «TilgjengeligJord».

### 1.2.1 Hvorfor ta med tidsdimensjonen?

For å kunne følge med på utvikling av fenomener eller objekter med geografisk plassering og utstrekning, må man ta vare på historikken etter hvert som dataene oppdateres.

Tidligere ble informasjon lagret i manuelle arkiver der man la til informasjon (et nytt papirark) når noe nytt hendte. Med data-alderens inntreden, ble det vanlig å overskrive data i databasene når vi fikk ny informasjon. De tidligste databasene tok rett og slett dårligere vare på historien enn de manuelle arkivene. Selv om historien gjerne var tilgjengelig i backup og transaksjonslogger, var den vanskelig tilgjengelig. Det tok i beste fall mye tid å gjenskape informasjonen, i verste fall klarte man ikke

gjenskape den i det hele tatt.

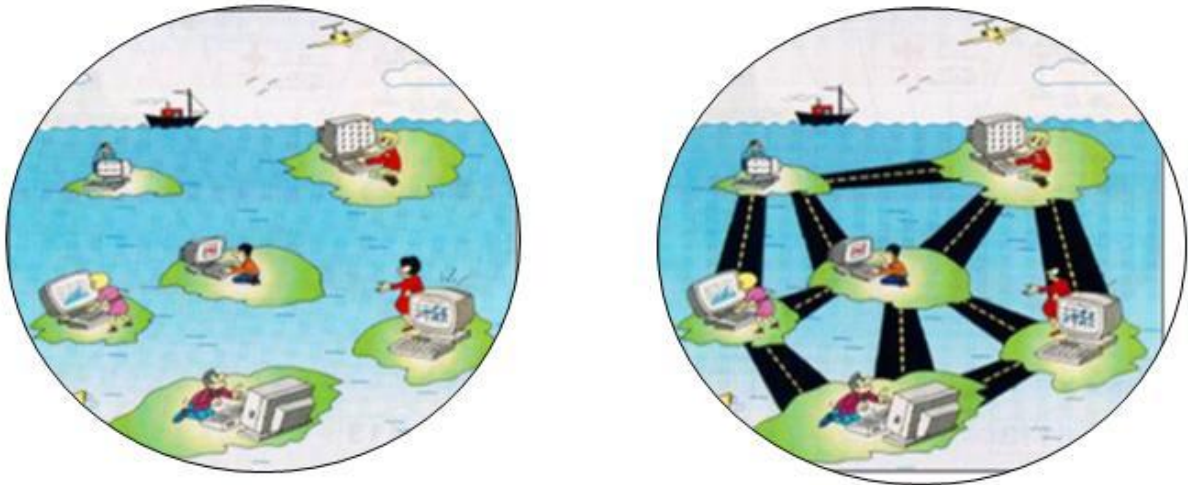
Geografiske databaser kan ta mye plass. Databaser som tar vare på historien, kan også ta mye plass. Geografiske databaser som lagrer endringer over tid, tar enda mer plass. Det har vært en viktig årsak til at geografiske databaser tradisjonelt ikke har håndtert tid. Etter hvert som prisen på disk har gått ned, er ikke selve lagringen av store mengder data noe problem. Imidlertid kompliserer tidsdimensjonen både innlegging, oppdatering og uthenting av informasjon. Datamodellen blir rett og slett ganske kompleks. Jeg antar at det er den viktigste årsaken til at mange geografiske databaser som burde håndtere tid, fortsatt ikke gjør det.

### **1.3 Hvorfor nå og ikke for fem år siden eller om fem år?**

Nå er vi sju milliarder mennesker på jorda. Alle trenger mat. Klimaendringer tvinger fram tiltak som igjen gir endringer i landbruket. Vi dyrker ikke lenger bare mat og dyrefôr, men også råvarer til miljøvennlig biodrivstoff. Presset på arealene øker. Samtidig har vi hatt en teknologieksplosjon innen geografisk informasjonsvitenskap (GIScience) som gir oss nye muligheter. Mens det for få år siden var viktig å spare diskplass, kan man i dag kjøpe billige terrabyte-disker på nærmeste elektronikkforretning. Derimot har vi ikke lenger tålmodighet til å vente på at IT-avdelingen skal gjenskape historiske data fra logg og backup, vi vil ha dataene lett tilgjengelig. PC-en med Internett-tilkobling er blitt allemannseie, Google og Microsoft har gjort web-kart tilgjengelig for alle. Myndighetene har i større og større grad sett betydningen og nytteverdien av oppdatert og tilgjengelig geoinformasjon i form av kart og statistikk.

Fra det første GIS-verktøy så dagens lys i Canada på starten av 1970-tallet, har den skjedd en rivende utvikling. De aller første GIS-programvarepakke ble først og fremst brukt av myndigheter, militæret og universiteter. Både programvare og datastrukturer var proprietære, og de kjørte som selvstendige applikasjoner i batch på en stormaskin. Fra starten av 80-tallet til midten av 90-tallet kom det flere brukere og bruksområder til, programvaren ble i stand til å håndtere topologi, den ble kommersialisert og kunne kjøres på PC-er og arbeidsstasjoner. Fra midten av 90-tallet har fokus vært på geografisk infrastruktur og tjenester. Med Internett, har det blitt mulig å utveksle informasjon i sann tid mellom distribuerte databaser. Det har satt nye krav til standardiserte og stabile tjenester, metadata og harmonisering av dataspesifikasjoner. Interoperabilitet på alle nivåer har blitt en kritisk faktor. For å løse dette, har GIS-miljøet i større og større grad tatt i bruk generelle IT-konsepser som WebServices, datamodellering, databasesystemer, tjenestekataloger og flerlags arkitektur. Det er nå mulig å bruke geografiske data om hverandre i ulike verktøy uten å måtte konvertere data til et nytt format for hvert verktøy. Generelle programvarebiblioteker i generelle programmeringsspråk som Java, håndterer geografiske data og operasjoner på disse.

Databaseløsningene for geografiske data, geodatabasesystemene, har også utviklet seg. De tidligste variantene håndterte geometri i en datastruktur, og egenskaper i en annen. Geometrien ble ofte håndtert i proprietære strukturer, mens egenskapene ble lagret i tabeller i en relasjonsdatabase. Med objekt-relasjonsdatabaser, kan geometri



Figur 1.1: Fra hver mann på sin øy til geografisk infrastruktur. Kilde: ukjent

og egenskaper håndteres samlet, geometrien er nå logisk sett kun en ekstra egenskap ved en forekomst (rad) i en databasetabell. De fleste av geodatabasene har også implementert operasjoner og spørringer på geodata i spørrespråket SQL.

Gjennom samarbeidet Norge digitalt, initiert gjennom Stortingsmeldingen med samme navn [37], har en i Norge bygget opp en standardisert og forutsigbar tilgang til geodata gjennom enighet om teknologiske standarder for informasjonsflyt, metoder for dokumentasjon og avtaler om tilgjengelighet og ytelse. Norge har fått sin geografisk infrastruktur (figur 1.1). Dermed er det blitt mulig å kombinere data fra ulike dataleverandører i sanntid eller ved automatiske nedlastingsløsninger. Det er ikke lenger nødvendig å bruke veldig store ressurser på innhenting av data fra ulike leverandører.

Det er grunn til å mistenke at måten tid og historikk håndteres på i AR5 har et forbedringspotensiale. Jo forttere vi identifiserer eventuelle nødvendige endringer som trengs, jo raskere får vi et datasett som egner seg best mulig til å understøtte det informasjonssystemet vi ønsker oss.

Sett i sammenheng med etterspørselen etter kunnskap om utviklingen av tilgjengelig jord i Norge, er det på høy tid å sette i gang. Det er bakgrunnen for valget av masteroppgave.

## 1.4 Oppbygging av oppgaven

Kapittel 2 går gjennom aktuell terminologi og definerer en del begreper innen fagområdene markslag og romlig-temporale databaser. Kapittel 3 går gjennom markslagskartets historie fra økonomisk kartverk til dagens ajourførte AR5-data. I de påfølgende fire kapitlene (kapittel 4–7) omhandles databasesystemer, romlige og geografiske databaser, temporale databaser og romlig-temporale databaser. Kapittel 8 omhandler aktuelle OGC- og ISO-standarder. I kapittel 9 og 10 omtales henholdsvis teknologi som er tilgjengelig ved Skog og landskap og brukt i denne oppgaven, og tilgjengelige datasett. Klassifikasjon av ulike typer endringer i geodata er tema for kapittel 11, mens

det i kapittel 12 er en gjennomgang av ulike problemstillinger knyttet til historikk i AR5. Kapittel 13 oppsummerer ulike definerte brukstilfeller for informasjonssystemet. I kapittel 14 testes brukstilfellene på ulike modeller for en romlig-temporale database for det nye datasettet «TilgjengeligJord». Gjennomgangen er delvis praktisk og delvis teoretisk. Til slutt diskuteres løsningene i kapittel 15, før kapittel 16 kommer med forslag til videre arbeid. Både konkrete forslag til Skog og landskap og mer vitenskapelig videreføring diskuteres kort.

Deler av kapittel 4 og kapittel 5 er omarbeidet og tilpasset temaet for denne masteroppgaven fra en mindre oppgave jeg skrev ved UMB våren 2011 om transaksjonsbasert WFS. I kapittel 4 gjelder det stoffet om transaksjonshåndtering og samtidighetsstyring, i kapittel 5 innledningen og stoffet om romlige indekser.

## 2 Definisjoner

### 2.1 Ord fra fagområdet markslag

**AR5:** Klassifikasjonssystem og kartdatasett for arealressurser i Norge. AR5 er en forenkling av Digitalt markslagskart, og bygger på klassifikasjonssystemet for markslag i økonomisk kartverk. Inngår i FKB [5].

**Digitalt markslagskart (DMK):** Klassifikasjonssystem og kartdatasett for informasjon om markslag. Tidligere en del av FKB.

**Driftsforhold i jordbruket:** Klassifisering av jordbruksareal og dyrkingsjord etter vilkår for maskinell drift. Tar hensyn til topografi, arrondering og størrelse. Klassene for driftsforhold, er lettbrukt, mindre lettbrukt og tungbrukt. På dyrkingsjord blir bare lettbrukt og mindre lettbrukt benyttet [21].

**Dyrka jord/mark:** Synonym for jordbruksareal

**Dyrka myr:** Jordbruksareal med minst 20 cm tykt mold- eller torvlag [21].

**Dyrkbar jord/mark:** Synonym for dyrkingsjord.

**Dyrkingsjord:** Følgende definisjon ble brukt opprinnelig: «Areal som ved oppdyrking kan settes i slik stand at de vil holde kravene til lettbrukt eller mindre lettbrukt fulldyrka jord, og som holder kravene til klima og jordkvalitet for plantedyrking [21]». Fordi driftsforhold i jordbruket ikke er videreført i AR5, må det vurderes å omdefinere dyrkingsjord til: «Areal som ved oppdyrking kan settes i slik stand at de vil holde kravene til fulldyrka jord».

**Felles kartdatabase (FKB):** En samling strukturerte datasett som utgjør en viktig del av grunnkartet for et område [5].

**Jordbruksareal:** Sterkt kulturpåvirket mark med permanent vegetasjonsdekke (eng/beite) eller periodisk vegetasjondekke (åker). Fellesbetegnelse for fulldyrka jord, overflatedyrka jord og innmarksbeite [5].

**Fulldyrka jord:** Jordbruksareal som er dyrka til vanlig pløyedyp, og som kan brukes til åkervekster eller eng som kan fornyes ved pløying [21].

**Gjødsla beite:** Tidligere betegnelse på «innmarksbeite».

**Innmarksbeite:** Jordbruksareal som kan benyttes som beite, men som ikke kan høstes maskinelt. Minst 50 % av arealet skal være dekt av grasarter eller beitetålende urter [5]. Ble tidligere kalt «gjødsla beite» [21].



**Lettbrukt areal:** Klasse for driftsforhold i jordbruket. Areal med slik størrelse, ar-rondering og topografi (helling mindre enn ca 1:5) at firehjulstraktor med vanlige redskaper kan nyttes på rasjonell måte [21].

**Marksflag:** Areal som er relativt ensartet for plantedyrking og naturlig planteproduksjon. I sammenheng med markslagsklassifikasjon i økonomisk kartverk brukes begrepet om areal som er relativt ensartet for jordbruk og skogbruksproduksjon [21].

**Mindre lettbrukt areal:** Klasse for driftsforhold i jordbruket. Areal som ikke holder kravene til lettbrukt areal, men maskinell høsting er mulig med mindre traktor (helling mindre enn ca 1:3) [21].

**Overflatedyrka jord:** Jordbruksareal som for det meste er rydda og jevnet i overflata, slik at maskinell høsting er mulig [21].

**Tilgjengelig jord:** Arealer som er tilgjengelige for jordbruk. Omfatter jordbruksarealer og dyrkingsjord.

**Tungbrukt areal:** Klasse for driftsforhold i jordbruket. Fulldyrka jord som ikke holder kravene til lettbrukt eller mindre lettbrukt areal [21].

## 2.2 Ord fra fagområdene romlige og temporale databaser

**Categorical coverage:** Klassifisert arealdekke på engelsk.

**Diskret polygondekke:** Se klassifisert arealdekke. Tilsvarende «discrete surface coverage» i ISO 19123.

**Feature:** Abstraksjon av et fenomen i den virkelige verden. En feature representeres ved et objekt i en objektorientert modell.

**Geografisk objekt:** Brukes både om et feature med romlig utstrekning, eller om et polygon, en kurve eller et punkt i en geografisk database. Sistnevnte sammenfaller ikke med definisjonen av objekt i objektorientert teori.

**Klassifisert arealdekke:** Heldekkende, ikke-overlappende polygondata. Et klassifisert arealdekke gir vektorisert stedsbasert informasjon, og er en blanding av feltmodellen og objektmodellen [24]. Engelsk betegnelse: Categorical coverage. NB: Ordet arealdekke brukt i denne sammenhengen, må ikke blandes sammen med arealdekke brukt som norsk oversettelse av det engelske «land cover».

**Rom-tid sammensetning:** Vektordatasett med polygoner der enhver endring får den endrede delen av et polygon til å løsrive seg fra det opprinnelige polygonet slik at det opprinnelige polygonet blir erstattet av to mindre polygoner. Alle polygoner har sitt livsløp, og ingenting blir slettet fra datasettet. Hva som er tidligere og nåværende polygoner, ser man fra tidstempler [24]. Engelsk betegnelse: Space-time composite.

**Snapshot:** Øyeblikksbilde på engelsk.

**Space-time composite:** Rom-tid sammensetning på engelsk.

**Øyeblikksbilde:** Et datasetts tilstand ved et gitt tidspunkt. En samling sekvensielle øyeblikksbilder kan si noe om dynamiske fenomen/utvikling over tid [41]. Engelsk betegnelse: Snapshot.

Artikkelen *A Consensus Glossary of Temporal Database Concepts* [19] er en god kilde til temporal database-terminologi.

## 3 Markslagskartets historie

Et markslag defineres som areal som er relativt ensartet for jordbruk og skogproduksjon, og er en inndeling av landarealet etter arealtilstand, arealbruk, driftsforhold og produksjonsevne [21, 60]. Markslag ble opprinnelig kartlagt som en del av økonomisk kartverk. Neste skritt var å digitalise markslagskartet. Det digitale produktet fikk navnet Digitalt markslagskart (DMK). Deretter ble modellen kraftig forenklet under navnet AR5. For å ikke forvitre, må geoinformasjon ajourføres (oppdateres) regelmessig. AR5 er underlagt et regime for ajourføring for å sikre et oppdatert datasett til nytte i ulike sammenhenger.

### 3.1 Økonomisk kartverk

Økonomisk kartverk (ØK), er detaljerte, analoge kart for områder som ble vurdert å ha økonomisk interesse. Kartet ble utgitt trykket på papir, og inneholdt eiendomsdata, høydekurver, arealinformasjon (markslag), veier, bygninger og andre tekniske anlegg. I praksis var det landarealet opp til tregrensa som ble kartlagt.

Markslag, inkludert informasjon om jordbruksarealer og dyrkingsjord, ble kartlagt gjennom markslagsklassifikasjonen i Økonomisk kartverk. Jordbruksareal er en fellesbetegnelse for fulldyrka jord, overflatedyrka jord og innmarksbeite (tidligere kalt gjødsla beite). Jordbruksareal blir også noe upresist benevnt dyrka jord og dyrka mark. Upresist fordi det ikke er naturlig å tenke på innmarksbeite er en del av dyrka jord. Begrepet «dyrka jord» vil gjerne oppfattes som et areal som plantes til og høstes maskinelt.

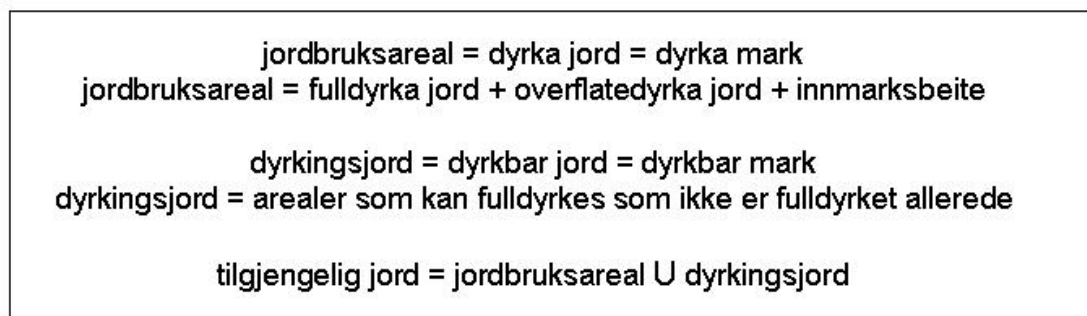
Dyrkingsjord blir i ulike sammenhenger også kalt dyrkbar jord eller dyrkbar mark. Ulik terminologi skaper forvirring. I denne oppgaven velger jeg å følge markslagsinstruksen [21] og bruke betegnelse jordbruksareal og dyrkingsjord. I tillegg innføres begrepet «tilgjengelig jord» som unionen av jordbruksareal og dyrkingsjord.

Dyrkingsjord er definert slik i markslagsinstruksen [21]:

«areal som ved oppdyrking kan setjast i slik stand at det vil halde krava til lettbrukt eller mindre lettbrukt fulldyrka jord [...], og som held krava til klima og jordkvalitet for plantedyrking.»

I stortingsmeldinga Norge digitalt [37] er ØK-kartlegginga omtalt slik:

«Allerede i 1805 ble detaljert oppmåling av arealressursene påbegynt (økonomisk kartlegging), men stoppet opp i 1814. Det tok deretter nesten 150 år før planmessig økonomisk kartlegging kom i gang igjen. Første utgave av økonomisk kartverk for hele landet var først på plass i juni 2002.»



Figur 3.1: Terminologien er forvirrende

I figur 3.1, gjør jeg bevisst en liten omdefinering av dyrkingsjord til å være all jord som kan fulldyrkes, uavhengig av om den vil være lettbrukt, mindre lettbrukt eller tungbrukt.

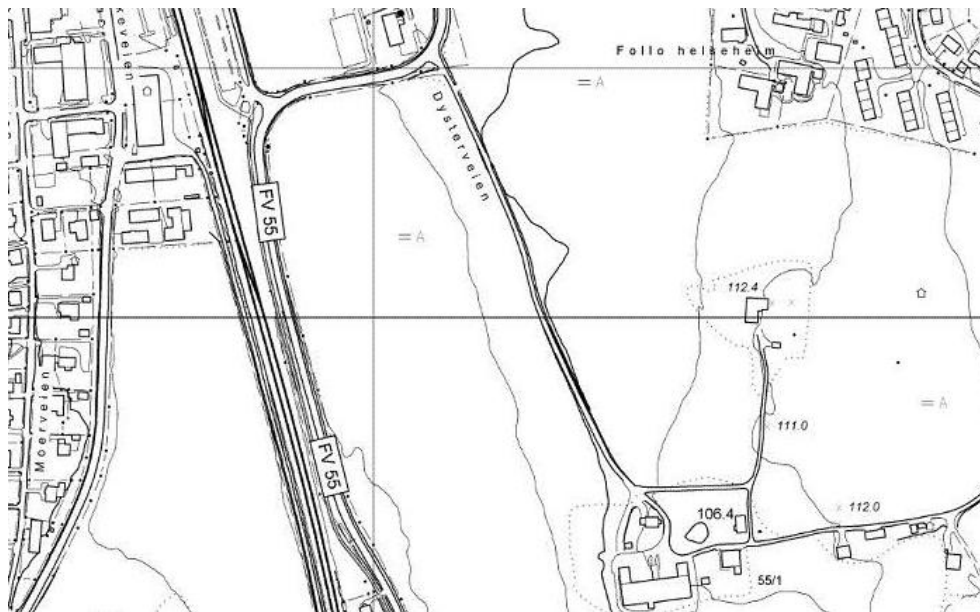
I boka *Kartlegginga av Noregs grønne gull* [43], gir Kristen Øyen en omfattende oppsummering av historien bak markslagskartlegginga.

I løpet av 1800-tallet fikk de fleste europeiske land register med oversikt over landbruksressursene sine (jordregister). Allerede i 1805 ble dette vurdert i Norge, og til og med så vidt igangsatt før det ble stoppet av økonomiske grunner. Utover 1800-tallet og første halvdel av 1900-tallet ble det gjennomført mange utredninger og skrevet mange rapporter om økonomisk kartverk og jordregister for Norge, men ingen av forslagene ble gjennomført. Det ble blant annet foreslått at et jordregister for Norge skulle opprettes på grunnlag av det økonomiske kartverket, men også at dersom ingen økonomisk kartlegging ble satt i gang, skulle et jordregister lages med basis i det allerede tilgjengelige materialet.

I 1955 ble Jordregisterutvalget oppnevnt, og deres framlegg fikk konsekvenser. Det ble i statsbudsjettet for 1959-60 satt av penger til å starte prøveregistrering med tanke på å utvikle et jordregister fra Norge. En viktig beslutning som ble tatt i prøveperioden, var å lage et enhetlig klassifikasjonssystem for landarealet (markslagsklassifikasjonssystem) for hele landet. Ikke minst for dyrkingsjord var dette krevende, da det var (og er) stor forskjell på hva som regnes som dyrkingsjord i bratte Vestlands-kommuner og flate bygder på Østlandet. Løsningen ble å definere to klasser for dyrkingsjord, lettbrukt (gruppe A) og mindre lettbrukt (gruppe B).

Inndeling av det jorddekte arealet i markslag etter bestemte kjennetegn skulle, sammen med eiendomsgrenser, danne utgangspunktet for arealopp gavene i jordregisteret. Jordregister-arbeidet var et stort nybrottsarbeid, som startet med prøveperioden fra 1959 til 1964. I prøveperioden ble klassifikasjonssystemet fastsatt, og metoder for flybildetolkning, kartfremstilling og utregning av arealer ble utviklet.

En stortingsmelding ble lagt fram med basis i dette prøvearbeidet, Stortingsmelding nr. 74 (1964-65): *Om prøvearbeidet med jordregisteret og økonomisk kartverk*. Stortingsmeldinga pekte ut retningen for Jordregisterets framtid. Litt tidligere hadde det, gjennom Landsplan for økonomisk kartverk, blitt klart at markslagsklassifisering med tilstrekkelig detaljeringsgrad for å lage jordregister, skulle inngå i ØK, sammen med eiendomsgrenser. Videre var det i Landsplanen bestemt at alt produktivt areal skulle



Figur 3.2: Økonomisk kartverk for et område nær Ås sentrum

kartlegges i målestokk 1:5000, mens resten av arealet opp til tregrensa skulle kartlegges i målestokk 1:10000. Den beslutningen la grunnlaget for markslagsregistering av alle landets kommuner. Heldigvis kom man fram til en detaljeringsgrad på markslagsklassifiseringen i ØK som også var god nok for jordregister. Slik sparte man både penger og tid.

Det var daværende Jorddirektoratets avdeling for jordregistrering, som hadde hatt ansvaret for utviklingsprosjektet og prøveperioden, som også fikk oppgaven med å kartlegge markslaget i ØK, et omfattende kartleggingsprogram med utstrakt bruk av feltarbeid. Den viktigste kilden til informasjon om hvordan jobben ble utført, er *Markslagsklassifisering i økonomisk kartverk* [21], som beskriver prinsipper, metode, symboler og definisjoner for kartlegginga. Denne har blitt revidert i flere omganger, siste versjon kom ut i 2007 [4].

Arbeidet med markslagsklassifisering pågikk i flere tiår, og var et nitidig og ressurskrevende arbeid. Øyen [43] nevner spesielt kartleggingen av dyrkingsjord som krevende, både på myr og fastmark. Jordbor og spade var nødvendige redskaper for å kunne fastsette et areals egnethet for oppdyrking. På fastmark skulle blant annet stein og blokkinnhold, behov for kunstig drenering og tørkeutsatthet vurderes. På myr skulle dybde, omdanningsgrad og vegetasjonstype vurderes. Kartlegging av dyrkingsjord var et prioritert arbeid som ble utført etter beste evne og skjønn. Informasjonen om dyrkingsjord i ØK må betraktes som svært verdifull, og vel verdt å ta vare på og foredle.

### 3.2 Markslaget digitaliseres – DMK oppstår

Økonomisk kartverk er et analogt kartprodukt. Det betyr at innholdet i kartet henger tett sammen med selve presentasjonen. Etter hvert som informasjonsteknologien

gjorde sitt inntog de siste par tiårene i forrige århundre, så man behovet for å gjøre markslagsinformasjonen mer digital. I digitale kart er innhold og presentasjon adskilt. Man lagrer innholdet i en geografiske database, og kan velge å presentere den samme informasjonen på ulike måter i ulike produkter. På det viset er det enklere å etterkomme ulike behov.

På begynnelsen av 1990-tallet startet daværende Norsk institutt for jord- og skogkartlegging (NIJOS) arbeidet med å digitalisere markslagsinformasjonen i ØK. Opprinnelig ble det gjort for å forbedre jordregisteret, den kommunevise oversikten over arealfordelingen til landbrukseiendommer. Imidlertid så man ganske snart hvilket potensiale det digitale markslagskartet (DMK) hadde som et generelt datasett, blant annet innen arealforvaltning og -planlegging.

Etableringen av DMK pågikk over mer enn ti år i en periode med rivende teknologisk utvikling. I starten av etableringsperioden benyttet man i stor grad flybilder og rissefolier fra ØK som utgangspunkt for digitaliseringen. I 1994 begynte en å benytte scannede rissefolier, og mot slutten av perioden var det i hovedsak ortofoto som ble benyttet i stedet for flybilder. Ortofoto har den store fordelen at de er målestokkriktige, slik at de kunne legges bakenfor de scannede foliene på en dataskjerm. Det lettet arbeidet betraktelig. Teknologiutviklingen har ført til stor kvalitetsforskjell på DMK etablert tidlig og sent i etableringsperioden [3].

Digitaliseringa av markslaget i ØK ble gjort ved at man definerte Digitalt markslagskart som en SOSI-standard for markslaget. SOSI står for «Samordnet opplegg for stedfestet informasjon», og er en standard for geografisk informasjon utviklet og brukt i Norge. SOSI er både en standard for hvordan geografisk informasjon skal beskrives, og et format for utveksling av geografiske data.

Alt markslagsklassifisert areal fikk i DMK kode for arealtilstand. Arealtilstand gir opplysninger om arealtype, arealbruk, treslag og grunnforhold. Eksempler på verdier for arealtilstand er fulldyrka jord, barskog og fjell i dagen. Jordbruksareal og dyrkingsjord fikk i tillegg kode for driftsforhold, skogarealer fikk kode for bonitet og tilleggsklassifisering, mens myr og torvmark fikk kode for omdanningsgrad, myrdybde og vegetasjonstype.

Felles kartdatabase (FKB) er en samling detaljerte grunndata tilpasset bruk i målestokk 1:5000 - 1:30000. DMK inngikk i FKB fra 1992 til 2009, og fulgte kravene til FKB-datasett. Se [27] for informasjon om overgangen til et SOSI-standardisert, digitalt datasett og [62] for informasjon om siste versjon av SOSI-standard for DMK.

### **Ajourføring av DMK**

NIJOS oppmuntret kommunene til administrativt og årlig ajourhold av DMK. Kartdata er ferskvare, og blir raskt utdaterte dersom de ikke ajourføres. Veier og hus bygges på jordbruksareal, golfbaner anlegges, skog hugges til fordel for industrifelt, elva finner seg et nytt løp og veien blir lagt om...

I løpet av kort tid skjer det svært mange endringer. Ikke alle arealegenskaper endrer seg like fort. For eksempel er høyde over havet noe som endrer seg svært sakte, mens skogsjordas egnethet for oppdyrking endrer seg ganske sakte (dersom det ikke nedbygges eller skjer andre spesielle hendelser).

Ajourføring av kart kan være ressurskrevende, men helt nødvendig for at informasjon som representerer dynamiske fenomen ikke skal forvitte og bli ubrukbar. Når

man ajourfører et digitalt kart, sier man gjerne at man ajourfører et datasett eller geodata. Årsaken er at digital geografisk informasjon er så mye mer enn det framstilte basiskartet.

Markslaget ble opprinnelig ikke oppdatert systematisk. Noe ajourføring har blitt gjort i sammenheng med jordregisterproduksjon og overgang til DMK, men det ble etter hvert et stort etterslep, og DMK fikk kritikk for og være gammelt og lite oppdatert [43].

Etter hvert som man så nytten av digitale kart i nye sammenhenger, slik som fordeling av arealtilskudd i landbruket og jordvern, utkrystalliserte det seg et tydelig behov for oppdaterte arealressurskart i stor målestokk. I dokumentet *Veileder for administrativt ajourhold av digitalt markslagskart* [3] er administrativt ajourhold beskrevet som:

«Ajourhold av datasett basert på hendelser og rapportering , mens årlig oppdatering omfatter tilgjengeliggjøring av de administrative ajourføringene sammen med oppdateringer basert på endringer i andre FKB-datasett som vei og vann.»

I tillegg til årlig administrativ ajourføring og kontinuerlig oppdatering gjort av kommunene, så man for seg et 5-10 årig periodisk ajourhold, et mer omfattende ajourhold basert på flybilder og utført av NIJOS. Det ble over jordbruksavtalen bevilget penger til ajourføring av DMK fra og med 2002. Det viste seg imidlertid vanskelig å få til en systematisk ajourføring, ikke minst fordi det ble for krevende for kommunene å gjennomføre sin del av arbeidet. Behovet for et enklere arealressursdatasett i stor målestokk som kunne dekke behovet innen landbruksforvaltning og arealplanlegging, samtidig som det var mulig for kommunene å holde det à jour, ble mer og mer tydelig. Løsningen ble et nytt og enklere datasett, AR5.

I kapittelet om tilgjengelige datasett (kapittel 10), avsnitt 10.2 er det mer å lese om DMK generelt, og dyrkingsjord i DMK spesielt.

### 3.3 DMK forenkles – AR5 skapes

I 2005 la NIJOS fram et forslag til modell for et nytt arealressurskart i målestokk 1:5000 [6]. Kartet fikk navnet AR5, og skulle dermed bli en naturlig del av instituttets serie med arealressurskart, som fra før bestod av AR50 og AR250. Forslaget til ny modell la opp til en forenkling av DMK, og målet var å lage et datasett (og kart) med nødvendig arealressursinformasjon for landbruksforvaltning og arealplanlegging, samtidig som det skulle bli enklere for kommunene å holde det oppdatert. Viktige og store forenklinger, var å gjøre klassifikasjonssystemet enklere og mer logisk, gjøre reglene for koding enklere og mer i samsvar med andre datasett, samt å utelate driftsforhold og en del andre ganske statiske egenskaper. Samtidig lovet man at alle egenskaper fra DMK fortsatt skulle vær enkelt tilgjengelig, men da ikke ajourført.

Det var lenge tvil om informasjon om dyrkingsjord skulle videreføres i AR5. Egenskapen dyrkingsjord er forholdsvis statisk, men dyrkingsjord har samme vern etter Jordlova som jordbruksareal, og bør derfor være lett tilgjengelig. I [6] foreslås det å føre informasjonen videre i en forenklet form for dyrkingsjord i skog, men å utelate

informasjon om innmarksbeite og overflatedyrka jord som kan fulldyrkes. Etter en høringsrunde og ytterlige diskusjoner, ble det bestemt at informasjon om dyrkingsjord ikke skulle tas med over i AR5, men ligge igjen i DMK (se beskrivelsen av det vedtatte klassifikasjonssystemet for AR5 [5]). Klassifiseringen av jordbruksarealer i fulldyrka jord, overflatedyrka jord og innmarksbeite ble videreført, mens informasjon om driftsforhold ikke ble tatt med over til AR5. I 2009 var overgangen til AR5 fullført for alle landets kommuner, og AR5 hadde erstattet DMK i FKB. Der hvor DMK hadde over 3700 lovlige kodekombinasjoner, har AR5 rett i overkant av 100. Behovet for markslagskompetanse i kommunene for å utføre ajourhold ble slik kraftig redusert.

I kapitlet om tilgjengelige datasett (kapittel 10), avsnitt 10.3 beskrives AR5 generelt, med spesielt fokus på jordbruksarealer.

### 3.4 Systematisk ajourføring av AR5 blir en realitet

AR5 blir holdt oppdatert gjennom kontinuerlig ajourhold i kommunene og periodisk ajourhold hos Skog og landskap. Det kontinuerlige ajourholdet er en del av Geovekst (samarbeid om etablering og vedlikehold av de mest nøyaktige kartdata i Norge), mens det periodiske ajourholdet i tillegg finansieres gjennom bevilgninger over Jordbruksavtalen, kombinert med bevilgninger over Statsbudsjettet. Slik sikres et tilstrekkelig oppdatert datasett som kan benyttes til ulike formål, men der landbruksforvaltning og arealplanlegging står i sentrum.

Det er opprettet et forvaltningsregime rundt ajourholdet som sørger for at de ulike versjonene av datasettet tas vare på i databaser hos Skog og landskap. Det vil i neste runde gjøre det mulig å foreta analyser på endringer gjort i datasettet. Kommunene er ulikt rustet og motivert til å utføre det kontinuerlige ajourholdet, og kvaliteten varierer naturlig nok fra kommune til kommune. Derfor er et regelmessig periodisk ajourhold viktig. På Skog og landskap sine nettsider [60] står det å lese:

«Det periodiske ajourholdet skal sikre et homogent nasjonalt datasett ved å rette feil, fange opp arealendringer, samt unøyaktigheter i det kontinuerlige ajourholdet.»

Kommunene skal levere kontinuerlig ajourholdt AR5 hvert år i henhold til Geovekstavtalen. Allikevel er det langt fra alle kommuner som gjør det. Det kontinuerlige ajourholdet vil typisk fange opp endringer knyttet til kommunal saksgang, og ikke endringer som skjer gradvis over tid, som for eksempel gjengroing. Det periodiske ajourholdet fanger opp alle typer endringer i tilknytning til jordbruksareal, og oppdaterer i tillegg informasjon om samferdsel, vann og bebyggt areal. Det er ennå ikke helt avklart hvor hyppig det periodiske ajourholdet vil skje. Det kan uansett ikke skje oftere enn det finnes nye ortofoto for et område.



## 4 Databasesystemer

For å kunne håndtere store datamengder på en skikkelig måte, bruker vi databasesystemer. Databasesystemet sørger for kritisk funksjonalitet, som sikker lagring, tilgang til data via programmeringsgrensesnitt og spørrespråk og transaksjonshåndtering [14]. Det å lagre informasjon i et databasesystem skiller seg fra det å lagre data i minnet (RAM) under eksekvering av et program, ved at databasen sikrer at dataene er tilgjengelige også når programmet er ferdig med å kjøre og minnet på datamaskinen tømmes. Det vil lagring i en fil også gjøre, men et databasesystem håndterer enormt mye større datamengder på en effektiv måte enn en fil vil kunne gjøre. Leter vi etter en særskilt «bit» med data i en fil, må vi lete gjennom hele filen. Det kan ta svært lang tid selv for en middels stor fil. I et databasesystem er det innebygde mekanismer som gjør det mye raskere å finne fram i dataene.

Et databasesystem er også designet for å oppføre seg forutsigbart og sikkert ved feilsituasjoner, som for eksempel ved disk-krasj eller programvarefeil. En annen fordel med databasen framfor en vanlig fil, er at i en database har man mulighet for å definere ulike logiske lag, eller eksterne view, tilpasset bruken av dataene. I tillegg separerer et databasesystem det konseptuelle skjemaet (som beskriver databasens logiske struktur) fra det interne skjemaet (som beskriver den fysiske lagringen av dataene) [13]. Slik blir det lettere å tilpasse de samme dataene ulik bruk samtidig som man ikke trenger å dukke ned i implementasjonsdetaljer om hvordan dataene faktisk lagres på disken.

Kort sagt: et riktig konfigurert databasesystem sikrer trygg lagring av data og effektiv gjenfinning av data, selv for svært stor datamengder.

### 4.1 Databasemodeller

Det finnes flere ulike modeller for databaser. Tidlige modeller var den grafbaserte nettverksmodellen og den trebaserte hierarkiske modellen. Begge disse to modellene manglet et høynivå spørrespråk, og brukeren måtte bruke mye ressurser på å lage en struktur på dataene som forenklet det å skrive programmer som brukte dataene. Uansett krevde det å skrive slike programmer mye, selv for enkle spørringer.

I dag er det i hovedsak tre modeller som er aktuelle: relasjonsmodellen, objektmodellen og objekt-relasjonsmodellen. Sistnevnte kombinerer de to foregående.

### 4.2 Relasjonsdatabaser

I 1970 skrev Ted Codd sin berømte artikkel som skulle komme til å prege databaseteknologien i årtier [14]. Han foreslo at databasen skulle baseres på det å strukturere data i todimensjonale tabeller som han kalte relasjoner (se figur 4.1). Hver kolonne

representerer et attributt, og alle attributter skal være atomiske. En rad i en tabell, kalles et tuppel. Bak den synlige tabellstrukturen, kan det i en relasjonsdatabase være komplekse lagringsstrukturer som sørger for rask respons på spørringer. I motsetning til tidligere databasemodeller, skulle brukeren slippe å tenke på hvordan tabellene faktisk ble lagret. Systemet skulle selv sørge for dette. For å optimalisere spørringer, kan man lage indekser på attributter, enten enkeltattributter eller flere attributter samtidig.

Relasjonsdatabasen er i dag, 41 år etter at Codd publiserte sin artikkel, fortsatt den mest brukte databasemodellen. Relasjonsdatabasen har et matematisk fundament i mengdelæren. Spørringer kan beskrives matematisk på ulike måter, relasjonsalgebra og tuppelkalkyle er to kjente notasjoner.

Codds relasjonsdatabase vant straks terreng framfor den hierarkiske modellen og nettverksmodellen. Det skyldtes ikke at relasjonsdatabasen hadde så mange flere muligheter, for det hadde den ikke. På grunn av sitt solide teoretiske fundament, var den mye enklere å bruke. Ikke minst spilte det en stor rolle at relasjonsdatabasen hadde en generell, matematisk og logisk fundert basis for spørringer.

Avdeling	Årsbudsjett
IT	3 000 000
HR	3 500 000
Salg	4 750 000

Navn	Avdeling	Lønn	Ansattdato
Ole Olsen	Salg	467 000	18.01.2009
Hans Hansen	IT	390 000	23.09.2008
Nils Nilsen	HR	430 000	15.03.2009
Bolla Pinnsvin	Salg	398 000	20.09.2010

Figur 4.1: Liten relasjonsdatabase med to tabeller

Flere høynivå spørrespråk har blitt utviklet for relasjonsdatabasen. Av disse er det SQL (Structured Query Language) som har blitt det viktigste. SQL implementerer funksjonene fra de matematiske notasjonene for spørringer. Alle de store relasjonsdatabasesystemene støtter SQL. SQL er i hovedsak et deskriptivt språk, i motsetning til for eksempel Java eller C, som er imperative språk. Det betyr at når man skriver en spørring i SQL, beskriver man resultatet man ønsker, ikke hvordan systemet skal komme fram til det ønskede resultatet.

Spørringen under er et eksempel på en SQL-spørring der de to tabellene fra figur 4.1 (tabellene avd og ans) og en aggregeringsfunksjon inngår. Resultatet av spørringen lister opp avdelingene og teller opp antall ansatte i hver avdeling. SELECT, FROM, WHERE og GROUP BY er nøkkelord i SQL-språket:

```
SELECT avd.avdeling, COUNT(navn) as ant_ansatte
FROM avd, ans
WHERE ans.avdeling=avd.avdeling
GROUP BY ans.avdeling;
```

```
avd.avdeling    ant_ansatte
-----
IT              1
HR              1
Salg           2
```

## 4.3 Objektdatabaser

Objektdatabaser, eller objektorienterte databaser, utvider konseptet med objektorienterte programmeringsspråk til å inkludere persistens. I objektorientert design, er objektet det sentrale konseptet. Et objekt kan både ha egenskaper og oppførsel (metoder), og det har alltid en unik identifikator. Et objekt kan ha relasjoner til andre objekter.

I en bedriftsdatabase er en ansatt et typisk objekt, en avdeling et annet og en kunde et tredje. Objekter med liknende egenskaper grupperes i klasser, for eksempel klassene *Ansatte*, *Avdelinger* og *Kunder*. Objektene kan ha relasjoner seg i mellom, for eksempel at «en ansatt jobber i en avdeling». Et objekts egenskaper kan være komplekse strukturer som mengder og lister. Det gjør den objektorienterte modellen mer fleksibel og kompleks enn relasjonsmodellen, der attributter er atomiske. Imidlertid mangler objektdatabasene et spørrespråk av samme kaliber som SQL. Det er en av de viktigste grunnene til at relasjonsmodellen fortsatt er den mest populære av de to modellene [14].

## 4.4 Objekt-relasjonsdatabaser

Objekt-relasjonsdatabaser er relasjonsdatabaser som implementerer konsepter fra objektorienterte databaser. Typiske elementer som implementeres, er komplekse strukturer i stedet for kun atomiske attributter, metoder (oppførsel), identifikasjon av tupler og referanser [14]. De fleste store relasjonsdatabasene er i dag utvidet med konsepter som gjør dem til objekt-relasjonsdatabaser. Det gjelder PostgreSQL, Oracle, DB2, Informix og Microsoft SQL Server [56, 50, 52]. Samtidig som man har implementert elementer fra objektdatabasene, har bakoverkompatibiliteten blitt tatt vare på. Det har vært viktig, siden det finnes enormt mange applikasjoner som bruker standard relasjonsdatabaser.

SQL tok opp i seg objektorienterte elementer i SQL:1999, slik at det nå er mulig å implementere abstrakte datatyper (ADT) i SQL. Det er også mulig å implementere sine egne datatyper, såkalte brukerdefinerte datatyper (engelsk: User defined datatype, UDT). I en abstrakt/brukerdefinert datatype, er det mulig å definere oppførsel. Det er ikke mulig i en standard relasjonsdatabase, selv om det til en viss grad kan imiteres ved hjelp av triggerer.

## 4.5 Bruksområder og transaksjonstyper

En transaksjon er en samling av en eller flere databaseoperasjoner. En transaksjon kan slette, oppdatere, legge til, eller bare lese data. I et moderne databasesystem, kan det foregå mange transaksjoner som bruker de samme dataene samtidig. Det setter krav til hvordan transaksjonene oppfører seg isolert og i samspill med de andre. Derfor må enhver transaksjon i et tradisjonelt databasesystem oppfylle ACID-testen, og ethvert databasesystem ha en form for samtidighetsstyring. Mer om dette i de påfølgende avsnittene.

## ACID

Tradisjonelt skal en transaksjon bestå «ACID-testen». Her er en forklaring på hvert av elementene i testen, oversatt til norsk fra en lærebok i databaseteori [14]:

- **A** står for det engelske ordet *atomicity*, atomisk, enten eksekveres hele transaksjonen eller ingenting av den
- **C** står for det engelske ordet *consistency*, konsistens, dvs at en transaksjon skal bevare en databases konsistens
- **I** står for det engelske ordet *isolation*, isolasjon, som betyr at enhver transaksjon eksekveres som om ingen andre transaksjoner eksekveres samtidig.
- **D** står for det engelske ordet *durability*, varighet, betingelsen om at en transaksjons effekt på databasen aldri går tapt igjen så snart en transaksjon er komplett.

ACID-tankegangen passer godt for ordinære, sentraliserte databasesystemer med korte transaksjoner. Databasesystemer er etter hvert tatt i bruk på nye måter. Det kan være løsninger som er distribuerte, de lagrer komplekse datatyper og de har lange transaksjoner. Da kan det være vanskelig å oppfylle ACID-kravene. Det kan også finnes tilfeller der det ikke er hensiktsmessig å kreve at ACID-kravene oppfylles i sin helhet, men at man med vilje slipper opp litt på ett eller flere av kravene. Dette diskuteres i artikkelen om AuTrA [22], en modell for transaksjoner som kan lempe på et eller flere av ACID-kravene etter behov.

### Samtidighetsstyring

I et databasesystem eksekveres ofte mange transaksjoner samtidig mot de samme dataene. Man må derfor ha en strategi, og en metode, for å sikre at alle transaksjoner oppfyller kravene til ACID-testen (eller et passende alternativ der kravene er lempet på). Dette kalles samtidighetsstyring (eng: concurrency control). Det mye brukte spørrespråket SQL, krever som standard at transaksjoner skal utføres på en serialiserbar måte. Seriell (engelsk: serial) betyr at en funksjon eksekverer fullstendig før den neste begynner. At noe er serialiserbart, innebærer at en eller flere handlinger/funksjoner oppfører seg som om de eksekveres serielt, selv om eksekveringen overlapper i tid. Det betyr at det ikke er noen interaksjon mellom transaksjonene.

Det er imidlertid mulig å sette alternative isolasjonsnivåer også i SQL. Alternativene er *serializable*, *read-uncommitted*, *read-committed* og *repeatable-read*. Isolasjonsnivå *read-uncommitted* tillater lesing av data som ikke ennå er committed, såkalte *dirty reads*. Read committed hindrer lesing av *dirty data*, men tillater at en transaksjon som utfører samme spørring flere ganger får ulikt svar så lenge svarene kommer fra data skrevet av allerede committede transaksjoner. Settes isolasjonsnivå *repeatable-read*, tillater man ikke at et tuppel hentet en gang i transaksjonen forandrer seg neste gang det hentes i samme transaksjon. Imidlertid kan påfølgende eksekvering av samme spørring introdusere såkalte fantom-tupler, det vil si tupler som har blitt satt inn i en relasjon av en annen transaksjon mens vår transaksjon har kjørt [14].

## Mange korte og samtidige transaksjoner eller tunge analyser på store datamengder?

Databaser må designes ulikt avhengig av deres primære bruksområde. Noen databaser er designet for mange samtidige og korte (raske) transaksjoner, gjerne i sann tid. Et eksempel er en bank-database med kunders bankkontoer. Et annet eksempel er en database som håndterer bestilling av flybilletter. Slike databaser kalles ofte OLTP-databaser der OLTP er en forkortelse for *Online transaction processing*. OLTP-databasen er det tradisjonelle databasen.

Andre databaser er designet spesielt for å kunne håndtere komplekse, sammensatte spørringer på svært store datamengder som kun oppdateres i bolker, ofte med data fra OLTP-databaser. Slike databaser kalles gjerne OLAP-databaser. OLAP er en forkortelse for *Online Analytic Processing*. Romlige databaser optimalisert for analyse og beslutningsstøtte kalles gjerne SOLAP (spatial OLAP) [15]. Andre databaser forsøker å være et kompromiss mellom OLTP og OLAP, i tillegg til å håndtere samtidige transaksjoner, skal den også kunne gjøre enklere analyser på rimelig tid.

## Lange transaksjoner

En lang transaksjon er en transaksjon som bruker for lang tid til at det er hensiktsmessig å fullstendig låse data som andre transaksjoner vil kunne trenge samtidig. Databasesystemenes strategi og metode for transaksjonshåndtering og samtidighetsstyring er ofte uegnet i slike situasjoner. Mens et tradisjonelt databasesystem, som for eksempel en database som håndterer penger ut og inn av en banks kundekontoer (en OLTP-database), typisk har mange og korte transaksjoner som passer som hånd i hanske med ACID-kravet og ønsket om serialiserbarhet, brukes nå også databasene til andre typer data det knyttes andre typer transaksjoner til. Hva som er for lang tid i en lang transaksjon, avhenger av omstendighetene [14].

En database med romlige data er et godt eksempel på en type database som kan kreve alternative løsninger på grunn av lange transaksjoner. I et geografisk informasjonssystem kan en lang transaksjon vare fra minutter til flere uker. Oppdatering av et enkelt objekt er forholdsvis raskt, mens ajourføring av et detaljert datasett for en hel kommune kan være svært tidkrevende (for eksempel AR5). Håndtering av lange transaksjoner skjer gjerne i API-er bygget over et geodatabasesystem. Et eksempel er Quadri Map Server (QMS) utviklet av Norkart for Statens kartverk. På Norkart sine websider for QMS [51], kan man lese om transaksjonshåndtering i QMS:

«Metoden som benyttes ved ajourføring/endring av data kalles lange transaksjoner. Når data innenfor et geografisk område skal ajourføres, overføres disse til en lokal prosjektorientert database hos klienten (utsjekk) der selve bearbeidingen og endringene utføres. Når data er «ute til oppdatering», blir de låst i databasesystemet slik at andre brukere ikke kan endre de samme dataene. Når man er ferdig med endringsprosjektet, føres den lokale databasen tilbake til systemet, og kun endringene oppdateres (innsjekk). I denne prosessen håndteres enhver form for topologi og sammenheng i datastrukturen.»

# 5 Geografiske databaser

En romlig database er et databasesystem som er optimalisert for lagring og spørringer mot data relatert til objekter som har en plassering i rommet, eller rett og slett til lagring av et fenomen og dets spredning i rommet.

Geografiske databaser (geodatabaser) kan sies å være en spesialisering av en romlig database. En geodatabase håndterer geografiske referansesystemer, eller mer presis: geodetiske referansesystemer (for eksempel Euref 89 UTM-soner). Først når innholdet i databasen er georeferert, er det å betrakte som geografiske data, og kan brukes til å lage et kart. Den geografiske databasen er kjernen i et moderne geografisk informasjonssystem. Ved hjelp av et geografisk databasesystem blir vi i stand til å utføre sentrale oppgaver som kreves av et moderne geografisk informasjonssystem, inkludert sikker lagring av data, håndtering av flere samtidige brukere og rask gjenfinning av data [32].

Den typiske romlige og geografiske database er implementert som et tillegg til en vanlig database. Et eksempel er fri programvaren PostGIS, som er et tillegg til fri programvare-databasen PostgreSQL. Et annet er ArcSDE fra ESRI, som kan benyttes med flere databaser, deriblant Oracle, Microsoft SQL server og PostgreSQL. Programvare som PostGIS og ArcSDE, sies på engelsk å *spatially enable the database*. Med det menes nettopp at databasen gjøres i stand til å ta i mot geografiske data og lagre dem på hensiktsmessige måter, samt å hente fram data man spør etter raskt, for eksempel gjennom romlige SQL-spørringer eller Queries/Filter i webtjenester som Web Feature Service.

## 5.1 Konseptuell modellering av geografiske fenomener

Det er i hovedsak to måter å modellere geografisk informasjon på, objektmodellen og feltmodellen [25, s. 67-69] [41, s. 140-163].

Objekter er diskrete og endelige. Det kan for eksempel være en vei, en eiendom eller en høyspentmast. De modelleres konseptuelt ved hjelp av objektmodellen, der enhver forekomst er et objekt i systemet med attributter og eventuelt oppførsel. Nedbør, gravitasjon og høyde, er eksempler på fenomener i naturen det ikke er naturlig å representere som objekter, fordi de ikke har noen naturlig romlig avgrensning i domenet. De modelleres gjerne i henhold til feltmodellen, der området man undersøker deles inn i små biter, det kan være rektangulære celler eller andre regelmessige eller uregelmessige representasjoner, som for eksempel et TIN. Hvert område gis så en eller flere verdier, for eksempel antall millimeter nedbør siste måned. Objektmodellen og feltmodellen er ikke fullstendig adskilte. Det er gjort forsøk på å koble de to modellene [12], eller å forene dem i et felles rammeverk med felles grunnkomponenter [23].

Heldekkende arealressurs-informasjon som AR5 og DMK er gode eksempler på noe som kan passe inn i begge modeller, avhengig av perspektiv og applikasjon. Bonden betrakter nok en bestemt skogteig eller et bestemt jorde som et objekt med diskret og endelig utstrekning og gitte egenskaper. For en nasjonal arealdekke-database, er det vel så naturlig å se på arealdekke som en kontinuerlig (om enn nominell) feltbasert egenskap.

ISO-standarden ISO 19123 [18] (se også kapittel 8 om standarder) spesifiserer heldekkende data (både diskrete og kontinuerlige). AR5 og DMK er det man kaller «discrete coverage» i denne standarden.

## 5.2 Lagring av geografiske data

Romlige data kan lagres som vektordata. Det inkluderer geografiske objekter med komplekse datatyper som punkter, kurver og polygoner som representerer «feature» i interesseområdet. Vektordata egner seg godt til å beskrive diskrete objekter, for eksempel bygninger. I objektmodellen kan de ulike objektenes romlige forhold til hverandre lagres (topologi). Romlige data kan også lagres i en rasterstruktur der området deles inn i et rutenett med rutestørrelse tilpasset de data som skal representeres. Hver rute har minst en verdi knyttet til seg. Verdien angir en karakteristikk ved denne ruten (for eksempel om det er skog eller vann).

Rasterdata-modellen egner seg best til å beskrive kontinuerlige fenomener, for eksempel arealdekke. Vektormodellen er konseptuelt mer kompleks enn rastermodellen [25, s. 67-69]. Andre måter å representere geografiske data i datasystemer, er halvplan og andre tesseleringer enn regulære rutenett.

Det er ikke et en-til-en forhold mellom feltmodellen og lagring som rasterdata, eller objektmodellen og lagring som vektordata. Felldata kan lagres som vektordata, og objektdata kan lagres som rasterdata [41].

## 5.3 Representasjon av vektorobjekter

For vektordata, kan man velge å beskrive forholdet mellom ulike objekter i databasen med ulike teknikker. Med forhold mellom ulike objekter, menes her topologiske relasjoner som overlapp og naboskap. Topologiske relasjoner kjennetegnes ved at de er invariante ved topologiske transformasjoner, det betyr at de ikke endrer seg ved translasjoner, rotasjoner og skaleringer i planet.

Det enkleste modellen for lagring av geometri, er simple feature-modellen som er spesifisert i OGC sin standard *Simple Feature Access* [28] (se også kapittel 8 om standarder). Her beskrives et geografisk objekts geometri uavhengig av alle de andre objektene i basen. Topologiske relasjoner er ikke lagret. Modellens sterke side er enkelhet. Ulemper er at det er tyngre å sikre konsistens ved editering fordi felles avgrensingslinjer lagres to ganger, og at det kan ta tid å regne ut de topologiske relasjonene når disse trengs. PostGIS er et eksempel på et geodatabasesystem som primært benytter denne geometrimodellen.

Den topologiske modellen er mer avansert. Her vil topologi være lagret. En felles

grenselinje mellom to polygoner vil vite at den er en felles linje. Dermed kjenner alle polygoner sine naboer. Ved oppdatering av avgrensingslinja til et polygon, vil nabopolygonet med felles grenselinje automatisk oppdateres. Med denne modellen, snus fordeler og ulemper med simple feature-modellen på hodet. Norkart sin Quadri Map Server er eksempel på en geodatabase som håndterer topologi. PostGIS har hatt en topologimodell i noen år, men uten at den har vært mye brukt eller er godt dokumentert. I versjon 2.0 av PostGIS, som forventes lansert i januar 2012, blir topologimodellen tatt inn i varmen. Topologimodellen og tilhørende funksjoner i PostGIS 2.0 sies å skulle implementere ISO standarden *SQL/MM Spatial*, i tillegg til å ha sine egne funksjoner [55].

## 5.4 Usikkerhet i geometrisk og temporal avgrensning og fastsetting av egenskaper

En avbildning av virkeligheten inn i et geografisk informasjonssystem vil aldri være perfekt. Avbildningen skapes gjennom først å observere interesseområdet, deretter representeres det i informasjonssystemet. Både observasjon og representasjon er en kilde til ufullstendig eller gal informasjon. Vi kan dele årsakene til ufullstendighet og feil i to grupper: unøyaktighet og upresisitet. Unøyaktighet er synonymt med feil, og skyldes mangel på sammenheng mellom informasjonssystemet og virkeligheten. Unøyaktighet kan oppstå både ved observasjon og representasjon. Upresisitet skyldes mangel på detaljer i observasjon eller representasjon [41].

I et informasjonssystem må man være bevisst på krav til nøyaktighet og presisjon, det er en viktig del av kvalitetsinformasjonen. Denne informasjonen er essensiell for å vite hva et datasett egner seg til, og hva det ikke kan brukes til. SOSI-standardene bruker begrepene målemetode, nøyaktighet og synbarhet for å beskrive stedfestingskvaliteten.

Noen geografiske objekter (i betydningen feature) har klare avgrensingslinjer. Det gjelder for eksempel en bygning eller en parkeringsplass. Andre overganger i naturen og landskapet er mer gradvise. Gode eksempler er overgang mellom løvskog og barskog, skillet mellom elv og land (vannføringen vil påvirke dette) og overgangen mellom by og land. Både geometriske avgrensninger og skillet mellom ulike egenskaper kan være uskarpe. Teorien om flytende mengder (fuzzy sets) er utviklet for å håndtere slike diffuse overganger. Ved hjelp av medlemsskapsfunksjoner og operatorer som union og snitt, spesifiseres et teoretisk fundament for behandling av objekter uten skarpe avgrensninger [8, 25, s. 68]. De store geodatabasesystemene har ikke støtte for flytende mengder.

Det er verdt å merke seg at overganger i tid også kan være uskarpe [30]. Det er for eksempel ingen klar skillelinje i tid for når et jordbruksareal under gjengroing slutter å være jordbruksareal og begynner å være skog.

## 5.5 Spørringer

De spørringer en geodatabase skal kunne svare raskt på, kan deles inn i tre [32]:



- spørringer med alfanumeriske kriterier: «hvor store arealer i dekar har arealtype fulldyrka jord.»
- spørringer med romlige kriterier: «finn alle markslagspolygoner med arealtype fulldyrka jord som grenser til bebygd areal»
- interaktive spørringer: initieres gjerne av en brukers handling, for eksempel det å tegne opp alle markslagspolygoner innenfor et gitt rektangel på brukerens skjerm

De ulike typene spørringer kan kombineres i komplekse spørringer, og spørringene kan kombinere flere datasett. En geografisk database må kunne svare effektivt på alle de tre typene spørringer, og kombinasjoner av disse.

For å kunne gjøre raske oppslag i store mengder geografiske data, brukes romlige indekser. Det finnes mange varianter av romlige indekser, den enkleste er kanskje cellemetoden. Det plasseres et rutenett over hele området, og man etablerer og vedlikeholder en liste med hvilke objekter som befinner seg i hvilke ruter. For å finne alle objekter i et gitt område, kan man først begrense søket til de rutene som befinner seg i området, deretter se på de objektene som befinner seg i de aktuelle rutene [32].

Romlige indekser basert på trestrukturer er en annen indekseringsmetode. Trestrukturer er bedre for dynamiske data enn cellemetoden, siden man i cellemetoden må bestemme rutestørrelsen på forhånd. R-tre er den kanskje mest kjente og brukte trestruktur-indeksen i romlige databaser, da den er godt optimalisert med tanke på få disk-aksesser i databaser [16].

PostGIS/PostgreSQL bruker R-tre implementert over en GiST-indeks («Generalized Search Tree») som standard indeks for romlige data [54]. ArcSDE-geodatabaser bruker enten cellemetoden eller R-tre avhengig av hvilket databasesystem geodatabasen er implementert i. Er det Oracle, brukes cellemetoden, er det PostgreSQL brukes R-tre [45].

# 6 Temporale databaser og informasjonssystemer

Til forskjell fra analoge arkiver, overskrev de tradisjonelle databasesystemene data ved oppdateringer, slik at det bare var den siste tilstanden som til enhver tid lå i systemet (i mange databaser er det fortsatt sånn). Man så ganske snart behovet for å utvide med mulighet for å håndtere lagring av endringer over tid i databasesystemene. Først ble dette gjort som utvidelser til relasjonsmodellen, senere har vi også fått utvidelser til objektmodellen. De enkleste modellene basert på arkivlogger, transaksjonslogger og separate historiske relasjoner for enkelte attributter, tilfredsstillte ikke kravene til moderne modellering og effektive spørringer.

## 6.1 Tidsbegreper

### Tidsoppløsning og datatyper

Hvilken oppløsning man skal bruke for tid, avhenger av sammenhengen. Tidspunktet for en transaksjon mot en bankkonto vil måtte registreres mer nøyaktig enn tidspunktet for hogst i skogen. For førstnevnte, vil det være naturlig å gå ned til sekund, eller i hvert fall til minutt. For hogsten, kan dag eller uke være passende. I en database som lagrer informasjon om beløp på en bankkonto til enhver tid, vil man derfor naturlig velge en mer nøyaktig tidsangivelse enn for en database med oversikt over skogressurser.

Datatypene man har til rådighet for tidsangivelser i databaser, er normalt *date* (dato), *time* (tid), *timestamp* (tidsstempel) og *interval* (intervall). Det må tas stilling til om tidssone er relevant informasjon, og om man trenger spesielle mekanismer for å håndtere overgang mellom sommer- og vintertid når klokka plutselig er «det samme» to timer på rad eller man rett og slett hopper over en time, slik at kl. 02.00 blir kl. 03.00

Man må også ta stilling til om tiden skal registreres som et enkelt tidspunkt (øyeblikk), eller som intervaller.

### Tilstand og hendelser (states and events)

Et objekt er en instans av en «ting» som eksisterer over tid (for eksempel en person, et tog eller et jorde). Samlingen av objektets egenskaper utgjør objektets tilstand [35]. En tilstand har normalt en viss varighet. Et objekts tilstand kan forandre seg ved en hendelse. Selv om en hendelse selvfølgelig har en liten utstrekning i tid, representeres den oftest ved et enkelt tidspunkt. En tilstand representeres best ved et tidsintervall.

I en objektdatabase er objektet selve kjernen, og implementeres som en klasse med attributter og oppførsel. En hendelse vil endre et objekts egenskaper, og sette det i

en ny tilstand.

I relasjonsdatabasen, vil et objekt være representert ved en rad i en relasjon (tabell), egenskapene som sammen utgjør tilstanden finnes i attributtene (kolonnene). En hendelse representeres ved en databasetransaksjon som endrer en eller flere raders attributter.

### Temporale relasjoner

Allen [2] definerer relasjoner («Allen relationships») mellom tidsintervaller, som mange av dem er lett gjenkjennbare fra romlige relasjoner. Det er relasjonene «before», «after», «during», «contains», «overlaps», «overlapped-by», «meets», «met-by», «starts», «started-by», «finishes», «finished-by» og «equals».

## 6.2 Tid i relasjonsdatabasen

Richard T. Snodgrass er en autoritet på området temporale databaser. Han var hovedarkitekten bak TSQL2 (temporal SQL), og utga i 2000 boka *Developing Time-Oriented Database Applications in SQL* [35], som er å regne som et hovedverk innenfor temaet tid i relasjonsdatabaser. Et annen bok om tid i relasjonsdatabaser, er *Managing Time in Relational Databases: How to Design, Update and Query Temporal Data* [20]. Den utvider sentrale konsepter i Snodgrass sin bok slik at feilretting i data kan håndteres enda bedre.

Relasjonsdatabasen bygger tradisjonelt på et viktig prinsipp: en rad representerer et objekt eller en entitet. Attributter som identifiserer et objekt, brukes som primærnøkkel. De fleste modeller for å håndtere temporale data, bryter med dette fundamentale prinsippet. I stedet for en rad for hvert objekt, får man rader for enhver tilstand et objekt har hatt. Som primærnøkler, må man da bruke surrogatnøkler (for eksempel seksvensielle id-er) eller en kombinasjon av objektets id og ulik dato/tid-attributter [35, 20].

Det har vært gjort forsøk på å standardisere temporale spørringer i SQL helt siden 1992. Det har ikke lyktes. Likevel har flere databaseleverandører implementert konsepter fra temporal SQL i sine løsninger [59]. Uansett, uten et standardisert temporalt SQL, er det i stor grad opp til hver enkelt å finne sin løsning.

Mange artikler og bøker er skrevet om temaet temporale data, tabeller og databaser. Ulike datamodeller foreslås, og «best practise» oppsummeres. Boka [20] nevnt over, gir en god oppsummering av hvordan IT-verden har håndtert tidsaspektet i data i sin innledende del.

Den enkleste løsningen er øyeblikksbilder («snapshots»). Ved et gitt tidspunkt tas en kopi av databasen, og denne kopien fryses. Ofte tas øyeblikksbilder til faste tider, for eksempel den første i hver måned, eller ved årsskiftet. Det kan også være aktuelt å ta et øyeblikksbilde av en database slik den så ut da en viktig rapport ble utarbeidet, eller da vesentlige endringer i basen skulle påbegynnes eller akkurat var avsluttet. Øyeblikksbilder er enkle å lage og gir god dokumentasjon på en databases innhold ved et gitt tidspunkt. Det kan være tungt å trekke ut informasjon om endringer mellom ulike øyeblikksbilder av samme database. De tar også forholdsvis stor plass, da alle data dupliseres, også når kun et lite subsett av dem faktisk er endret.

En annen løsning, er å legge data som er utgått på dato over i en historisk tabell. Ulempen med dette, er at man må aksessere flere tabeller med potensielt ulike kolonner for å finne den informasjonen man er ute etter. Det finnes ulike modeller der utgåtte data ikke legges over i historiske tabeller, men blir liggende i original-tabellene sammen med de til enhver tid aktuelle dataene. Da har man versjonstabeller der radene i tabellen versjoneres. Den helt enkle varianten, er å legge til en kolonne for versjonsdato i databasen. Ved innsetting av ny rad, legges dagens dato i versjonsdato-kolonnen. Ved en oppdatering av et objekts egenskaper, lages en ny rad med en ny dato, mens den gamle blir liggende. Ved en sletting, forsvinner derimot alle data om objektet fra databasen. Det er ikke ønskelig. For å unngå det, kan man lage en boolsk kolonne med informasjon om et objekt er slettet eller ikke. Faren er at man glemmer denne egenskapen i spørringer, og at feil oppstår. Det er også et minus at man i de enkleste versjonstabellene ikke håndterer intervaller. Man går rett og slett ut fra at en rads startdato er forrige rad for samme objekt sin sluttdato. Dette er ikke alltid tilfelle. I tillegg må man aksessere flere rader for å finne et objekts levetid, siden sluttdatoen ikke er registrert i samme rad som startdatoen. Det kan vi løse ved å bruke tidsversjonering med intervaller, der alle rader har en start-dato og en slutt-dato. Her kan man legge skranker på data som hindrer overlapp og gap ved behov.

Fortsatt er det et stort problem med modellen vår: Den håndterer ikke retting av feil i data på en skikkelig måte. La oss si at vi i mars 2009 produserte en rapport over antall dekar jordbruksareal i en gitt kommune på basis av ajourført AR5. I juli 2009 fant vi ut det var en feil i AR5 for kommunen. Et stort område med jordbruksareal hadde blitt bygd ut allerede i 2006, men det hadde ikke kommet med ved ajourføringen i 2007. Derfor gikk vi inn i databasen og rettet dette i ettertid, slik at området ble næringsareal med startdato i 2006. Deretter ble vi bedt av oppdragsgiver å produsere den samme rapporten over jordbruksareal i den samme kommunen for mars 2009. Nå ville man fått et annet svar på grunn av feilrettingen, men man ville ikke vite hvorfor med mindre den som hadde rettet feilen, husket det. Løsningen på dette problemet, er å innføre flere tidsdimensjoner.

Det vanlige er å bruke to dimensjoner, da snakker man om bitemporale data. Snodgrass sin bok [35] omhandler bitemporale data, og hvordan de effektivt gjennom spørringer på tids-indekserte data kan brukes til å gi oss svar på spørsmål som inkluderer utvikling over tid.

### 6.2.1 Bitemporale data i relasjonsdatabasen

Bitemporale data har to ortogonale tidsakser, en for gyldig tid og en for transaksjonstid. Gyldig tid representerer tidsperioden for når noe forekom i den virkelige verden, og transaksjonstid representerer tiden noe var registrert i databasen. Ved å la begge tidsaksene bli representert ved en kolonne for startdato og en for sluttdato, får vi ved å legge til fire ekstra kolonner det som kalles en bitemporal tabell [35]. I tillegg, sier Snodgrass, finnes en tredje type tid: brukerdefinert tid. Alle de tre typene tid er ortogonale, det vil si helt uavhengige av hverandre.

En bitemporal tabell kan, sammen med passende spørringer i SQL gi svar på spørsmål om utvikling over tid, både i den virkelige verden (gyldig tid) og hva vi hadde registrert om verden (transaksjonstid):

- Hva skjedde i perioden 1998-2000?
- Hva hadde vi i 2008 registrert som høyden på huset i Kirkegata 23 før påbyggingen i 2009?
- Hva var høyden til huset i Kirkegata 23 i 2008 korrigert til i 2010?

I 2010 kom en annen bok som også omhandler bitemporale data og tabeller, men her er transaksjonstid omdefinert til «assertion time» [20]. For transaksjonstid, er alltid startdatoen sammenfallende med når noe ble registrert i databasen. For «assertion time» er startdatoen tidspunktet fra når vi mente/påstod at et objekt hadde en viss tilstand. Den kan registreres både i forkant og etterkant av startdatoen. Sluttdato håndteres på samme måten. I tillegg finnes en dato og tid-kolonne for når en rad ble registrert og eventuelt oppdatert i databasen.

## 6.3 Tid i objekt databaser

Om det mangler standarder og anerkjente rammeverk og spørrespråk for tid i relasjonsdatabaser, er objekt databasene enda mer basert på ad hoc-løsning av det enkelte problem man står overfor. Mye forskning er gjort på objekt databaser, ikke minst fordi objekt databaser klarer å avbilde interesseområdet mer realistisk enn relasjonsdatabaser. Allikevel er relasjonsdatabaser den absolutt mest anvendte utenfor akademia.

Donna Pequet [30] oppsummerer i sin bok bruk av objektmodell og feltmodell i henholdsvis objekt databaser og relasjonsdatabaser.

Det er relasjonsdatabaser som brukes ved Skog og landskap. Alle de store og mest brukte databasesystemene, i hvert fall utenfor akademia, er relasjonsdatabaser. Jeg velger derfor å ikke omtale tid i objekt databaser nærmere.

# 7 Romlig-temporale databaser og temporal GIS

Dersom et geografisk informasjonssystem skal kunne gi svar på spørsmål som omhandler prosesser, endringer eller modellering av fremtidige scenarier, må ikke bare det geografiske rommet håndteres, men også tidsdimensjonen. Databaser som håndterer både tid og rom, kalles romlig-temporale databaser. En romlig-temporal database er en viktig del av et temporalt GIS. Det vil variere hvor mye funksjonalitet som er direkte implementert i databasen, og hvor mye som er implementert i annen applikasjonsprogramvare som hører til et GIS. At databasen håndterer lagringen, er en selvfølge, men den kan også håndtere andre operasjoner som analyse, kvalitetskontroll og tidsplanlegging.

Geografiske data kan endre seg over tid på tre ulike måter: de kan endre egenskaper, de kan endre form og de kan endre plassering. En storm som flytter seg samtidig som den blir sterkere og dekker et større område, er eksempel på et fenomen som representerer alle tre typene endring. Endring kan være kontinuerlig (slik som stormens forflytning), eller diskret [39]. Et eksempel på en diskret endring, er et område med skog som hugges for å gi plass til et hus.

Eksempler på temporale databaser er en banks database med tidfestede konto-transaksjoner, en database med værdata for en værstasjon og bomstasjonens database over bilers passeringer. Disse databasene kan svare på spørsmål som hvor mye penger som stod på en gitt bankkonto et visst tidspunkt, hvor mye det regnet i juli 2011, og hvor mange biler som passerte bomstasjonen mellom kl 22 og 24 en onsdag kveld. Det er masse litteratur om temporale databaser. Geografiske databaser uten tidsdimensjonen finnes det også mange av, og det er skrevet mye om dem. Et eksempel er en vanlig topografisk kartdatabase, et annet er en database over høyspentledninger. Disse kan svare på spørsmål som hvor mange innsjøer over en viss størrelse det er i Norge, og hvor man må være ekstra oppmerksom på høyspentmaster når man flyr lavt i helikopter.

Det er lett å se at det å kombinere tid og rom åpner for svar på mer kompliserte spørsmål. Det kan være spørsmål om utviklingen innenfor et område over tid (historien), om prosesser og om fremtidsscenarier:

- Beskrive endring: Hvor mange dekar landbruksjord er blitt borte i perioden 2000-2010, og hva brukes områdene til nå?
- Prosess: Hvor beveger skogbrannen seg?
- Fremtidsscenario: Hvordan vil skoggrensa mot fjellet flytte seg i årene framover?

Det å lage en modell for romlig-temporale databaser kan angripes på tre ulike måter. Enten kan en romlig/geografisk modell utvides med temporale elementer, en

eksisterende temporal modell kan utvides med romlige elementer, eller man kan starte helt på nytt med en helt ny modell. Metoden der en romlig modell utvides med egenskaper som håndterer tid er mest brukt [1].

## 7.1 Sintons rammeverk

Sinton [34] sier at det er tre komponenter som må være en del av enhver observasjon som skal inngå i et geografisk informasjonssystem. Det er tema, sted og tid. Sinton brukes ordet attributt i stedet for komponent, men det er lett å forveksle med ordet attributt i relasjonsdatabaser. I tillegg trenger man informasjon om presisjon og hvor pålitelige dataene er. Uten disse elementene, er dataene ubrukelige i et geografisk informasjonssystem. Observasjoner må ofte generaliseres for å gjøre dem mindre voluminøse og lettere å forstå. Det er ulike teknikker for generalisering. Sinton nevner aggregering, induksjon og klassifikasjon. All generalisering fører til tap av informasjon, og den følger i henhold til Sinton et gitt mønster: en av de tre komponentene holdes konstant, en komponent får variere kontrollert og en måles. Det er avgjørende for det geografiske informasjonssystemet å vite og ta hensyn til i hvilken utstrekning informasjon går tapt i generaliseringsprosessen. Det gis i artikkelen en rekke eksempler på hvordan ulike typer data passer inn i dette mønsteret. Tabell 7.1 er hentet fra [24], og er en utvidet variant av Sintons tabell med eksempler på hvordan en komponent låses, en kontrolleres og en måles i konvensjonelle GIS. I tabellen brukes terminologien fra [24], der «tema» er byttet ut med «attributt».

Gail Langran gir i sin bok *Time in Geographic Information Systems*[24] en grundig fremstilling av forskning på og arbeid med romlig-temporale databaser fram til utgivelsesåret 1992. Sinton sin teori om målinger er svært sentral i boka. En geodatabase, er en samling av målinger. De fleste geodatabaser låser tiden, hevder Langran, og i et virkelig temporalt GIS/romlig-temporalt databasesystem, må både tid og rom kunne måles. Sintons rammeverk er viktig, sier Langran, fordi begrensninger i representasjon begrenser metoder og områder for geografisk forskning. De færreste databaser er både romlige og temporale Det å skape et virkelig temporalt GIS er derfor, i følge Langran, å frigjøre tidskomponenten fra stedfestingskomponenten slik at begge kan måles samtidig og uavhengig av hverandre.

<b>Datasekk</b>	<b>Låst</b>	<b>Kontrollert</b>	<b>Målt</b>
Jordsmonndata	Tid	Attributt	Sted
Topografisk kart	Tid	Attributt	Sted
Befolkningsdata	Tid	Sted	Attributt
Rasterbilder	Tid	Sted	Attributt
Værstasjonrapporter	Sted	Tid	Attributt
Flomoversikter	Sted	Tid	Attributt
Tidevanntabeller	Sted	Attributt	Tid

Tabell 7.1: Representasjon av geografiske data i varierte formater, utvidelse av Sintons eksempler, fra Langrans bok [24]

Etter at Langran skrev sin bok, har nok tilfanget av databaser som på en eller annen måte er både temporale og romlige økt betraktelig, men det er fortsatt vanlig å låse en av de tre komponentene sted, tid og attributt.

Jeg ønsker i denne oppgaven primært å lage et datasett med en datamodell som lar både tid- og stedkomponenten måles. Det er mulig å gjøre, men øker kompleksiteten.

## 7.2 Operasjoner i et geografisk informasjonssystem

Gail Langran definerer i sin bok følgende operasjoner som hovedoperasjoner i et temporalt GIS [24, s. 5] (liste fritt oversatt til norsk fra boken):

- Lagring (database): Lagre en fullstendig beskrivelse av studieområdet, samt endringer i den virkelige verden og i datalageret.
- Analyse (spørringer): Forklare, utforske og forutse komponenter og prosesser i en region.
- Oppdatering (bearbeiding): Erstatte utdatert informasjon med ny informasjon
- Kvalitetskontroll (= konsistenssjekk): Vurdere hvorvidt nye data er logisk konsistente med tidligere versjoner og tilstander.
- Tidsplanlegging (= prognose): Identifisere og forutse databasetilstander som fungerer som terskler som trigger predefinerte systemhandlinger.
- Visning (kart) Generere statiske og dynamiske kart, eller tabulariske oversikter, over temporale prosesser som virker i en region.

Boken kom i 1992, så det har skjedd noe siden utgivelsesdatoen. Allikevel er boken fortsatt mye sitert (Google Scholar viste 47 siteringer i 2010 og 2011 per 25.11.2011). De seks operasjonene er fortsatt like aktuelle, og et naturlig utgangspunkt for en gjennomgang av hovedoperasjoner i et temporalt GIS. I denne oppgaven vil jeg gå nærmere inn på lagring, analyse og oppdatering.

## 7.3 Lagring

Evnen til å lagre store mengder data er en kritisk faktor for et geografisk informasjonssystem [24]. Når tid innføres, vil datamengdene vokse ytterligere.

I dag er det nesten en selvfølge å benytte et geodatabasesystem for lagring av geodata. Det gir sikker lagring og effektiv gjenfinning. De aller fleste databasesystemene lagrer romlige data og ikke-romlige data i samme system, og de romlige dataene er ofte like lett tilgjengelige som de ikke-romlige egenskapene, for eksempel gjennom SQL-spørringer. Geodatabasene, som før gjerne var proprietære systemer, er nå en del av standard databasesystemer som Oracle og PostgreSQL, og nyter godt av den generelle IT-utviklingen. Det er knapt nok grenser for hvor store datamengder det er mulig å lagre i en database.

De fleste geodatabaser implementerer tid i en eller annen form, enten som SQL sine datatyper intervall og tidsstempel i relasjonsdatabasen, eller som ADT-er i objekt-databasen. I dag finnes standarder for hvordan tid kan håndteres i romlig-temporale databaser, se kapittel 8 om standarder.



## 7.4 Analyse

Geografisk analyse kan gi oss ny kunnskap. Ofte har vi behov for å studere endringer eller prosesser, og da kommer tidsdimensjonen inn i bildet. Langran hevdet at de fleste reelle romlig-temporale data enten mangler tidsdimensjonen, eller håndterer den ved hjelp av øyeblikksbilder. Et godt eksempel er markslagsdatabasen ved Skog og landskap. For DMK har vi ikke mer enn en versjon, for AR5 har vi såkalte øyeblikksbilder i form av nasjonale årsversjoner. Analyser på øyeblikksbilder kan være vanskelig og lite effektivt. Et problem er at mengden data som må prosesseres er avhengig av datasettets størrelse, ikke av antall endringer som er registrert. To øyeblikksbilder det har skjedd få endringer mellom, er like tunge å prosessere som to øyeblikksbilder det er store mengder endringer mellom.

En annet utfordring, uansett øyeblikksbilder eller ekte romlig-temporal database, er det å skille mellom reelle endringer og endringer som skyldes oppretting av feil i forrige versjon eller endring i metode eller teknologi. En nylig publisert artikkel som tar opp dette problemet, som også Langran belyser, er en artikkel som tar for seg endringsanalyse mellom ulike versjoner av den nederlandske arealbruk-databasen, og foreslår metoder for å håndtere skillet mellom de ulike kategoriene av endringer [17]. Etter en metodeomlegging mellom to versjoner, var mesteparten av endringene i arealklassen landbruksland (engelsk: Agricultural land) endringer som skyltes metodeomleggingen, og ikke reelle endringer i landskapet. Det viser at dersom romlig-temporale data skal brukes for å analysere endringer skjedd i den virkelige verden, må skillet mellom ulike kategorier av endringer håndteres. Dersom det ikke gjøres, risikerer man å rapportere mye større endringer enn det som faktisk er tilfelle. Mer om ulike typer endringer i kapittel 11.

For å gjøre analyser effektivt, er det nødvendig å finne fram til de dataene man trenger raskt. Det er behov for gode indekser. I en artikkel om indeksering av bitemporale romlige data [33], belyses dette, og forfatterne foreslår en indeks basert på en variant av R\*-treer. Tid-parametriserte spørringer er tema for en annen artikkel [38]. Det er spørringer som returnerer det faktiske resultatet sammen med gyldighetsperioden og endringen som gjør at resultatet slutter å være gyldig. Forfatterne tar for seg tre vanlige typer spørringer og reduserer dem til nærmeste nabo-spørringer som gjør det mulig å effektivt benytte kjente indekseringsteknikker som R-trær.

## 7.5 Oppdateringer

Oppdateringer av geodata kan foregå på ulike måter. En metode som omtales av Langran, og som fortsatt er utbredt, er å utgi kart/geografiske data i versjoner. Når et tilstrekkelig antall endringer er skjedd, eller etter en viss tid, utgir man en ny versjon av kartet (dataene). Tid er da en fastlåst parameter, låst til utgivelsestidspunktet. En annen metode er inkrementell oppdatering, der endringer foretas og utgis kontinuerlig. I henhold til Sintons rammeverk er da verken tid eller rom en låst komponent, og normalt vil heller ikke attributtene være det (selv om de ofte er kontrollert ved at de er bundet til et sett av lovlige verdier). Dermed er inkrementell oppdatering en god kandidat for det temporale GIS.

Kontinertlig oppdatering av AR5 er inkrementell oppdatering. I de nasjonale databasene for AR5 erstattes hele det gamle datasettet for en kommune av et nytt, uavhengig av hvor mange endringer som faktisk har skjedd. Dermed blir det i praksis versjonering for alle andre enn de som benytter kommunens data direkte. Det er en stor utfordring å distribuere ajourførte data effektivt.

## 7.6 Modeller for romlig-temporal informasjon

Det er hensiktsmessig å skille konseptuell modellering av data fra lagring av data. Feltmodellen og objektmodellen er konseptuelle modeller for romlige data, mens raster- og vektormodellene er lagringsmodeller.

De ulike modellene for romlig-temporale data er konseptuelle, men de fleste binder seg allikevel til å lagre data enten som raster- eller vektordata. Donna Peuquet [30] mente at den største utfordringen for utviklingen av gode geografiske informasjonssystemer, var sammenblanding mellom det konseptuelle laget og implementasjonslaget, slik at skogbestandet blir et polygon i tankene til modellereren, i stedet for å være et bestand som i databasen implementeres som et polygon. Vi modellerer kartene og dataene våre, sier Peuquet, ikke den verdenen som kartene representerer. Pequets bok kom ut for ni år siden. Fremdeles tror jeg at GIS-verdenen i stor grad lider under denne sammenblanding. Det finnes ikke anerkjente verktøy for ekte konseptuell modellering av romlige fenomener, og heller ikke gode metoder for å oversette en konseptuell modell til en implementasjonsmodell.

Det er ulike måter å modellere tid i romlige databaser på, i henhold til Langran [24]. Den enkleste, er øyeblikksbilder (se avsnitt 6.2).

En annen måte å lagre tid på, er å ha et utgangspunkt, en såkalt basistilstand (base state). Deretter lagrer man alle endringer på denne. Det kan spare betydelig mengde lagringsplass i forhold til å bruke øyeblikksbilder. Fortsatt vil forskjeller i tidsdimensjonen opptre som nye objekter i et todimensjonalt rom.

Den «ekte» måten å modellere og lagre romlig-temporale data på, i følge Langran, er å bruke en rom-tid sammensetning (space-time composite). Hver gang det skjer en endring, vil den endrede delen av et objekt frigjøres fra den uendrede delen og bli et eget objekt med sitt eget livsløp. Det er en modell som er lett å jobbe med, ulempen er at det etterhvert kan bli mange små objekter.

I en artikkel som oppsummerer litteratur om romlig-temporale databasemodeller oppsummeres 11 ulike modeller for romlig-temporale data og deres egenskaper [29]. Langrans øyeblikksbilder, rom-tid sammensetning og basistilstand med endringer er nevnt. De aller fleste modeller fokuserer på tilstand, og ikke den hendelsen som førte til tilstanden. De hendelsesbaserte modellene (event oriented models) fokuserer på hendelser, mens den historiske graf-modellen forsøker å omfavne både hendelser og tilstand. Den romlige-temporale ER-modellen nevnes, det gjør også ulike modeller basert på objekt-relasjonsmodellering. Sistnevnte bruker relasjoner mellom romlig-temporale objekter for å modellere prosesser.

Worboys trekkes i oppsummerings-artikkelen fram som den første som introduserte en objektorientert modell for romlig-bitemporal informasjon. Han definerer et romlig-temporalt objekt kalt ST\_Simplex, og ulike operasjoner på objektene [42].

Mye av litteraturen om romlig-temporale databaser omhandler objektmodellen, der et geografisk objekts identitet beholdes gjennom endringer i plassering, form og egenskaper. Endringer i form implementeres gjerne som oppdatering av et objekts geometri, samt sammenslåing eller oppsplitting av objekter. Med framveksten av objektorientert programmering, har den objektorienterte temporale databasemodellen blitt mer og mer aktuell, selv om et objekt også kan implementeres som en rad i en relasjonsdatabase. Den objektorienterte databasen lar oss følge et geografisk objekt (feature) implementert som et objekt (som i objektorientering) gjennom dets livsløp. Et objekt kan få nye egenskaper, bli ersatt, oppsplittet eller slått sammen med et annet. Objektets identitet forblir uendret gjennom endringene [1, 42, 29]. Spørsmålet for de temporale objektmodellene blir da om/når endringene på et objekt er så omfattende at det ikke lenger er riktig å kalle det det samme objektet lenger, slik at man i stedet burde ødelegge det eksisterende objektet og lage et nytt. En annen problemstilling får man når man slår sammen eller splitter opp objekter. Hvilket objekt av de oppsplittede representerer det opprinnelige, og hvilken identitet skal det sammenslåtte objektet ha?

I artikkelen som oppsummerer ulike romlige-temporale modeller [29], presenteres også modeller som håndterer objekter i bevegelse.

## 7.7 Multimedia-data

Nyere litteratur om romlig-temporale problemstillinger er like gjerne knyttet til multimediaapplikasjoner som til GIS. Modellering og indeksering av multimedia-innhold i rom (skjermen det vises på) og tid (etter hvert som avspillingen går sin gang) for å sikre effektiv gjenfinning av video-innhold er gjenstand for forskning [40, 31].

Teknikker fra multimedia-applikasjoner, kan åpenbart overføres til GIS og geodatabaser i en del sammenhenger. Objekter i kontinuerlig bevegelse er i fokus i multimedia-sammenheng. I AR5 og DMK er endringene diskrete. Visualisering, som er viktig for multimedia, er heller ikke tema for denne oppgaven.

## 7.8 Egenskaper ved arealdekke-data i et romlig-temporalt perspektiv

Heldekkende arealdekkedata, som denne oppgaven omhandler, kan sees på både som feltdata og objektdata. Når man kartlegger vegetasjon, jordtyper, arealdekke og liknende, kan man se for seg at man inndeler verden i små celler. For hver celle registrerer man for eksempel jordtypen i cellen. Gjøres cellene små nok, vil en jordtype dominere i cellen, og jordtypen kan velges som verdi for cellen. Dette er feltbasert tankegang. I arealdekkesammenheng, vil en da for eksempel registrere om en celle inneholder fulldyrka jord eller barskog.

For bonden som eier jordet, og for landbruksmyndighetene som betaler ut tilskudd og sporer maten vår «fra jord til bord» for å sikre oss trygg mat, kan det være mer aktuelt å se på jordet som et objekt med en diskret, endelig utstrekning og unik identifikator. Da kan jordet identifiseres og følges over tid. I en generell arealdekke-database

blir spørsmålene om når et objekt bør erstattes av et nytt, og hva man gjør med objektens identitet ved sammenslåinger og oppsplittinger så mange og omfattende, at det kan bli vanskelig å implementere i praksis.

Arealdekkedata kan derfor kanskje aller best representeres ved den geografiske datatypen som av Langran omtales som «categorical coverage» [24]. En norsk oversettelse, kan vær klassifisert arealdekke. Et klassifisert arealdekke er et heldekkende, ikke-overlappende polygondatasett. Konseptuelt blir et klassifisert arealdekke en blanding av felt- og objektbasert modell. ISO-standarden *ISO 19123 Schema for coverage geometry and functions* [18] spesifiserer slike data, som i standarden kalles «discrete polygon coverage» (diskret polygondekke). Endringer i slike diskrete polygondekker er endringer i attributter som igjen skaper en ny polygoninndeling. Langran testet sin modell rom-tid sammensetning på slike data [24]. DMK og AR5 er diskrete polygondekker i henhold til ISO 19123.

Skillene mellom ulike arealtyper i landskapet er ofte flytende (for eksempel mellom løvskog og barskog), og i et landskap i naturlig endring, er også grensene mellom ulike arealtyper flytende i tid. Det er vanskelig å sette fingeren på et eksakt tidspunkt for når jordbruksareal i gjengroing er blitt til ung skog. Arealdekkeinformasjon er derfor en god kandidat for bruk av flytende mengder. Fordi flytende mengder kompliserer datamodellen betydelig, og ikke minst fordi flytende mengder ikke er implementert i de mest brukte geodatabasene, har man allikevel i de aller fleste tilfeller valgt å bruke skarpe skillelinjer også der hvor dette egentlig ikke er konseptuelt riktig.

Den virkelige verden er kompleks. Ethvert klassifikasjonssystem vil derfor være en forenkling og generalisering. Det er også tilfelle med klassifisering av arealdekke. Man må ta en rekke valg om hva som skal være med, og hva som skal utelates. Valgene må tas på bakgrunn av bruken av data, økonomi, tilgjengelige ressurser og tilgjengelig teknologi.

## 8 OGC- og ISO-standarder

Open Geospatial Consortium (OGC) er en internasjonal standardiseringsorganisasjon for både kommersielle firmaer og myndighetsorganer. OGC søker å fremme åpne standarder for geografisk innhold og tjenester. Det er over 400 medlemmer i OGC, og organisasjonen har mange standarder i sin portefølje.

Flere av standardene OGC har utviklet, har senere blitt tatt opp som offisielle standarder gjennom International Organization for Standardization (ISO). Det gjelder standarden *Simple Feature Access* [28], som også er kjent som ISO 19125. Simple feature-standarden spesifiserer en felles lagringsmodell for geografiske data, hvordan geometri assosieres med geografiske referansesystemer og metoder for romlige spørringer.

At en standard er en ISO-standard betyr at den er stabil og underlagt et strengt regime for endring og medbestemmelse. Det er komitéen K176 organisert under Standard Norge som samordner og stemmer for og mot standarder for geografisk informasjon på vegne av Norge i ISO-systemet. Dersom en standard blir fastsatt som Europeisk Standard gjennom CEN, blir den også automatisk Norsk Standard. ISO har 54 publiserte standarder per oktober 2011.

ISO/TC211 er en komité under ISO som er ansvarlig for ISO sine standarder innenfor geografisk informasjon. ISO/TC211 har sterk norsk innflytelse med blant annet norsk leder (Olaf Østensen) og sekretariat i regi av Standard Norge, i tillegg til norsk deltakelse i mange ulike arbeidsgrupper. Mange ISO-standarder blir som sagt europeiske standarder og dermed automatisk norske standarder. I denne sammenhengen er *ISO 19108:2002 – Geographic information – Temporal schema* [36], jobbet fram av TC211, spesielt interessant. Den spesifiserer hvordan tidsaspektet kan integreres med geografisk informasjon.

### 8.1 ISO 19125 Geographic information – Simple feature access

Standarden er todelt, den første delen heter «Common Architecture», og er implementasjonsuavhengig, den andre heter «SQL Option». OGC har også standardisert implementasjon i CORBA og OLE/COM.

Del 1 av standarden beskriver geometrimodell, representasjon av geometrimodell, relasjoner mellom geometrier og operasjoner på geometrier. Geotrimodellen består av et basis geometriobjekt, som «Point», «Curve», «Surface» og «GeometryCollection» er subclasser av. Hvert geometrisk objekt er assosiert med et referansesystem, og kan representeres som en «well-known text» eller en «well-known binary».

Standarden forholder seg ikke til topologi. Felles grenselinjer mellom polygoner er lagret to ganger, og relasjoner mellom objekter må regnes ut ved behov. Romlige

relasjoner i Simple feature-standarden er for eksempel «crosses», «touches» og «overlaps», som alle returnerer «true» eller «false». Relasjonene bygger på «the dimensionally extended nine intersection matrix» som beskrevet av Clementini [11]. Spesifiserte operasjoner er blant annet «intersection», «union» og «difference», som alle returnerer en ny geometri. Operasjonen «distance» er den eneste som returnerer en tallverdi. I figur 8.1 er en UML-modell for klassen «Geometry» med alle egenskaper og metoder. For detaljert spesifikasjon av geometrimodell, representasjon og metoder, henvises til selve standarden [28].

Del 2 av standarden spesifiserer hvordan standarden skal implementeres i SQL. Den romlige utvidelsen til relasjonsdatabasen PostgreSQL, PostGIS, er et eksempel på en geodatabase som har implementert standarden i SQL. Det gjør det mulig å både å lagre geometri og gjøre romlige spørringer i SQL i en PostGIS-database uten komplisert programmering og intrikate GIS-verktøy. Standarden fortjener mye av æren for at geografisk informasjon, inkludert geometri, i dag kan lagres og brukes direkte i vanlige, generelle databasesystemer gjennom utvidelser til den tradisjonelle relasjonsmodellen.

## 8.2 NS-EN ISO 19108 Geografisk informasjon - Modell for å beskrive tidsaspekter

Standarden *ISO 19108:2002 Geographic Information - Temporal schema* ble europeisk standard i 2005, og dermed også en norsk standard. Som norsk standard fikk den navnet *NS-EN ISO 19108 Geografisk informasjon - Modell for å beskrive tidsaspekter*. Selve standarden er imidlertid ikke oversatt til norsk, det er den engelske versjonen av den europeiske standarden som gjelder i Norge.

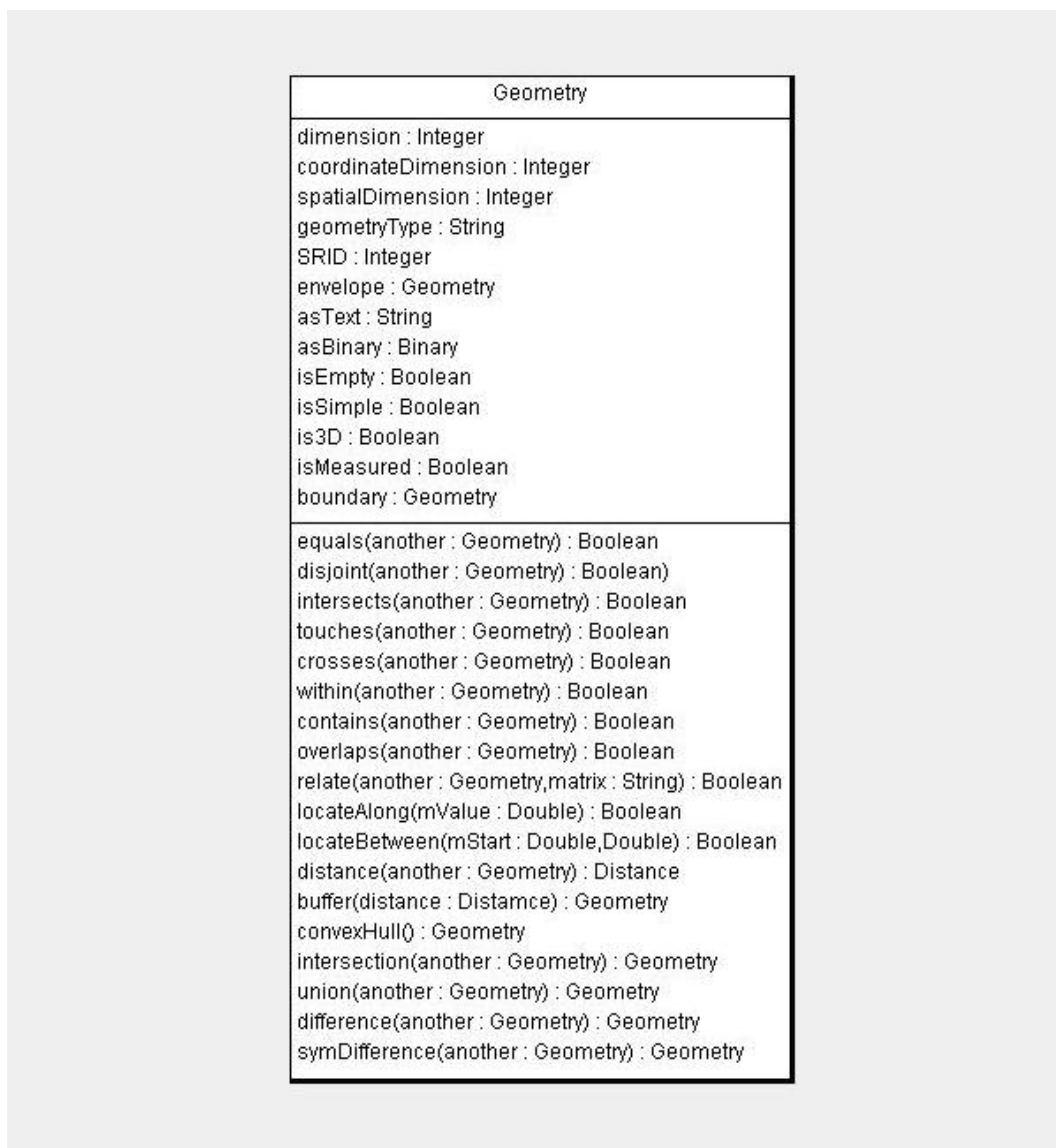
Standarden beskriver nødvendige konsepter for å beskrive tidsaspekter ved geografisk informasjon som attributter. Standarden konsentrerer seg kun om gyldig tid, siden målet med standarden er å definere temporale karakteristikk ved geografisk informasjon som en abstraksjon av den virkelige verden. Konseptene som beskrives, er geometriske og topologiske temporale objekter, temporale referansesystemer, temporal posisjon og hvordan dette kan benyttes i geografiske informasjonssystemer.

Dette avsnittet er i sin helhet et sammendrag av standarden, og alle figurer er gjengivelser av figurer i standarden. Jeg vil konsentrere meg om temporale geometriske primitiver framfor topologiske primitiver, siden det er mest relevant for denne masteroppgaven.

### 8.2.1 Temporale geometriske og topologiske objekter

De temporale objektene (TM\_Object) som defineres, er enten enkle (engelsk: primitive) eller komplekse. En TM\_Primitive, kan være et geometrisk eller et topologisk primitiv, henholdsvis TM\_GeometricPrimitive og TM\_TopologicPrimitive. Et komplekst objekt, er alltid et kompleks (TM\_TopologicalComplex) av flere topologiske primitiver. Se figur 8.2 for UML-diagram over temporale objekter i ISO 19108.

Et geometrisk primitiv er enten et øyeblikk (TM\_Instant) eller en periode (TM\_Period). TM\_Instant har kun ett attributt: temporal posisjon av datatype



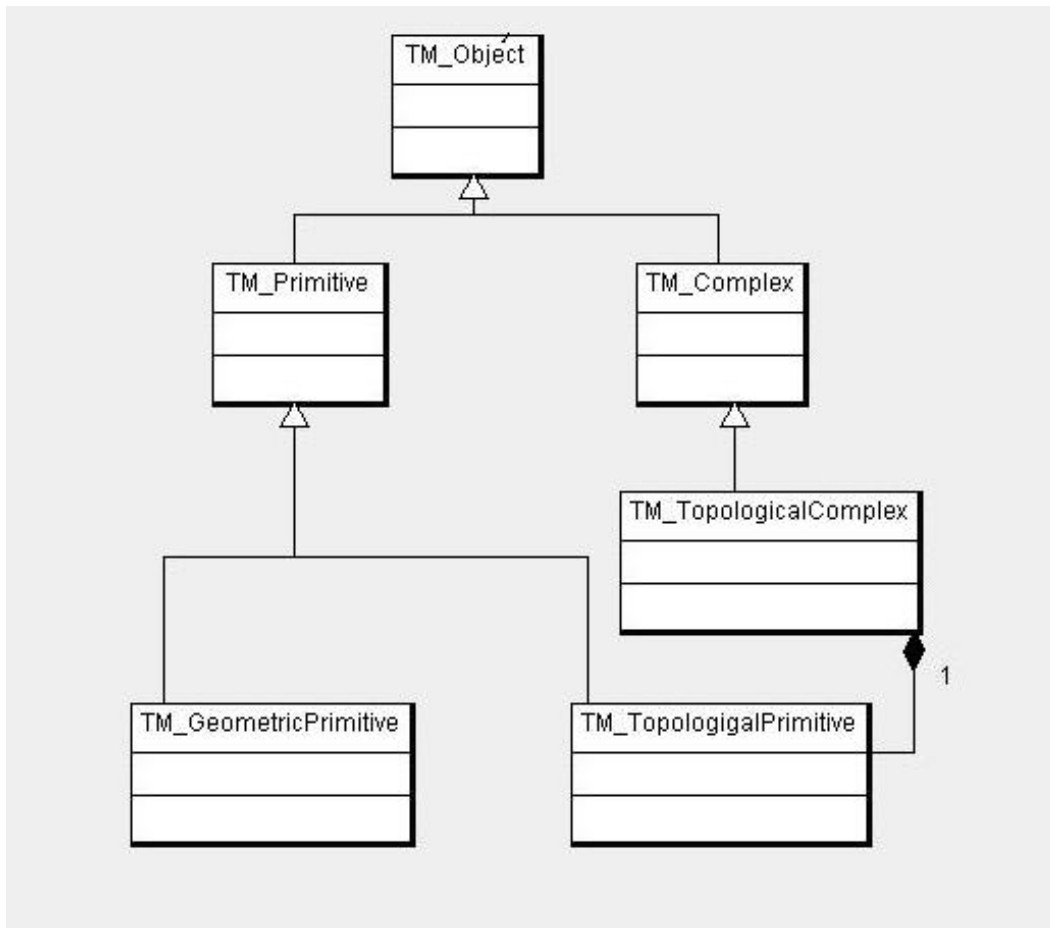
Figur 8.1: UML-modell av klassen Geometry i ISO19125

TM\_Position. En temporal posisjon kan være en direkte tidsangivelse eller en angivelse av posisjon i forhold til noe annet. En periode starter og slutter i et øyeblikk, men dette kan også være ukjent, eller ligger i framtiden, dvs være NULL.

Klassen TM\_GeometricPrimitive har operasjonene avstand (distance) og lengde (length). Begge disse er av datatype TM\_Duration. Klassen TM\_Primitive har en operasjon: relativePosition, som avgjør den relative rekkefølgen av temporale primitiver. Figur 8.3 viser et UML-diagram over primitivene og deres operasjoner og relasjoner. Datatypen TM\_Duration er enten angitt på formatet for varighet i henhold til ISO 8601, eller som et tidsintervall etter ISO/IEC 11404.

De topologiske primitivene er enten kanter (TM\_edge) og eller noder (TM\_node). Disse er igjen realisert som henholdsvis periode (TM\_Period) og øyeblikk (TM\_Instant).

Enhver temporal posisjon må knyttes til et temporalt referansesystem. Det er den



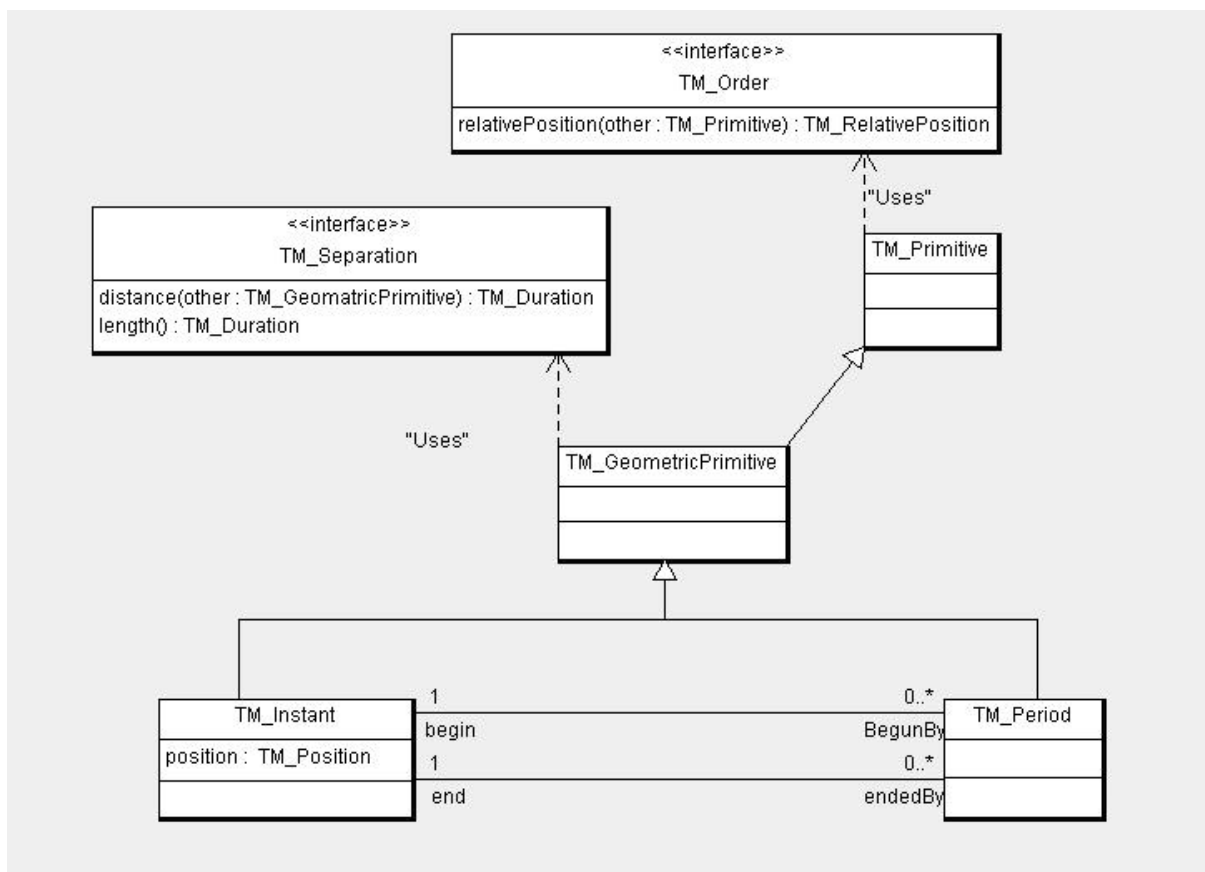
Figur 8.2: Temporale objekter

Gregorianske kalenderen og Coordinated Universal Time (UTC) som er de primære referansesystemene ved utveksling av geografisk informasjon

### 8.2.2 Tid og komponenter i geografisk informasjon

Geografisk informasjon kan på et metanivå modelleres som geografiske features med attributter operasjoner og relasjoner i henhold til ISO 19109. Både features, attributter, operasjoner og relasjoner kan ha et tidsaspekt ved seg. En hendelse skal representeres med `TM_Instant` eller `TM_Node`. En hendelse er noe som forekommer i et gitt øyeblikk. Selv om enhver hendelse i praksis opptar et lite intervall, omtaler vi det som et øyeblikk når tiden hendelser tar, er kort sammenliknet med måleskalaen. En tilstand er derimot en eller flere egenskaper ved en feature som varer i en periode, og representres med `TM_Period` eller `TM_Edge`, som begge representerer intervaller med en begynnelse og slutt.





Figur 8.3: Temporale geometriske primitiver

### 8.3 Andre aktuelle standarder

ISO 19123:2005 definerer et konseptuelt skjema for arealdekker («coverages»). Det skilles mellom kontinuerlige og diskrete arealdekker.

ISO/IEC 13249-3:2011, også kjent som *SQL/MM Spatial* er en utvidelse av Simple Feature-standarden, blant annet med topologi og romlige brukerdefinerte typer.

# 9 Tilgjengelig teknologi

I denne masteroppgaven vil jeg komme med forslag til hvordan Skog og landskap bedre kan forvalte historikk i sine markslagsdata. Jeg ønsker altså å komme med et forslag som lar seg realisere i praksis i et reelt informasjonssystem for nasjonale data. Da er det både fornuftig og mest hensiktsmessig å forholde seg til den teknologi og kompetanse Skog og landskap allerede har, i hvert fall så langt det er mulig. Dette kapittelet redegjør for tilgjengelig database- og GIS-teknologi ved instituttet.

## 9.1 Geodatabasesystemer i bruk ved Skog og landskap

Skog og landskap benytter både Oracle/ArcSDE og PostgreSQL/PostGIS for å lagre geodata. Oracle og PostgreSQL er relasjonsdatabasesystemer med objektorienterte konsepter implementert, såkalte objekt-relasjonsdatabaser. PostGIS og ArcSDE er romlige utvidelser til relasjonsdatabaser. PostGIS er fri programvare, mens ArcSDE er proprietær programvare som eies av ESRI. Oracle og ArcSDE sammen er et geodatabasesystem, det samme er PostgreSQL og PostGIS. ArcSDE kan benyttes sammen med andre relasjonsdatabasesystemer enn Oracle, men det er ikke brukt hos Skog og landskap. Skog og landskap bruker heller ikke Oracle Spatial og Oracle Locator, som er Oracles utvidelser for lagring av geografiske data.

Det siste året har PostgreSQL/PostGIS vært det foretrukne geodatabasesystemet ved instituttet. Årsakene er flere: PostgreSQL er, for Skog og landskap som ikke har en databaseadministrator på fulltid, enklere å drifte enn Oracle. PostGIS implementerer Simple feature-standarden for SQL. Det gjør det lett å skrive SQL-spørringer mot dataene, både for å modifisere data og å hente dem ut. ArcSDE implementerer også simple feature dersom man bruker geometritypen «ST\_Geomtry», men det har ikke vært standard geometritype før i de to siste versjonene av ArcSDE. Derfor har ikke Skog og landskap brukt «ST\_Geomtry» i sine ArcSDE-databaser. I tillegg er PostgreSQL og PostGIS fri programvare som gir en mer fleksibel hverdag. Trenger man en databaseserver til, er det bare å sette den opp.

AR5 er også lagret i Norkart sitt geodatabasesystem Quadri Map Server (QMS). Serveren driftes av Statens kartverk. Alle Geovekst-dataene er lagret der. Tilgangen til data i QMS-basen er begrenset til funksjonalitet tilgjengelig gjennom et eget API. QMS-basen er derfor ikke aktuell for det praktiske arbeidet i denne masteroppgaven.

## 9.2 Desktop-GIS og datamodelleringsverktøy

Av desktop-GIS er Fysak, ArcGIS, QGIS, Udig og gvSig i generell bruk ved Skog og landskap. Fysak er laget spesielt for å arbeide med SOSI-filer. ArcGIS er ESRI sitt desktop-GIS, og er godt integrert med ArcSDE. QGIS, Udig og gvSig er alle fri programvare.

Instituttet har ikke standardisert verktøy for datamodellering.

## 9.3 Støtte for tid i tilgjengelige databasesystemer

Både Oracle og PostgreSQL har støtte for SQL, som har følgende datatyper for dato og tid:

Datatype	Beskrivelse
DATE	for datoer, f.eks 28.10.2011
TIME	for klokkeslett, f.eks 15:28:34.
TIME WITH TIME ZONE / TIMESTAMP	det samme som TIME, men med informasjon om tidssone
TIMESTAMP	DATE og TIME satt sammen i en variabel (f.eks 2011-05-03 15:51:36).
TIMESTAMP WITH TIME ZONE / TIMESTAMPTZ	det samme som TIMESTAMP, men med informasjon om tidssone.
INTERVAL	Tidsintervall med starttidspunkt og sluttidspunkt. Implementeres oftest ikke som en datatype for attributter, kun i operasjoner og funksjoner

Tabell 9.1: SQL-datatyper for tid

SQL har flere funksjoner for å lage datatyper for tid: TO\_DATE, TO\_TIME og TO\_TIMESTAMP. Det finnes funksjoner for å trekke ut deler av en variabel (for eksempel minutter eller måneder). Dessuten er det definert flere operasjoner og funksjoner på tidsvariable, for eksempel «overlaps», «day of week» og «epoch» (antall sekunder siden 01.01.1970). Det er også definert en funksjon som returnerer aktuell systemdato og tidspunkt.

SQL mangler allikevel viktige elementer for å støtte tid på en god måte. Både en datatype «period» og funksjoner for alle «Allen relationships» (se også avsnitt 6.1) og gode temporale indekseringsmetoder ville gjort relasjonsdatabasen og SQL bedre for til håndtere tid [35, 2]. Da hadde man sluppet å programmere alt som har med tid å gjøre i egne API-er over databasen, eller direkte i applikasjonene som håndterer og bruker dataene.

### 9.3.1 Oracle

Oracle gir mulighet for lagring av geometri gjennom Oracle Locator/Oracle Spatial, som implementerer Simple feature-standarden. Oracle Locator følger med Oracle Enterprise Edition, mens Oracle Spatial er mer avansert og må kjøpes som et tillegg. Skog og landskap har lisens på Oracle Locator, men benytter den ikke.

Oracle implementerer SQL-datatypene timestamp, date, og interval [53]. Timestamp kan være knyttet til tidssone.

Fra Oracle 9i har transaksjonstid vært automatisk tilgjengelig gjennom flashback-spørringer. Denne funksjonaliteten ble utvidet i Oracle 10g med støtte for gyldig tid, bi-temporale tabeller og sekvensielle primærnøkler, fremmednøkler og utvalg/spørringer gjennom Workspace Manager. Måten det er gjort på, likner veldig på det som i sin tid ble foreslått i temporal SQL [59].

For å få klokkeslett og dato for «akkurat nå», benyttes «sysdate». Det er en funksjon som ikke tar noen parametre, og returnerer aktuell dato og klokkeslett som en DATE datatype [53]:

```
SQL> SELECT sysdate FROM dual;
```

```
SYSDATE
-----
04.11.2011
```

### 9.3.2 PostgreSQL/PostGIS

PostgreSQL støtter alle SQL-datatypeer for dato og tid: timestamp, date, time og interval. Med unntak av date, som har oppløsning på ett døgn, har alle de nevnte datatypene oppløsning på 1 mikrosekund ( $1 * 10^{-6}$  s). Timestamp og time kan angis med eller uten tidssone. Definisjonene av de ulike datatypene, funksjonene og operatorene er konforme med SQL-standarden. Mer om dette, finnes i PostgreSQL-dokumentasjonen [57].

For å få aktuell systemdata og -tidspunkt, kan en benytte funksjonen «now()»:

```
postgres=# SELECT now();
          now
-----
2011-11-04 13:00:45.827+01
(1 row)
```

PostGIS installeres enkelt i en eksisterende PostgreSQL-database, og gir tilgang til datatyper for geometri, referansesystemer og funksjoner for å manipulere og hente ut data som spesifisert i Simple feature-standarden for SQL. PostGIS er opprinnelig utviklet av det kanadiske firmaet *Refractions Research*, og er utgitt under GNU General Public License. Dokumentasjonen for PostGIS er tilgjengelig på Refractions Research sine nettsider for PostGIS [54].

### 9.3.3 ArcSDE

ArcSDE er firmaet *ESRI* sin geodatabase, og kommer som en del av ArcGIS Server-pakken. Skog og landskap har benyttet ArcSDE siden 1997.

I ArcSDE finnes et konsept som kalles versjonering. Versjoneringen har som formål å gjøre det mulig for flere samtidige brukere å editere samme datasett samtidig gjennom lange transaksjoner. Konseptet består av en basisversjon, samt såkalte delta-tabeller for slettinger og innsetninger av data. En oppdatering håndteres som en sletting etterfulgt av en innsetting [47]. Det vil også være mulig å bruke versjoneringsmekanismen til å holde styr på tid i data. Øyeblikksbilder kan for eksempel lages ved å lage en ny

versjon av et datasett. Mer kompleks tidshåndtering enn øyeblikksbilder med ArcSDE-versjonering, kan fort bli tungvindt fordi det vil bli ekstremt mange versjoner (en for hver oppdatering). En annen ulempe er at versjonerte data blir vanskelig tilgjengelig gjennom andre verktøy enn ArcSDE og ArcGIS, da metoden er proprietær. Skog og landskap benytter ikke versjonering i ArcSDE. Begrunnelsen for det er kompleksiteten ved versjoneringa, at hele databasen versjoneres dersom et layer versjoneres (selv om man ikke editerer dataene), og at MapServer (i hvert fall i tidligere versjoner), ikke klarer å vise versjonerte kartlag fra ArcSDE.

ArcSDE har et API i C og Java som implementerer Simple feature-standarden. Fra versjon 9.3 av ArcSDE er datatypen «ST\_Geometry» standard for ArcSDE over Oracle. Den gir tilgang til SQL-spørringer basert på simple feature [46]. De fleste av Skog og landskaps datasett er etablert før «ST\_Geometry» ble et mulig valg, og bruker derfor ikke denne. Konsekvensen er at det ikke er mulig å skrive romlige SQL-spørringer mot dataene uten å konvertere dem først. Konverteringsjobben er tidkrevende, og vil først bli gjort ved overgang til ny databaseserver.

## 9.4 Teknologi valgt i denne oppgaven

PostGIS er valgt framfor ArcSDE og ArcGIS som primærverktøy for de praktiske eksperimentene i denne masteroppgaven. Både PostGIS og ArcSDE/ArcGIS lagrer geometrier som enkle geometrier uten topologi. Begge gir gode muligheter for scripting og programmering. I tillegg implementerer PostGIS, i motsetning til ArcSDE (med de datatyper for geometri som Skog og landskap bruker), romlig SQL. PostGIS er også fri programvare, som spiller godt på lag med annen fri programvare Skog og landskap har gode erfaringer med.

Skulle eksperimentene vært gjort i ArcSDE, ville jeg vært helt avhengig av å benytte en lisens på den kostbare programvaren ArcGIS fra ESRI. Skog og landskap har gode erfaringer med å gjøre forvaltningsoppgaver og analyseoppgaver i PostGIS, både ytelsesmessig og kvalitetsmessig er PostGIS på høyde med eller bedre enn ArcSDE. I denne oppgaven forholder jeg meg derfor til ArcSDE kun som et datalager jeg henter data ut fra, alt praktisk arbeid vil foregå på kopidata i PostGIS.

Jeg velger å benytte ArcGIS for å eksportere originaldata fra ArcSDE, ellers vil jeg benytte QGIS for å se på data i PostGIS. QGIS er enkel å koble til PostGIS.

Som UML-verktøy har jeg valgt å bruke ArgoUML. Det er fri programvare som implementerer UML 1.4 og kjører på enhver Java-plattform. Se programvarens Internett-side [44] for nedlasting og dokumentasjon.

# 10 Tilgjengelige databaser og datasett

## 10.1 Økonomisk kartverk

Både gamle kartblad av økonomisk kartverk (ØK) og nyere N5-raster generert fra oppdaterte, detaljerte vektordata finnes på georeferert rasterformat. Begge er tilgjengelig blant annet som en Web Map Service (WMS) fra Statens kartverk. Da disse kartene kun er å betrakte som en scannet og georeferert variant av det analoge økonomisk kartverket, har det ikke stor verdi i et geografisk informasjonssystem, i hvert fall ikke for automatiske analyser og prosesser. Det kan imidlertid ha verdi for manuell tolkning, inspeksjon og kontroll. Det er ikke slik at gammelt ØK alltid er det opprinnelige ØK, alderen på kartene varierer fra kartblad til kartblad. I denne oppgaven vil ØK og N5-raster ikke ha noen direkte nytte, men det er greit å ha det tilgjengelig for manuell inspeksjon dersom man mistenker feil i overgangsprosessen til digitale kart.

## 10.2 Digitalt Markslagskart

Det nasjonale datasettet digitalt markslagskart (DMK) er tilgjengelig i en nasjonal, sømløs vektor-database med polygoner lagret etter simple feature-modellen i Oracle/ArcSDE ved Skog og landskap. Det finnes også en kopi av datasettet i PostGIS.

Datasettet har ingen håndtering av tid annet enn et attributt tilsvarende det tidligere SOSI-elementet ..DATO. Det er en viss risiko for at dette elementet har blitt brukt litt forskjellig opp gjennom tiden, men det vil normalt bety enten registreringsdato, flybilledato eller oppdateringsdato.

DMK ble til ved en digitalisering av markslaget i økonomisk kartverk, som beskrevet i kapittel 3. SOSI-modellen for DMK er gjengitt fra SOSI-standarden [62] i figur 10.1. I denne oppgaven er de viktigste egenskapene i DMK arealtilstand (ATIL i SOSI-filer og databaser) og jordklassifikasjon (JORD i SOSI-filer og databaser).

### 10.2.1 Arealtilstand

Arealtilstanden definerer naturlige terrengetyper (for eksempel skog og myr) og typer opparbeidet areal (for eksempel dyrka jord og beite). Arealtilstanden er det viktigste kriteriet for inndeling av markslag, og er en obligatorisk egenskap for alle markslagsfigurer. Kodeverdiene for arealtilstand i DMK er beskrevet i tabell 10.1.

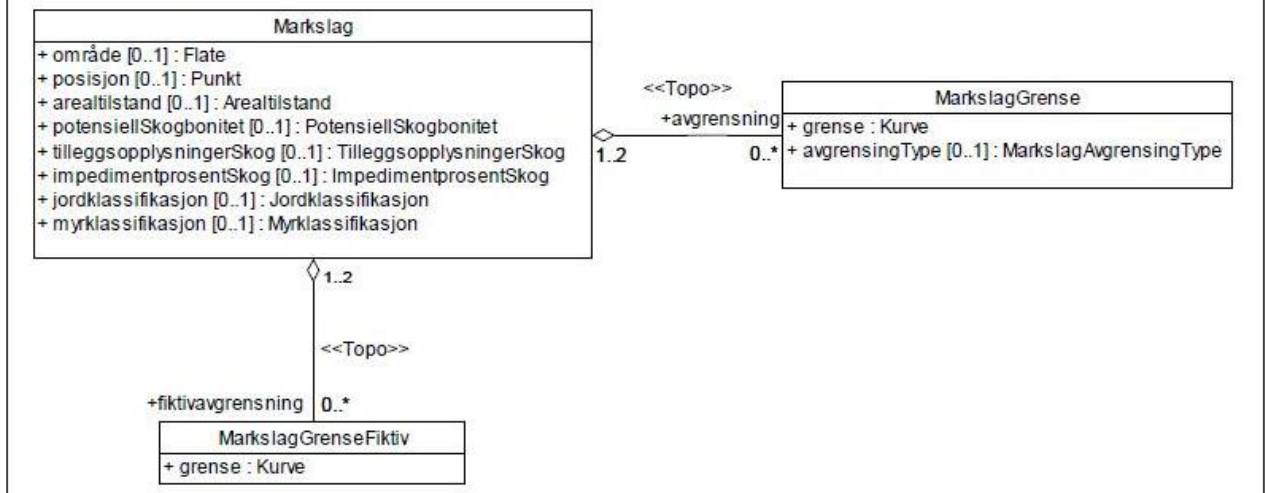
## 6 Applikasjonsskjema

### 6.1 Introduksjon

Applikasjonsskjema viser objekttypen Markslag med tilhørende avgrensingsobjekter.

### 6.2 Grafisk visning av applikasjonsskjema

#### 6.2.1 Objekttyper



Figur 10.1: SOSI UML-modell for DMK

### 10.2.2 Jordklassifikasjon

Jordklassifikasjonen klassifiserer jordbruksareal etter driftsforhold, og dyrkingsjord etter driftsforhold og jordegenskaper. Jordklassifikasjonen forteller også om et område er dyrka eller dyrkbar myr. Det er de øvste 30 cm av jordprofilen som vurderes når dyrkingsjorda klassifiseres.

#### Driftsforhold

Opplysningene om driftsforhold klassifiserer arealer i lettbrukt, mindre lettbrukt og tungbrukt, og tar hensyn til terrenghelling, form/arronding og størrelse. Tungbrukt brukes ikke om dyrkingsjord.

#### Jordegenskaper for dyrkingsjord

Jordegenskapene forteller hvor skikket et areal er for dyrking, og gir informasjon om stein- og blokkinnhold, drenerings- og vanningsbehov.

Jordklassifikasjonen er egentlig fire ulike egenskaper fra ØK satt sammen til en:

- driftsforhold
- stein- og blokkinnhold

Koder for arealtilstand (ATIL) i DMK		
KODE	NAVN	BESKRIVELSE
1	Uklassifisert/ukjent arealtilstand	Vart nytta når ein ikkje kunne kontrollere signatur ved hjelp av flybilette eller på andre måtar. Ikkje lenger i bruk, men kan finnast i eldre datasett.
11	Myr	Areal med minst 30 cm tjukt torvlag som på overflata har preg av myr.
12	Myr m/barskog	Myr som også held kravet til barskog.
13	Myr m/blandingsskog	Myr som også held kravet til blandingsskog.
14	Myr m/lauvskog	Myr som også held kravet til lauvskog.
15	Kombinasjon myr/fastmark	Areal der det vekslar så mykje mellom myr og fastmark at oppdelinga vert for sterk. Ikkje lenger i bruk, men kan finnast i eldre data.
16	Grunnlendt myr	Areal der det vekslar så mykje mellom grunnlendt mark og myr at oppdelinga vert for sterk. Ikkje lenger i bruk, men kan finnast i eldre data.
21	Fulldyrka jord	Areal som er dyrka til vanleg pløvedjupn, og som kan nyttast til åkervekstar eller til eng som kan fornyast ved pløying.
22	Overflatedyrka jord	Areal som for det meste er rydda og jamna i overflata, slik at maskinell hausting er mogleg.
23	Innmarksbeite	Innmarksareal som kan nyttast som beite, men som ikkje kan haustast maskinelt. Minst 50 % av arealet skal vere dekt av grasartar. Nemninga var tidlegare Gjødsla beite (endra f.o.m. 2001).
24	Barskog	Areal som held kravet til skog (minst 6 tre pr. daa som er eller kan bli 5 m høge), og der minst 50 % av skogdekt areal er dekt av bartre.
25	Blandingsskog	Areal som held kravet til skog (minst 6 tre pr. daa som er eller kan bli 5 m høge), og der 20-50 % av skogdekt areal er dekt av bartre.
26	Lauvskog	Areal som held kravet til skog (minst 6 tre pr. daa som er eller kan bli 5 m høge), og der mindre enn 20 % av skogdekt areal er dekt av bartre.
27	Anna jorddekt fastmark	Fastmark som ikkje er skog- eller jordbruksareal, og der meir enn 50 % har større jorddjupn enn 30 cm.
28	Grunnlendt mark	Areal der meir enn 50 % har mindre jorddjupn enn 30 cm, men som ikkje kan klassifiserast som fjell i dagen.
29	Fjell i dagen	Areal der meir enn 50 % er bert fjell og mindre enn 10 % har jord djupare enn 30 cm.
31	Blokkdekt mark	Areal der overflata i hovudsak er dekt med blokker. Erstattar tidlegare Ur, steinrøys (endra f.o.m. 2001).
32	Grustak	Ikkje systematisk registrering. og det kan t.d. også ha vore nytta Anna jorddekt fastmark på grustak. Ikkje lenger i bruk, men kan finnast i eldre data.

Tabell 10.1: Koder for egenskapen arealtilstand i DMK [7].

- drenerings- og vanningsbehov
- dyrka/dyrbar myr

Denne sammenblandinga av flere eigenskapar i ett og samme attributt, har nok sin bakgrunn i datamodelleringa. Da man gjorde digitaliseringsjobben, var det markslag i økonomisk kartverk som ble modellert, og ikke markslag i den virkelige verden [64, 27]. Uansett årsak, gjør denne sammenblandinga det vanskelig å skrive riktige spørringer mot dataene, og sannsynligheten for å gjøre feil er unødvendig stor. Skog og landskap har laget en normalisert versjon av markslagsdatabasen, men den er dessverre ikke tatt i bruk. Derfor er det opprinnelig (ikke-normalisert) DMK som vil bli brukt i denne oppgaven, men enkelte av funksjonene som ble laget til den normaliserte versjonen, vil bli brukt til omkoding av dataene.

Kodeverdiene for jordklassifisering er beskrevet i tabellen 10.2.



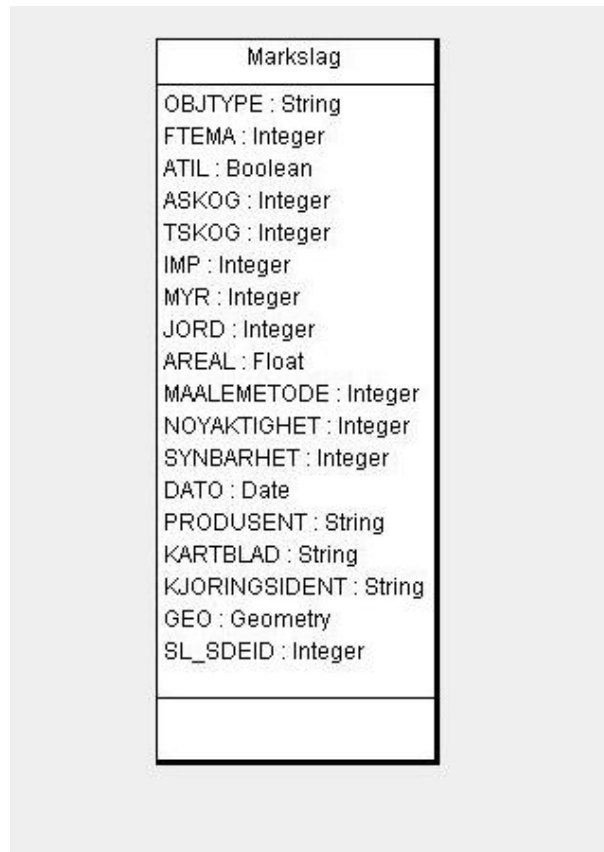
Egenskapen jordklassifisering (JORD) i DMK	
KODE	NAVN
0	Ikkje registrert
23	Dyrka myr
31	Lettbrukt, fulldyrka jord og dyrkingsjord
32	Lettbrukt, sjølvdrenerert dyrkingsjord
33	Lettbrukt, tørkesvak dyrkingsjord
34	Lettbrukt dyrka myr og dyrkingsmyr
41	Lettbrukt, blokkrik dyrkingsjord
42	Lettbrukt, blokkrik, sjølvdrenerert dyrkingsjord
43	Lettbrukt, blokkrik, tørkesvak dyrkingsjord
44	Lettbrukt, blokkrik, dyrkbar myr
51	Lettbrukt, svært blokkrik dyrkingsjord
52	Lettbrukt, svært blokkrik, sjølvdrenerert dyrkingsjord
53	Lettbrukt, svært blokkrik, tørkesvak dyrkingsjord
54	Lettbrukt, svært blokkrik, dyrkbar myr
61	Mindre lettbrukt fulldyrka jord og dyrkingsjord
62	Mindre lettbrukt, sjølvdrenerert dyrkingsjord
63	Mindre lettbrukt, tørkesvak dyrkingsjord
64	Mindre lettbrukt dyrka myr og dyrkingsmyr
71	Mindre lettbrukt, blokkrik dyrkingsjord
72	Mindre lettbrukt, blokkrik, sjølvdrenerert dyrkingsjord
73	Mindre lettbrukt, blokkrik, tørkesvak dyrkingsjord
74	Mindre lettbrukt, blokkrik dyrkbar myr
81	Mindre lettbrukt, svært blokkrik dyrkingsjord
82	Mindre lettbrukt, svært blokkrik, sjølvdrenerert dyrkingsjord
83	Mindre lettbrukt, svært blokkrik, tørkesvak, dyrkingsjord
84	Mindre lettbrukt, svært blokkrik, dyrkbar myr
91	Tungbrukt, fulldyrka jord og dyrkingsjord, «restjord»

Tabell 10.2: Koder for egenskapen jordklassifisering i DMK [7].

### DMK-databasen: implementasjon, historikk og skranker

I databasene for DMK, har gamle data blitt slettet før nye data for samme område har blitt lagt inn. Databasene ivaretar således ikke historikk. Gamle versjoner av SOSI-filer for en del kommuner finnes på filsystem og gamle backup taper hos Skog og landskap, men det har ikke vært noen systematikk i det å ta vare på dem. Jeg vil derfor i denne oppgaven forholde meg kun til den nasjonale databasen. Den har et spenn i datering av polygoner fra 1956 til 2009. Databasen er tiggjengelig i Oracle/ArcSDE og PostGIS, og har datamodell som vist i figur 10.2.

Det er ingen skranker på dataene i DMK-databasene hos Skog og landskap, bortsett fra at intern primærnøkkel ikke kan være NULL. En har også valgt å kun lagre polygoner i ArcSDE og PostGIS. Hoved-layeret for DMK i ArcSDE heter MARKS-LAG\_FLATE, og er laget i ArcSDE. Det er kun grunnleggende funksjonalitet i ArcSDE som er benyttet (dvs funksjonalitet som fantes i SDE versjon 3). Det betyr at nyere og mer avanserte muligheter som kontroll av topologi, domenesjekk etc. ikke er benyttet. Egenskapene i layeret er implementert som kolonner med følgende datatyper



Figur 10.2: DMK har en enkel datamodell i Skog og landskaps nasjonale database

og beskrivelser i Oracle-databasen:

MARKSLAG_FLATE	Navn	Null?	Type	Beskrivelse
	OBJTYPE		VARCHAR2(40)	SOSI objekttype
	FTEMA		NUMBER(4)	SOSI ftema.kode
	FTEMA1		NUMBER(4)	SOSI ftema for multiple ftema
	ATIL		NUMBER(2)	Arealtilstand
	ASKOG		NUMBER(2)	Potensiell skogbonitet
	TSKOG		NUMBER(2)	Tilleggsklassifikasjon skog
	IMP		NUMBER(2)	Impedimentsprosent
	MYR		NUMBER(2)	Informasjon om myr/torvmark
	JORD		NUMBER(2)	Informasjon om jordklassifikasjon
	H40		NUMBER(2)	Skogbonitet etter h40-systemet
	AREAL		NUMBER(18,5)	Polygonets areal i m2
	MAALEMETODE		NUMBER(2)	SOSI kvalitetsinformasjon
	NOYAKTIGHET		NUMBER(5)	SOSI kvalitetsinformasjon
	SYNBARHET		NUMBER(2)	SOSI kvalitetsinformasjon
	DATO		VARCHAR2(8)	Dato for hva?
	PRODUSENT		VARCHAR2(40)	SOSI produsent
	KARTBLAD		VARCHAR2(9)	ØK-kartblad
	KJORINGSIDENT		VARCHAR2(40)	Tidsstempel for innlegging i base
	GEO		NUMBER(38)	Fremmednøkkel til geometritabell
	SL_SDEID	NOT NULL	NUMBER(9)	Intern primærnøkkel

I tillegg til selve DMK-kartet, finnes et omfattende kodeverk, som holder styr på de over 3700 lovlige kombinasjonene av arealtilstand, askog, tskog, jord og myr. Tar man med det mer detaljerte kodesystemet for skogbonitet, kalt H40-koder, får man

i overkant av 23000 lovlige kombinasjoner. H40-systemet er kun brukt på 5 % av skogarealet i DMK. Det gir mer detaljert informasjon om skogens bonitet enn de vanlige boniteringskodene i ASKOG-egenskapen.

Det er ingen fremmednøkkler eller andre skranker mellom MARKSLAG\_FLATE og kodeverket. Årsaken til dette, er at data med feil bruk av koder, ønskes lagt inn i databasen slik at feilene kan plukkes opp der og rettes. Mange typer feil er enklere å plukke opp i en relasjonsdatabase enn i kommunevise filer. Ulempen med å «slippe inn» data med feil, er at databasen kan inneholde feil som av en eller annen grunn ikke er blitt rettet opp, for eksempel fordi man ikke vet hvilket markslag som var ment brukt for en markslagsfigur med ulovlig markslagskode.

Datamodellen for kodeverket for DMK er vist i figur 10.3. H40-egenskapen for skogbonitet er ikke tatt med i modellen, da alle polygoner som har denne egenskapen, også har bonitetskode etter det ordinære systemet.

I tillegg til lovlige kombinasjoner av egenskaper, holder kodeverket også styr på omkoding til ulike temakart (kalles standarder i kodeverket), hvorav dyrkingsjord er det ene. Disse avledningene er beskrevet i NIJOS-dokumentet [7]. Alle markslag skal ha avledningsregler til alle definerte temakart, slik at temakartene i utgangspunktet har samme dekning som grunnkartet. Det er spesifisert 13 ulike temakart. En markslagsfigur med ulovlig markslagskode, vil ikke kunne omkodes til de ulike temakartene i dagens system. Det er ordinært system for skogbonitet (kolonne ASKOG) som alltid brukes ved omkoding til temakart, ikke H40-koden for de områdene som har denne.

### 10.2.3 Temakartet dyrkingsjord skal inngå i TilgjengeligJord

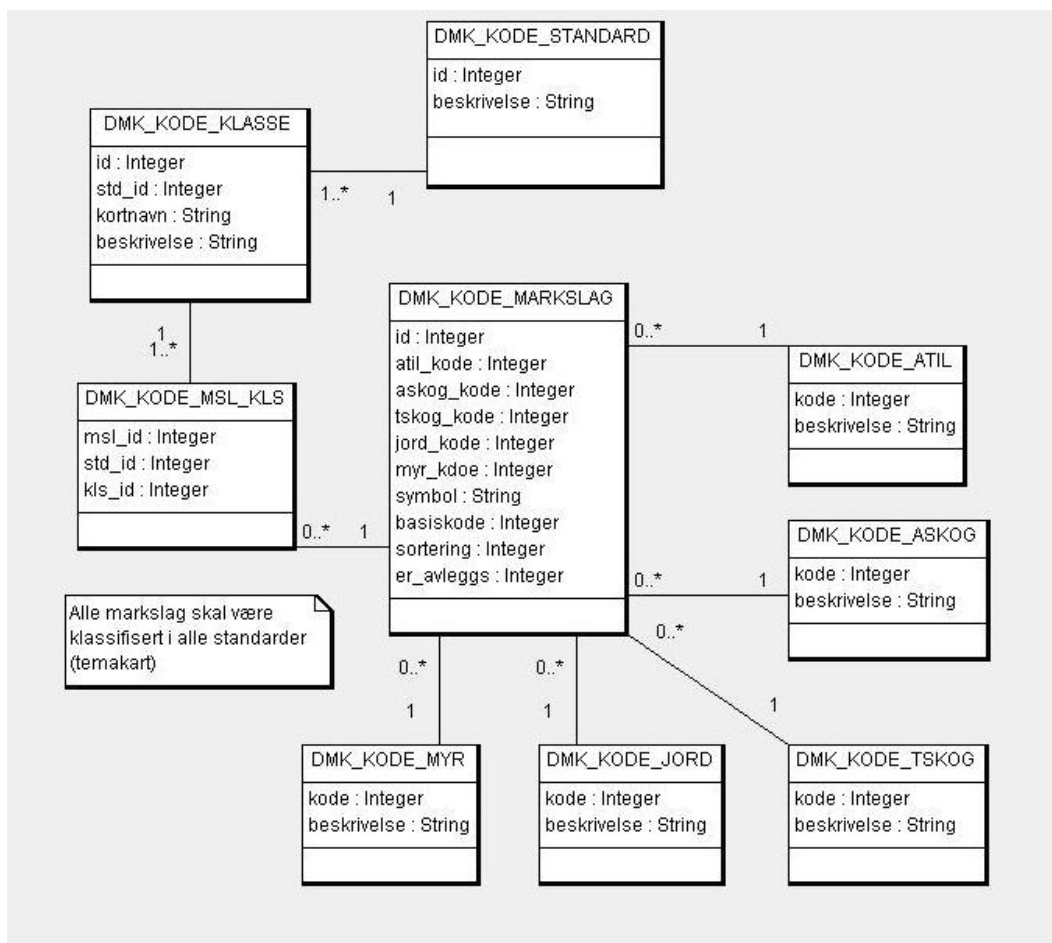
I tabellen DMK\_KODE\_STANDARD i kodeverket for DMK, er 13 standarder definert, med kolonnene id, kortnavn og beskrivelse. I tabellen DMK\_KODE\_KLASSE finnes de ulike klassene hver standard er delt inn i. Følgende klasser er spesifisert for standarden med kortnavn dyrkingsjord:

Klasser temakart dyrkingsjord		
ID	KORTNAVN	FORKLARING
1	0	Ikke dyrkingsjord
2	A	Lettbrukt dyrkingsjord
3	B	Mindre lettbrukt dyrkingsjord

Tabell 10.3: Klasser i temakart dyrkingsjord [7].

På toppen av kodeverket, er det laget et logisk lag med en rekke views, som gjør det enkelt å bruke kodeverket til å omkode til ulike temakart. Et av disse viewene er DMK\_KODE\_DYRKING\_V. Viewet er igjen er brukt for å lage et temakart for dyrkingsjord, implementert som et view i ArcSDE, som heter MARKSLAG\_DYRKING\_FLATE\_V. Se vedlegg 2 for spesifikasjon av viewene.

Omkodingen til dyrkingsjord er definert ut fra egenskapen jord\_kode i DMK som beskrevet i tabell 10.4. Legg merke til at fulldyrka jord ikke er dyrkingsjord.



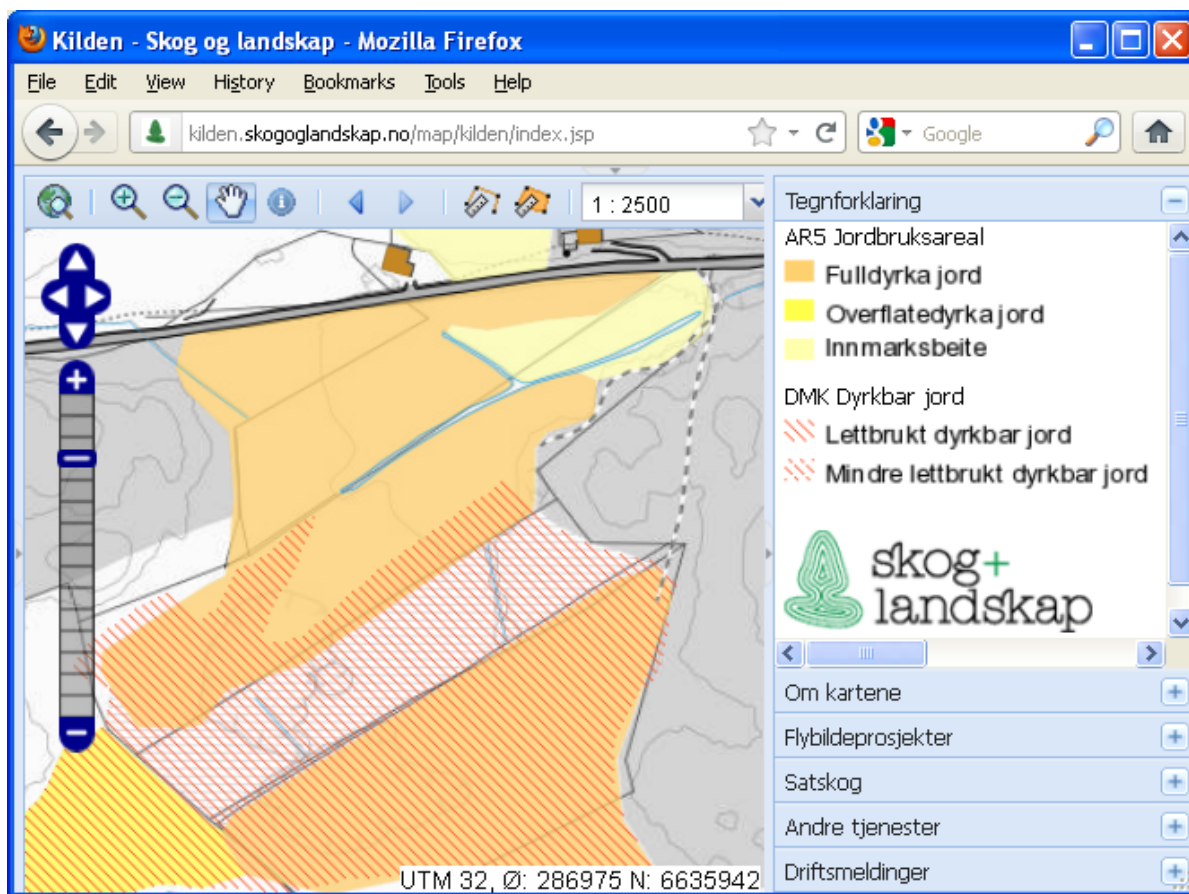
Figur 10.3: Datamodell for kodeverket

Omkoding til tematkart dyrkingsjord		
Kode	Gammel kode	Programmeringsrettledning
1	0	(JORD<31 OR JORD>84) OR ATIL = 21
2	A	JORD BETWEEN(31,54) AND ATIL <> 21
3	B	JORD BETWEEN(61,84) AND ATIL <> 21

Tabell 10.4: Omkoding av tema dyrkingsjord. Utvidelse av tabell i [7].

Dyrkingsjord er altså ikke en eksplisitt egenskap i DMK, men kan utledes fra koden for egenskapen jord. Den utledninga er gjort i kodeverket, slik at den kan gjenbrukes i ulike applikasjoner. Slik reduseres sannsynligheten for feil bruk av den sammensatte jordklassifikasjonen.

Som i definert for Økonomisk kartverk i [21], er det slik at både områder som ikke allerede er jordbruksareal, og områder som er overflatedyrket eller innmarksbeite, kan få betegnelsen dyrkingsjord. Når overflatedyrka jord eller innmarksbeite får betegnelsen dyrkingsjord, betyr det at området kan fulldyrkes til lettbrukt eller mindre



Figur 10.4: Visning av dyrka jord (gulaktig farge) og dyrkbar jord (skravert)

lettbrukt areal.

I figur 10.4 er et eksempel på en visning av jordbruksareal og dyrkingsjord. Legg merke til at noen områder er klassifisert som bege deler.

#### 10.2.4 Datoangivelsenes kvalitet i DMK-databasen

For å kunne lage informasjonssystemet vårt, er vi avhengig av god kvalitet på datoangivelsene i databasen.

DMK er standardisert etter SOSI versjon 3. I den versjonen av SOSI, var det kun en datoegenskap: dato. I prinsippet skulle denne datoen settes til dato for datafangst for nye objekter, og for eksisterende objekt til den datoen objektet sist var sjekket mot den virkelige verden og oppdatert om nødvendig. Som standard dato for ukjent dato, ble først 1.1.1000 brukt, deretter 1.1.1800.

I databasene hos Skog og landskap er dato-egenskapen for datatypen en tekststreng som er formatert slik: «yyyymmdd». Den største verdien egenskapen dato har i databasen er '70020422', som oversatt til en dato blir 22. april 7002. Andre datoer har så lave årstall at det ikke kan være riktig, for eksempel 1194. Det er også objekter der dato er null, og det er objekter som har dato-egenskap som ikke følger riktig format, slik at de ikke kan oversettes til en lovlig dato. Det er grunn til å tro at bruken av

dato-egenskapen ikke alltid har vært like gjennomført.

I tillegg til dato-egenskapen fra SOSI, er det lagt på en egenskap som viser når et objekt ble lagt inn i databasen (transaksjonstid). Dessverre ble det brukt en tekststreng som datatype i stedet for en dato. Denne strengen har blitt formatert litt ulikt opp igjennom årene. Det gjør søk på denne egenskapen vanskelig.

En mer detaljert undersøkelse av kvaliteten på dato-egenskapen gjøres i kapittel 12.

### 10.2.5 Generell kvalitet i DMK-databasen

Som nevnt i kapittel 3, kan kvaliteten på DMK variere fra område til område fordi digitaliseringen foregikk over en lang periode med bruk av ulik teknologi. Imidlertid kan man anta at kvalitetsforskjellen ikke betyr så mye i forhold til klassifiseringen av dyrkingsjord. Den ble gjort i felt, og har uansett en ganske grov nøyaktighet.

Kvalitet angis i DMK ved SOSI-egenskapene målemetode, nøyaktighet og synbarhet. På tidlig 90-tall gikk man over fra desimeter nøyaktighet til centimeter. Et resultat av dette, er at det noen steder i databasen er små gap og overlapp mellom ulike ØK-kartblad (ØK-kartblad er den administrative inndelingen av DMK-databasen i ArcSDE/PostGIS).

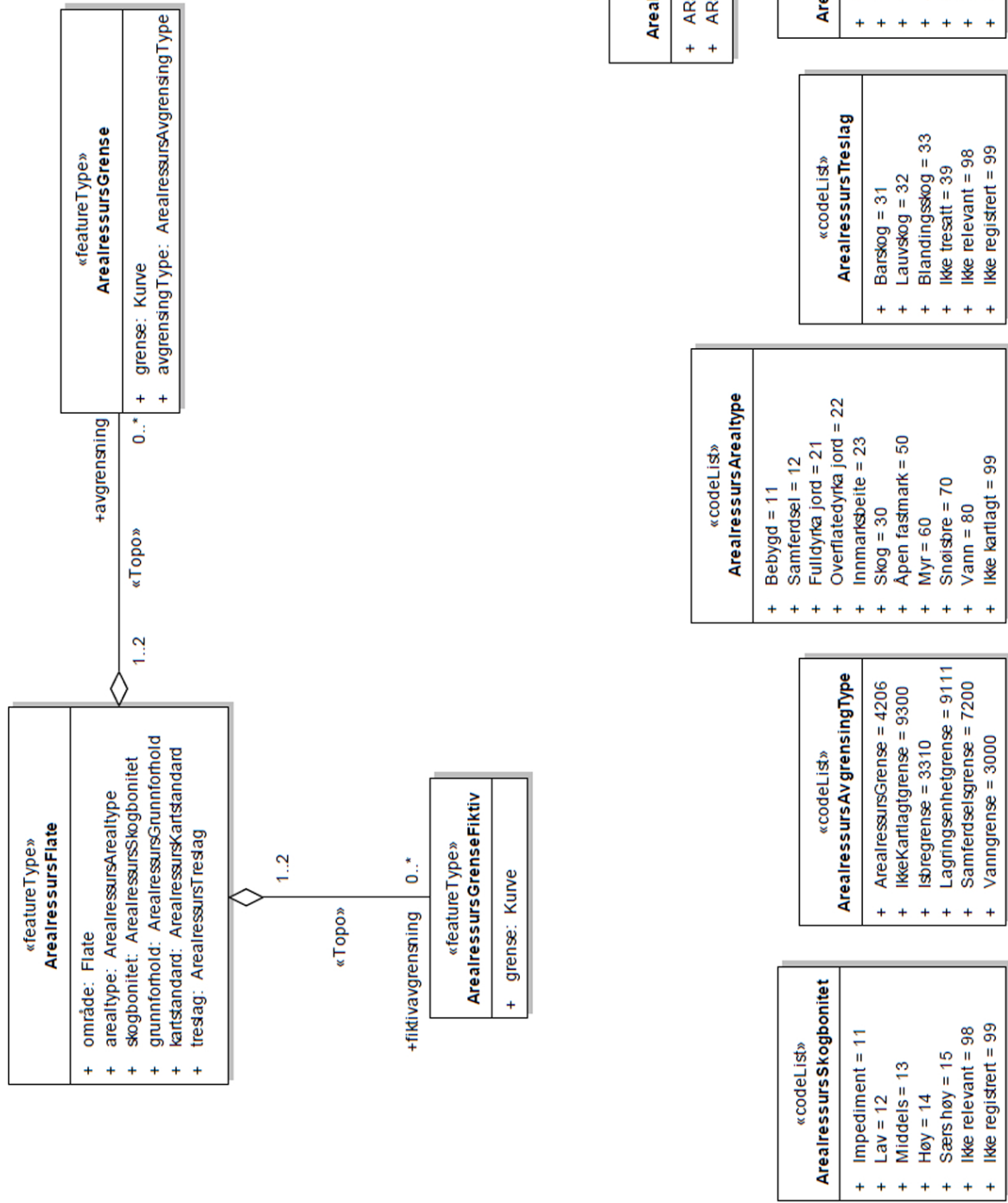
## 10.3 AR5

DMK er et komplisert datasett med mange egenskaper og lovlige kombinasjoner. En del informasjon som ligger i DMK, er å betrakte som mer statisk enn annen informasjon. For å lette ajourføring av datasettet, ble det bestemt at DMK-modellen skulle forenkles betydelig. AR5 ble resultatet av denne forenklingen, og klassifikasjonssystemet ble ferdigstilt i 2006 [5]. AR5 har fire ulike egenskaper som representerer inndelingen i ulike markslag, se tabell 10.5.

Egenskaper som beskriver markslag i AR5	
Egenskapstype	Definisjon
Arealtype	Hovedinndeling etter kriterier for vegetasjon, naturlig drenering og kulturpåvirkning.
Skogbonitet	Inndeling etter arealets evne til å produsere trevirke.
Treslag	Inndeling etter fordeling av kronedekket mellom bartrær og løvtrær.
Grunnforhold	Inndeling etter tykkelse, type og fordeling av jorddekket.

Tabell 10.5: AR5 egenskaper [5].

Antall lovlige kombinasjoner av koder ble redusert fra over 3700 i DMK til rett i overkant av 100 i AR5. SOSI-modellen for markslag etter overgangen til AR5-klassifikasjonssystem vises i figur 10.5. Den er hentet fra produktspesifikasjonen for FKB [48].



Figur 10.5: SOSI UML-modell for AR5 i FKB

## Datamodell og temakart

Første versjon av AR5 ble laget ved en avledning (omkoding) av DMK. AR5 er, i likhet med DMK, vektordata i form av polygoner avgrenset av kurver. AR5 forvaltes av kommunene, men det ligger en nasjonal kopidatabase i QMS hos Statens kartverk, en nasjonal kopibase i ArcSDE hos Skog og landskap og en nasjonal kopidatabase i PostGIS, også hos Skog og landskap. Det er kun polygoner etter simple feature-modellen som lagres i ArcSDE og PostGIS. I tillegg er AR5 tilgjengelig som SOSI-filer. Alle SOSI-filer tas vare på i en filstruktur hos Skog og landskap.

Databasene i QMS og ArcSDE er synkronisert gjennom manuelle rutiner. Det utelukker selvfølgelig ikke at den ene databasen i korte perioder kan være mer oppdatert enn den andre. PostGIS-basen synkroniseres automatisk med ArcSDE-basen en gang i uka. AR5 er et svært dynamisk datasett som gjennomgår både periodiske og kontinuerlige ajourføringer. Den administrative inndelingen av AR5-basen i ArcSDE ved oppdatering, er Euref-ruter. En hel rute slettes, før den legges inn på nytt. En tar dermed ikke vare på historikken i databasen. Jordbruksareal er i AR5 fordelt på klassene fulldyrka jord, overflatedyrka jord og innmarksbeite. Dyrkingsjord er som sagt ikke tatt med over i AR5, så den informasjonen må man fortsatt hente i DMK.

Akkurat som for DMK, er datamodellen for AR5 i Skog og landskap sine databaser svært enkel. Den ivaretar ikke topologi, og inneholder kun polygoner. AR5 har også et mye enklere kodeverk enn DMK. Det er ingen skranker mellom layer og kodeverk. I QMS er det en mer avansert datamodell basert på SOSI-modellen, som også ivaretar topologi.

Temakart er implementert som views i både ArcSDE og PostGIS. I QMS finnes ikke temakart. Omkodingen til temakart i AR5 er definert i en enkel tabell som inneholder alle lovlige kombinasjoner av de fire kodene i tabell 10.5 og hvilken klasse hver enkelt kombinasjon hører hjemme i for hvert av temakartene.

### Bruk av AR5 i TilgjengeligJord

I datasettet for tilgjengelig jord, er det egenskapen «Arealtype» (ARTYPE i SOSI-filer og databaser) og «Grunnforhold» (ARGRUNNF i SOSI-filer og databaser) som skal brukes. Arealtype vil brukes for å velge ut nye jordbruksarealer, og for å se om områder allerede i TilgjengeligJord har endret status. Det kan være dyrkingsjord som ikke er dyrkingsjord lenger fordi den har blitt nedbygd eller dekket av vann eller snøisbre, eller tidligere jordbruksarealer som ikke er det lenger.

Grunnforhold har koder for om et område er myr eller ikke. Den er interessant for å kunne identifisere nydyrking på myr og oppdyrka myr. Se tabell 10.6 for verdier for egenskapstypen «Arealtype», og tabell 10.7 for verdier for egenskapstypen «Grunnforhold». Egenskapen vil brukes til å oppdatere myr-statusen til arealer i TilgjengeligJord.

### 10.3.1 Ajourføring av AR5

Ajourføringen gjøres kontinuerlig av kommunene og periodisk av Skog og landskap.

Kommunenens ajourføring vil i stor grad være rettet mot løpende forvaltningsoppgaver i kommunen, som nedbygging og nydyrking. Endringer som berører jordbruksareal skal ha høyeste prioritet [60]. Kommunene skal levere ajourholdt AR5 i henhold til



Arealtype		
Egenskapsverdi	Kodeverdi	Definisjon
Fylldyrka jord	21	Jordbruksareal som er dyrk til vamlig pløyedyp, og kan benyttes til åkervekster eller eng, og som kan fornyes ved pløying
Overflatedyrka jord	22	Jordbruksareal som for det meste er rydda og jevna i overflata, slik at maskinell hesting er mulig.
Innmarksbeite	23	Jordbruksareal som kan benyttes som beite, men som ikke kan hastes maskinelt. Minst 50 % av arealet skal vær edekt av grasarter eller beitetålende urter.
Skog	30	Areal med minst 6 trær per dekar som er eller kan bli 5 meter høye, og disse bør være jevnt fordelt over arealet.
Myr	60	Areal med myrvegetasjon og minst 30 cm tjukt torvlag
Åpen fastmark	50	Fastmark som ikke er jordbruksareal, skog, bebygd eller samferdsel.
Vann	80	Uspesifisert vannflate.
Snøisbre	70	Blanding av snø og isbre som ikke smelter i løpet av sommeren.
Bebygd	11	Areal som er utbygd eller i betydelig grad opparbeidet, samt tilstøtede arealer som i funksjon er nært knyttet til bebyggelsen.
Samferdsel	12	Areal som brukes til samferdsel.
Ikke kartlagt	99	Areal som har ukjent beskaffenhet.

Tabell 10.6: Verdier for egenskapstypen Arealtype i AR5. Hentet fra [5].

Grunnforhold		
Egenskapsverdi	Kodeverdi	Definisjon
Organiske jordlag	45	Areal som har et organisk jordlag tjukkere enn 30 cm (20 cm).
Jorddekt	44	Fastmark der mer enn 50 % av arealet har større jorddybde enn 30 cm.
Grunnlendt	43	Areal der mer enn 50 % har mindre jorddybde enn 30 cm, men som ikke kan klassifiseres som fjell i dagen.
Fjell i dagen	42	Areal der mer enn 50 % er bart fjell og mindre enn 10 % har jord dypere enn 30 cm.
Blokkmark	41	Areal der overflata i hovedsak er dekt med steinblokker.
Ikke relevant	98	Opplysning om grunnforhold er ikke relevant.
Ikke registrert	99	Opplysning om grunnforhold er ikke registrert..

Tabell 10.7: Verdier for egenskapstypen Grunnforhold i AR5. Hentet fra [5].

forvaltningsavtalen i Geovekst, det er ikke nedfelt i lov og forskrift at AR5 skal ajourholdes. Det er Skog og landskaps erfaring at arbeidet ofte nedprioriteres, og at leverte data ikke er fullstendig ajourholdt. Mange kommuner leverer ikke til avtalt tid.

Det periodiske ajourholdet av AR5 skal sikre et homogent nasjonalt datasett. Når Skog og landskap gjør en periodisk ajourføring, går en gjennom alt jordbruksareal og oppdaterer grenser og egenskaper. Man ser etter nedbygde arealer, gjengroddde arealer. Informasjon om samferdsel og vann byttes ut med siste versjon av FKB-data. Mange endringer i ajourholdet er marginale, og kan like mye ha sitt opphav i et ortofoto der for eksempel skyggene er annerledes enn på ortofoto brukt ved tidligere ajourføring. I noen tilfeller retter man også opp endringer gjort av kommunene i kontinuerlig ajourhold fordi man ser at det er gjort feil eller unøyaktigheter [60].

### 10.3.2 Datoangivelsenes kvalitet i AR5-databasen

AR5 er standardisert i SOSI versjon 4. Det er to typer påkrevde datoer i datasettet: verifiseringsdato og datafangstdato. Begge er standardisert i SOSI. Verifiseringsdato er dato for når dataene er fastslått å være i samsvar med virkeligheten. Verifiseringsdato

kan benyttes når en har fastslått at objektet som finnes i basen fortsatt er i samsvar med virkeligheten [63]. Datafangstdato er dato for når når objektet siste gang ble registrert/observert/målt i terrenget [63].

I en presentasjon om metadata for AR5 tilgjengelig på Skog og landskaps nettsider [61], er følgende veiledning for bruk av verifiseringsdato og datafangstdato gitt:

**«Ved registrering og klassifisering i felt:**

- Datafangstdato den dagen arbeidet utføres i felt. Dette blir også verifiseringsdato.

**Ved registrering og tolking i ortofoto inne:**

- Datafangstdato den dagen flybildene ble tatt. (Ikke datoen da arbeidet i kartverktøyet ble utført.)
- Samme dato benyttes som verifiseringsdato.

**Kontrollert på feltbefaring:**

- Hvis flate eller grense i kartet fortsatt stemmer med virkeligheten, settes feltdagens dato som verifiseringsdato.
- Flaten eller grensen beholder opprinnelig datafangstdato, fordi det ikke er foretatt noen ny datafangst.»

De to datoangivelsene, er ikke ortogonale. Verifiseringsdato vil ikke, ut fra dagens definisjon, kunne brukes til å finne ut når noe var registrert som en sannhet i databasen. En samtale med Vidar Svendsen i Arealressurs-seksjonen ved Skog og landskap bekreftet også dette [65]. Verifiseringsdato blir ikke brukt på en slik måte at den vil kunne identifisere feilretting. Når et AR5-polygon endrer egenskaper på grunnlag av ny datafangst, vil man kalle det ajourføring, og begge datoer skal endres. Dersom et AR5-polygon endrer egenskaper uten at det har foregått ny datafangst (for eksempel fordi det er åpenbart at det er gjort en feiltolkning av et flybilde), vil egenskapene rettes, men ingen av datoene vil bli oppdatert.

Verifiseringsdato er allikevel nesten det samme som «assertion time» omtalt i avsnitt 6.2. En liten endring i rutine ved feilretting er det som skal til. Datafangstdato har sterke likhetstrekk med gyldig tid. Datafangstdato angir riktig nok ikke tidspunktet for når noe var sant i den virkelige verden (selv om den i enkelte tilfeller kan gjøre det). Når datafangsten gjøres ved hjelp av flybilder, vil man være litt på etterskudd.

I tillegg til verifiseringsdato og datafangstdato, legges det på en kjøringdato som identifiserer når et polygon ble lagt inn i databasen (transaksjonstid). Da AR5 ble etablert ved en omkodning av DMK, ble verifiseringsdato satt til dato-egenskapen i DMK. Dermed ble en del av de åpenbart gale datoene fra DMK overført til AR5. Fordi planlagte tiltak ikke skal legges inn i AR5 før de er gjennomført, skal en dato i AR5 aldri være senere enn den til enhver tid gjeldende dagens dato. Det ligger allikevel en god del slike datoer i databasen. En annen tallrik feil, er at datafangstdatoen er nyere enn verifiseringsdatoen. Det skal ikke forekomme. I kapittel 12 er det mer om kvaliteten til datoangivelsene.

### 10.3.3 Generell kvalitet i AR5-databasen

Som for DMK, er det SOSI-egenskapene målemetode, nøyaktighet og synbarhet som angir generell kvalitet. For AR5 er det et generelt krav til stedfestingsnøyaktighet for markslagsgrenser på +/- 2 meter. Nøyaktighet registrert på hver linje, er generell stedfestingsnøyaktighet for brukt målemetode. Nøyaktighet angis ikke på flater. Synbarhet brukes litt ulikt på grenser og flater. For grenser er det registreringssikkerheten som beskrives, på flater er det sikkerheten av marslagsklassifiseringen som beskrives. Fordi både avgrensninglinjer og klassifisering kan være usikre, må man passe på å beskrive begge [48, 61].

AR5 har generelt bedre kvalitet enn DMK.

## 10.4 Årsversjoner av AR5

I 2008 bestemte Skog og landskap at de ønsket å ta vare på årsversjoner av AR5. Det tas et øyeblikksbilde av AR5 så nær årsskiftet som mulig ved at alle data i AR5-layeret kopieres ut i et annet layer, som får navnet AR5\_<ÅRSTALL>\_KOMM\_FLATE. Først gjøres et overlay med gjeldende kommunekart, slik at kommunenummer blir en egenskap for alle polygoner. Årsversjonen gjennomgår deretter en utstrakt kvalitetskontroll av både topologi, geometri og egenskaper før det fryses som et statisk datasett.

Dette kvalitetskontrollerte datasettet brukes deretter i produksjon av generaliserte datasett som AR50 og AR250, og til statistikkprodukter og ad-hoc leveranser av kart og data der det er viktigere å ha kontroll på hvilke data som faktisk er brukt enn at det er de aller nyeste data som inngår.

Per i dag er årsversjonen fra 2008 og 2009 lagret i Oracle/ArcSDE, mens årsversjonen fra 2010 er lagret i PostGIS. Med unntak av kommunenummer, er datamodellen for årsversjonene lik datamodellen for AR5.

## 10.5 Endringsanalyse

Når en kommunes AR5 er ajourført, gjennomføres det som kalles en endringsanalyse. Det er gjort fra og med våren 2011. Gammel AR5 for kommunen sammenliknes med ny AR5 ved at det gjøres et overlay (spatial join) mellom ny og gammel SOSI-fil, og ulikhetene mellom de to versjonene registreres og tas vare på i egne endrings-layerer i Oracle/ArcSDE hos Skog og landskap, med en kopi i PostGIS. Endringsdataene visualiseres som kart og statistikk, og det er laget Internett-tjenester over endringslayerne som brukes i dialogen mellom Skog og landskap og kommunene. Endringsanalysen gir forskjellene mellom to SOSI-filer for samme område, ikke arealendringer i landskapet. Noen av forskjellene mellom to filer er selvfølgelig også arealendringer, men endringsanalysen forteller ikke hvilke endringer det gjelder. Figur 10.6 viser bilde fra webapplikasjon for inspeksjon av endringer, og figur 10.7 viser alle endringene på kart.

**Kommune:** 1219  
**Endring utført av:** Kommunen  
**Fra:** 25.01.2010  
**Til:** 30.06.2011  
**Sum endringer:** 6 104,9 dekar



#### Arealendringer fra jordbruksareal til annet areal

Fra	Til	daa
Fulldyrka/Overflatedyrka	Bebygd/Samferdsel	2,2
Fulldyrka/Overflatedyrka	Skog/Åpen fastmark	31,4
Innmarksbeite	Bebygd/Samferdsel	9,8
Innmarksbeite	Skog/Åpen fastmark	146,8
Fulldyrka/Overflatedyrka/Innmarksbeite	Myr/Snøisbre/Vann	35,6
<b>Sum</b>		<b>225,8</b>

#### Arealendring fra annet areal til jordbruksareal

Fra	Til	daa
Skog	Fulldyrka/Overflatedyrka	23,9
Skog	Innmarksbeite	476,2
Åpen fastmark	Fulldyrka/Overflatedyrka	47,6
Åpen fastmark	Innmarksbeite	924,3
Bebygd/Samferdsel	Fulldyrka/Overflatedyrka/Innmarksbeite	7,9
Myr/Snøisbre/Vann/Ikke kartlagt	Fulldyrka/Overflatedyrka/Innmarksbeite	55,3
<b>Sum</b>		<b>1 535,2</b>

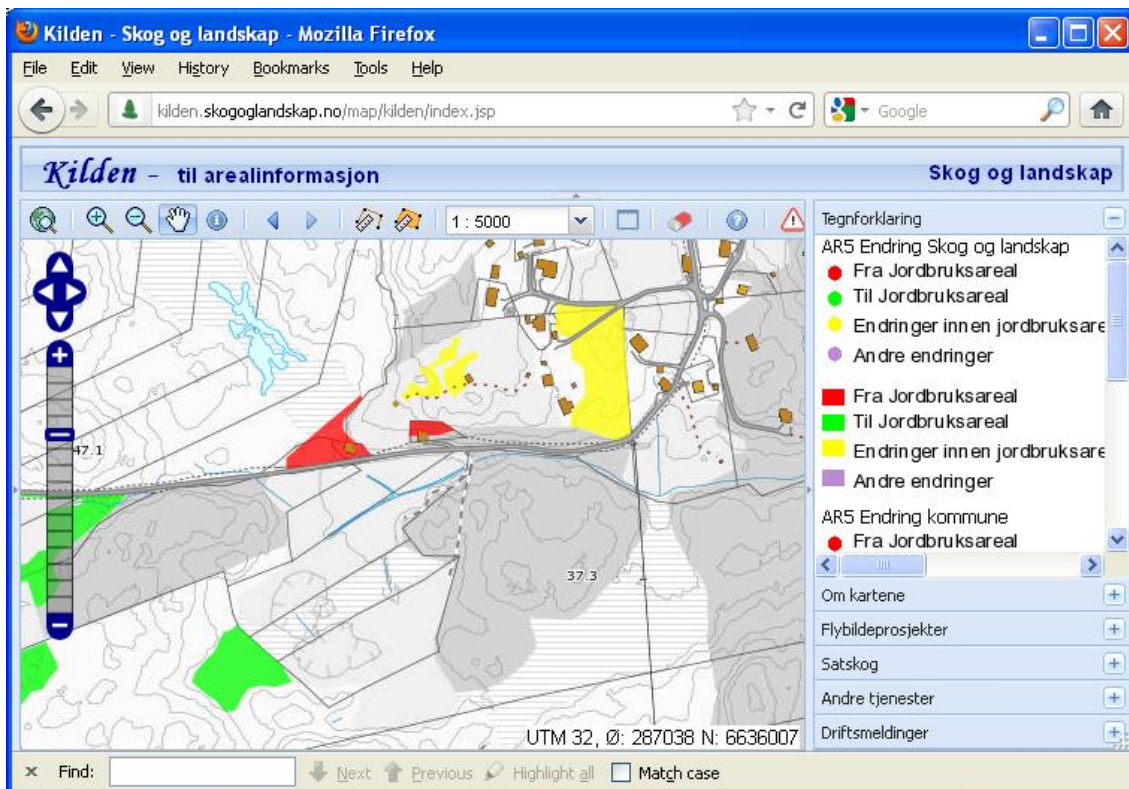
#### Endringer innen jordbruksareal

Fra	Til	daa
Fulldyrka	Overflatedyrka/Innmarksbeite	102,1
Overflatedyrka	Fulldyrka/Innmarksbeite	340,3
Innmarksbeite	Fulldyrka/Overflatedyrka	339,9
<b>Sum</b>		<b>782,3</b>

#### Andre endringer

Fra	Til	daa
Skog	Bebygd/Samferdsel	2,4
Skog	Åpen fastmark/Myr/Snøisbre/Vann	60,7
Åpen fastmark	Bebygd/Samferdsel	7,9
Åpen fastmark	Skog/Myr/Snøisbre/Vann	193,3
Bebygd/Samferdsel	Skog/Åpen fastmark/Myr/Snøisbre/Vann	0,1
Bebygd	Samferdsel	0,0
Samferdsel	Bebygd	0,0
Myr	Bebygd/Samferdsel/Skog/Åpen fastmark/Snøisbre/Vann	2,1
Snøisbre	Bebygd/Samferdsel/Skog/Åpen fastmark/Myr/Vann	0,0
Vann	Bebygd/Samferdsel/Skog/Åpen fastmark/Myr/Snøisbre	0,2
Ikke kartlagt	Alle areal typer	2 952,1
Alle areal typer	Ikke kartlagt	342,8
<b>Sum</b>		<b>3 561,6</b>

Figur 10.6: Webapplikasjon for innsyn i detaljerte endringer



Figur 10.7: Kart som viser endringer mellom ulike datasett for et område

Datasettet som inneholder endringsanalysen, ligger i et layer som heter AR5\_ENDRING\_FLATE i ArcSDE. Det ser slik ut:

AR5\_ENDRING\_FLATE:

Navn på kolonne	Null?	Type
OBJTYPE		VARCHAR2(40)      Objekttype
ARTYPE_NY		NUMBER(2)        Arealtype i det nye datasettet
ARTRESLAG_NY		NUMBER(2)        Treslag i det nye datasettet
ARSKOGBON_NY		NUMBER(2)        Skogbonitet i det nye datasettet
ARGRUNNF_NY		NUMBER(2)        Grunnforhold i det nye datasettet
ARTYPE_GML		NUMBER(2)        Arealtype i det gamle datasettet
ARTRESLAG_GML		NUMBER(2)        Treslag i det gamle datasettet
ARSKOGBON_GML		NUMBER(2)        Skogbonitet i det gamle datasettet
ARGRUNNF_GML		NUMBER(2)        Grunnforhold i det gamle datasettet
AREAL		NUMBER(18,5)     Polygonets areal i m2
KOMM		NUMBER(4)        Kommunenummer
TYPE		VARCHAR2(8)      Type ajourføring (periodisk (før og etter)/ kontinuerlig)
OPPHAV		VARCHAR2(50)     Opphav til endringen
DATASETT1		VARCHAR2(8)      Dato på gammel SOSI-fil
DATASETT2		VARCHAR2(8)      Dato på ny SOSI-fil
KJORINGSIDENT		DATE              Den datoen endringsanalysen er kjørt. Normalt det samme som transaksjons- datoen for innlegging i basen.
SL_SDEID	NOT NULL	NUMBER(9)        Unik ID for polygon i SDE
GEO		NUMBER(38)       Fremmednøkkel til geometrikolonne

I layeret AR5\_ENDRING\_FLATE, er det to datoer, begge implementert som tekststrenger. Det er datasett1, som er generert ut fra filnavnet til SOSI-fila for det gamle datasettet som inngikk i analysen, og datasett2, som er generert ut fra filnavnet til SOSI-fila for det nye datasettet. Filnavnet til SOSI-filene genereres etter når den er tatt ut fra QMS-serveren til Kartverket.

Det er ingen ulovlige datoer i AR5\_ENDRING\_FLATE i dag, til tross for at datasett1 og datasett2 er implementert som tekststrenger, og dermed tillater ulovlige datoer. Fordi de representerer dato for når en kommunes AR5 ble tatt ut av QMS og sendt Skog og landskap, og verken tilsvarende datafangstdato eller verifiseringsdato, er det imidlertid vanskelig å benytte endringsanalysen i et informasjonssystem for TilgjengeligJord. Det eneste vi med sikkerhet vet om en endring i endringsanalyse-layeret med gitte datasett1 og datasett2-datoer, er at endringen må ha skjedd før datasett2-datoen.

Endringsanalysen er en godt verktøy for å kommunisere med kommunene om ajourføring av AR5, men er i nåværende form ikke egnet til å inngå i informasjonssystemet for tilgjengelig jord. Dersom datafangstdato og verifiseringsdato for objekt i nytt og gammelt datasett hadde inngått, ville endringsanalysen vært et nyttig bidrag til informasjonssystemet. Endringsanalysen ville da kunne fanget opp endringer mellom årsversjoner av AR5 (i den grad hendelser som gjenspeiles i AR5 skjer så raskt), og man ville kunne inkludere nye data før de kom med i neste årsversjon. I tillegg ville det være et godt verktøy for å fange opp manglende bruk av datafangstdato og verifiseringsdato ved ajourføring.

Å kjøre endringsanalyse er en ny rutine som av og til uteglemmes. Noen få kommuner ajourfører AR5 direkte i QMS-basen hos Statens kartverk. Disse må huske å

gi oss beskjed når de er ferdig, slik at vi kan kjøre endringsanalyse. Det er behov for bedre rutiner både hos Skog og landskap og kommunene.

## 10.6 Testdata for masteroppgaven

I forbindelse med denne masteroppgaven er DMK, AR5, årsversjoner fra 2008 og 2009 av AR5 og endringsdata for AR5 for Hobøl kommune (kommunennummer 0138) kopiert over til en egen database i PostGIS. Hobøl er valgt fordi kommunen er periodisk ajourført to ganger, og det er gjort kontinuerlig ajourføring. Det er kun polygoner som brukes, ikke avgrensingslinjer. Med unntak av DMK, som ble normalisert i uttaket (se vedlegg 1 for programkode og spørringer), har de beholdt navn, egenskaper og datamodell fra Oracle/ArcSDE-basen. Uttrekket av DMK har også fått omkodet arealtilstand-egenskapen til tilsvarende arealtype i AR5.

I tillegg er årsversjon av AR5 fra 2010 hentet fra Skog og landskaps PostGIS-database. Også her er originalstrukturen beholdt. Alle dataene er tatt ut i shape-filer for overføring til arbeidsdatabasen. Shape-formatet har en begrensning på 10 tegn i kolonnenavn, derfor har en del kolonnenavn blitt avkortet. Alle data er i Euref89 UTM 32 (EPSG-kode 25832). De beskrevne dataene vil bli brukt i de praktiske eksperimentene i oppgaven. Se vedlegg 1 for kode brukt for å etablere testdata.

Jeg innså fort at det ble tidkrevende å jobbe med hele kommunen. Jeg klippet derfor ut ca 9 km<sup>2</sup> av dataene, og la inn i egne layer. Layerne med utklippede dataene, heter det samme som layerne med data for hele kommunen, men med «\_test» bakerst i navnet.

Layer i PostGIS med data for deler av Hobøl kommune	
LAYERNAVN	BESKRIVELSE
tilgjengeligjord_0138_test	Omkodet uttrekk av DMK for Hobøl. Det er jordegenskapen som er omkodet for lettere bruk. Dataene fikk en tur innom ArcGIS-formatet cover for å dekke enkelte bittesmå gap mellom kartblad.
ar5_2008_0138_test	Årsversjon 2008 for deler av Hobøl kommune. Samme datamodell som AR5
ar5_2009_0138_test	Årsversjon 2009 for deler av Hobøl kommune. Samme datamodell som AR5
ar5_2010_0138_test	Årsversjon 2010 for deler av Hobøl kommune. Samme datamodell som AR5

Tabell 10.8: Layerne i PostGIS med uttrekk av eksisterende data for Hobøl kommune

Testdata satt inn i Sintons rammeverk			
Datasett	Tid	Attributt	Sted
DMK	Låst	Kontrollert	Målt
AR5	Låst	Kontrollert	Målt
Endringsanalyse	Målt	Kontrollert	Målt
AR5 - sekvensielle årsversjoner	Kontrollert/Låst	Kontrollert	Målt

Tabell 10.9: Tilgjengelige datasett satt inn i Sintons rammeverk

### 10.6.1 Testdata satt inn i Sintons modell

Tabell 10.9 gir en oversikt over hvordan dataene som brukes i oppgaven ville plassert seg i Sinton sitt rammeverk. Vi ser at det kun er endringsanalysen, som ikke kommer til å bli brukt i TilgjengeligJord, som ikke låser tiden. For samlingen av sekvensielle årsversjoner kan det argumenteres for at tiden er kontrollert, men for hver enkelt årsversjon er den låst.



# 11 Klassifikasjon av endringer i geodata

Oppdateringer av et datasett kan ha ulike årsaker. For denne oppgaven, er det fire typer endringer som er spesielt interessante. Først og fremst er det oppdateringer som skyldes endringer i den virkelige verden. Disse omtales videre som reelle endringer. Det vil også kunne skje endringer på grunn av ny og bedre teknologi, eller nye metoder. Til slutt er det endringer som skyldes retting av feil. Det er viktig å holde så god orden som mulig på hvilke endringer som skyldes hva.

Formålet med informasjonssystemet vårt, er å kunne si noe om hvor mye tilgjengelig jord vi har i Norge, og hva som skjer med den. Ikke minst er det interessant å vite hvor mye tilgjengelig jord som blir nedbygd eller går ut av produksjon av andre årsaker. I den sammenhengen er det de reelle endringene som er viktige. Tas det med endringer som skyldes endret metode, ny teknologi eller feilretting, svarer man på et annet spørsmål enn det som er stilt.

Datasettet for tilgjengelig jord skal blant annet kunne brukes til å rapportere endringer for å sammenlikne med Kostra-tall. Da må kun reelle endringer brukes i sammenlikningen. Datasettet vil også brukes til arealstatistikk og presentasjon på kart. Siden endringer av alle typer forhåpentligvis bidrar til et bedre datasett, kan alle endringer inngå i statistikken og øyeblikksbildene som brukes til presentasjon av dataene på kart. Allikevel bør man vite, og dokumentere, hvor mye av endringene fra forrige versjon av statistikken som skyldes reelle endringer, og hvor mye som har andre årsaker.

## 11.1 Klassifikasjon av endringer

### Reelle endringer

Når det skjer endringer i landskapet, skal det reflekteres i dataene. Det kan for eksempel være et jorde som bygges ned eller skog som hugges før arealene dyrkes opp.

De reelle endringene er interessante for å følge med på utviklingen av jordbruksarealer og dyrkingsjord i Norge.

### Endringer som skyldes forandret metodikk

Endringer i klassifikasjonssystemet til datasettet, vil igjen føre til endringer i datasettet. For AR5 kan det for eksempel være endringer i klassene for arealtyper, eller endring i minstearealer som skal registreres.

Dersom kravene til barskog endrer seg fra å være skogareal der minst 50 % av arealet er dekket av bartrær til skogareal der minst 60 % av arealet er dekket av bartrær, får

vi mindre barskog enn før. Men det betyr ikke at det er mindre barskog i Norge, kun at registreringen av barskogen er forandret.

Skog og landskap ønsker å splitte arealtypen «åpen fastmark» i AR5, slik at det blir mulig å skille menneskeskapt åpen fastmark (kulturpåvirkede arealer) fra naturlige arealer. Dersom dette blir gjennomført, vil all åpen fastmark etter hvert bli omkodet. Slike endringer må kodes som metodiske endringer, slik at de ikke rapporteres som reelle endringer.

Endres minsteareal for hvor store arealer med «ikke jordbruksareal» man skiller ut på fulldyrka jord fra 0,2 dekar til 0,5 dekar, vil vi få mer fulldyrka jord etter hvert som endringen gjør seg gjeldende. Men det skal ikke rapporteres som nydyrking.

Det kan være komplisert å identifisere endringer som skyldes metode fra reelle endringer. I eksemplet med barskog må det identifiseres av den som tolker hver gang barskog går over til blandingsskog hvorvidt endringen skyldes metodeendring eller reell endring. I tilfellet med åpen fastmark er det lettere, da må all omkodning av eksisterende åpen fastmark merkes med metodeendring.

Sett i et tidsperspektiv, er det god grunn til å være svært forsiktig med endringer i metode. De må være gjennomtenkte, godt begrunnet og dokumentert før de settes i verk.

### **Endringer som skyldes ny teknologi**

Som for metodeendringer, vil endringer i teknologi ofte føre til endringer i et datasett. Teknologien er under utvikling, og vi får stadig nye og bedre løsninger. Blant annet blir oppløsningen på ortofoto som brukes i tolkning i AR5, bedre og bedre. Dermed blir det lettere å se hvor jordekanten faktisk går. Skog og landskap er i startfasen av et prosjekt som skal bruke automatisert flybildetolkning for å detektere skoggrensene. Automatikken vil garantert gjøre jobben litt annerledes enn et menneske. Laserdata er i ferd med å innta kart-Norge, og vil helt sikkert inngå i datafangsten i AR5 om noen år.

Overgangene mellom ulike areal typer i den virkelige verden kan være gradvise, og er ofte vanskelige å trekke med en linje. I dag er det allikevel vanlig å implementere gradvise overganger absolutt som en skarp avgrensning. Det gjøres også i DMK og AR5. Dersom man etterhvert får teknologi som håndterer gradvise overganger gjennom flytende mengder, vil det føre til endringer i datasettet som må håndteres som endringer både på grunn av ny teknologi og ny metode.

Akkurat som for metodeendringer, er det både komplisert og viktig å skille endringer som skyldes bruk av ny teknologi fra de reelle endringene så godt som mulig. Det er nødvendig å ta i bruk ny teknologi for å gjøre en effektiv og best mulig kartlegging. Det er viktig å reflektere over hva den nye teknologien gjør med datasettet, og hvordan klassifikasjon av endringer kan ivaretas på best mulig måte når ny teknologi tas i bruk.

### **Retting av feil**

Noen ganger gjør man feil. Det kan skyldes dårlig kompetanse, mye skygge på ortofoto, at tolker ikke har forstått klassifikasjonssystemet eller forglemmelser. Når feil oppdages, ønsker man å rette dem. Endringer i et datasett på grunn av feilretting representerer ikke endringer i selve landskapet, og må ikke rapporteres som reelle endringer. Bitemporale databaser gjør det mulig å rette endringer uten å miste informasjon om

hva man trodde var riktige data før feilrettingen. Feilrettingen vil gjenspeiles i dataene.

## 11.2 Hva gjør andre?

Nederlandske Alterra forvalter arealbruks-informasjon i en multi-dato database. I en artikkel om databasen beskrives hvordan ulike typer endringer håndteres [17]. Databasen er en rasterdatabase med øyeblikksbilder som genereres automatisk fra en rekke datakilder. Endringer som skyldes metode og teknologi er slått sammen, mens feilretting ikke er tatt hensyn til. Reelle endringer identifiseres ved at de lager to datasett når det har skjedd metode- eller teknologiendringer. Ett basert på ny metodikk, og ett basert på gammel. Deretter sammenliknes endringene mellom begge disse datasettene og det gamle datasettet. Det gir et mål på hvor stor andel av endringene som skyldes ny metodikk.

# 12 Problemstillinger rundt håndtering av historikk i AR5 og DMK

TilgjengeligJord-datasettet har to hovedformål. For det første, skal det gi informasjon om hvor det er jordbruksareal og dyrkingsjord i Norge, og hvor mye det er av hver i ulike områder. For det andre, ønsker vi å følge med på hva som skjer med jordbruksareal og dyrkingsjord; hva som har skjedd der jordbruksareal eller dyrkingsjord blir borte, og hvor det nydyrkes. For begge disse formålene, trenger vi et oppdatert datasett som ivaretar historikk.

## 12.1 Kontinuerlig og periodisk ajourføring

For å oppdatere TilgjengeligJord, ønsker Skog og landskap i første omgang å benytte AR5, som ajourføres kontinuerlig av kommunene og periodisk av Skog og landskap. Det er en forutsetning at ajourføringsregimet for AR5 faktisk fungerer. Det betyr at de aller fleste kommuner må levere oppdatert AR5 hvert år, og at de ajourførte kartene virkelig er oppdatert med nedbygde arealer og nydyrking. Både nedbygging og nydyrking er gjenstand for saksbehandling i kommunene, så det burde være mulig for kommunene å fange opp dette.

Andre mer gradvise endringer, vil oftest ikke kommunene få med seg. De fanges opp i det periodiske ajourholdet. Målet er at det periodiske ajourholdet skal gjøres omtrent hvert sjuende år, men dette er ikke helt klarlagt ennå. I det periodiske ajourholdet retter Skog og landskap opp feil og mangler i det kontinuerlige ajourholdet. Det er jordbruksareal som står i fokus, men nedbygde arealer vil tas med også når de er utenfor jordbruksarealet. Det er vesentlig for å kunne bruke AR5 til å oppdaterere informasjon om dyrkingsjord i skog og på myr som blir bygd ned.

## 12.2 Identifisere ulike typer endringer

For å kunne si noe presist om utviklingen av jordbruksarealer og dyrkingsjord over tid, må vi klare å skille reelle endringer fra andre typer endringer. Det ivaretas ikke i AR5 i dag. Dermed vil mer endringer enn det som faktisk skjer i virkeligheten bli rapportert. Identifisering av ulike typer endringer må, der det er mulig, på plass så raskt som mulig for at TilgjengeligJord skal kunne brukes som ønsket.

Ikke alle typer endringer er like enkle å klassifisere. Spesielt en del endringer som skyldes ny teknologi eller bedret kompetanse hos tolkerne, vil føre til mange små



## Målestokk 1:5

Figur 12.1: Små skalke-polygoner (slivers)

endringer i grensene mellom ulike arealtyper når det foretas periodisk ajourføring. Dette kan være feilrettinger eller rettinger på grunn av ny og bedre teknologi, men det kan også være reelle endringer (for eksempel at grensa mellom et fulldyrka areal og skogen ved siden av har flyttet seg litt siden sist) Skalkepolygoner (såkalte «slivers», se figur 12.1) kan fjernes fra rapporteringa ved å ikke ta med små arealer og arealer som har stor omkrets i forhold til arealet. Da risikerer man å fjerne noe areal som representerer reelle endringer. Det beste ville være om endringene ble klassifisert.

## 12.3 Mer korrekt og konsistent bruk av datafangstdato og verifiseringsdato

Det hadde vært en fordel for det nye informasjonssystemet vårt med bedre oppfølging av bruken av egenskapene datafangstdato og verifiseringsdato i AR5. En gjennomgang av den nasjonale AR5-databasen og årsversjonene, viser en rekke feil og mangler, i tillegg til at instruksene for bruk av datoer kunne vært mer gjennomtenkt, spesielt for feilretting.

I AR5-databasen finnes en rekke gale datoer. Man skal ikke kartlegge endringer som ikke har skjedd i den virkelige verden ennå. Ergo skal ingen datoer være fram i tid. Reglene for bruk av verifiseringsdato og datafangstdato, medfører at datafangstdato aldri skal være senere enn verifiseringsdato (men omvendt kan forekomme). Både datoer fram i tid, og objekter som har senere datafangstdato enn verifiseringsdato, finnes i databasen. I tillegg mangler det datafangstdato for en del objekter. Tabellen under oppsummerer antall gale og manglende datoer av ulike typer i AR5. Fordi kommunenummer ikke finnes som egenskap i AR5\_FLATE, kan ikke antall kommuner hvert problem forekommer i finnes ved en enkel SQL. Antall kommuner er derfor ikke tatt med for AR5.

AR5, totalt 7734573 polygoner	
Problem	Antall objekter
Mangler datafangstdato	35721
Mangler verifiseringsdato	0
Verifiseringsdato < datfangstdato	204144
Datafangstdato fram i tid	7
Verifiseringsdato fram i tid	16

Antall datoer fram i tid ser ut til å reduseres etterhvert som tiden går. Tabellene under oppsummerer antall gale og manglende datoer av ulike typer i årsversjon 2010, 2009 og 2008. Årsversjonene er klippet mot kommunegrenser, derfor er antall kommuner hvert problem forekommer i, tatt med for å demonstrere den geografiske utbredelsen av problemene.

<b>Årsversjon 2010, totalt 7858915 polygoner</b>		
<b>Problem</b>	<b>Ant. objekter</b>	<b>Ant. kommuner</b>
Mangler datafangstdato	83548	37
Mangler verifiseringsdato	0	0
Verifiseringsdato < datfangstdato	218605	322
Datafangstdato senere enn 31.12.2010	6 (703) <sup>1</sup>	26
Verifiseringsdato senere enn 31.12.2010	10 (719) <sup>1</sup>	31

<b>Årsversjon 2009, totalt 7945413 polygoner</b>		
<b>Problem</b>	<b>Ant. objekter</b>	<b>Ant. kommuner</b>
Mangler datafangstdato	940198	228
Mangler verifiseringsdato	5750	14
Mangler datafangstdato og verifiseringsdato	5750	14
Verifiseringsdato < datfangstdato	104452	139
Datafangstdato senere enn 31.12.2009	7	5
Verifiseringsdato senere enn 31.12.2009	9	7

<b>Årsversjon 2008, totalt 8911919 polygoner</b>		
<b>Problem</b>	<b>Ant. objekter</b>	<b>Ant. kommuner</b>
Mangler datafangstdato	4023062	353
Mangler verifiseringsdato	8302	29
Mangler datafangstdato og verifiseringsdato	8302	29
Verifiseringsdato < datfangstdato	5821	31
Datafangstdato senere enn 31.12.2008	90	10
Verifiseringsdato senere enn 31.12.2008	90	10

Det ser ut som om noen av problemene med gal bruk av datoer i AR5 øker eller er konstante, mens andre reduseres. Vi ser at færre og færre objekter mangler verifiseringsdato. Tallet har aldri har vært høyt, og nå er det nede i null. Antall objekter som mangler datafangstdato reduseres stadig, men gjelder fortsatt ca 0,5 % av objektene i databasen. Særlig mange som manglet datafangstdato var det i 2008, da nesten halvparten av objektene mangler denne. Antall objekter der datafangstdato er senere enn verifiseringsdato har økt, men det kan se ut som om det har stabilisert seg nå.

<sup>1</sup>For å få med alle kommuner etter første gangs periodiske ajourføring, ble ikke årsversjon av AR5 for 2010 laget før noen uker inn i 2011. Det kan derfor være naturlig at det er en del objekter med dato i januar og februar 2011. Tallene uten parentes er antall datoer som er etter februar 2011.

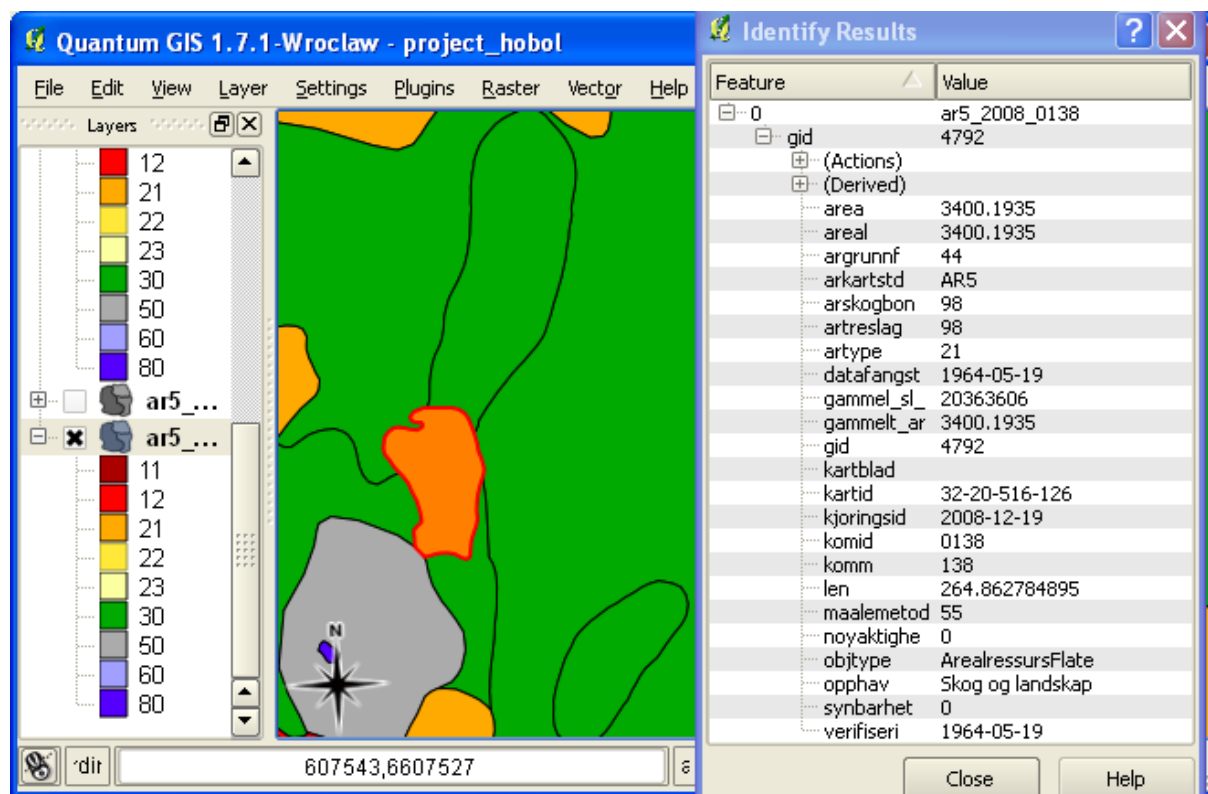
Det største problemet er polygoner med verifiseringsdato < datafangstdato. Det tilsvarer ca. 2,8 % av polygonene i AR5-layeret og årsversjon 2010 av AR5.

I DMK-databasen er det også gale datoer. Her er det kun en datoegenskap, som heter dato. Den har verdi for alle rader, men den er implementert som en tekststreng. Det har ført til en del ulovlige datoer. Det er i underkant av 8 758 000 polygoner i DMK, så prosentandelen gale og manglende datoer, er totalt sett liten.

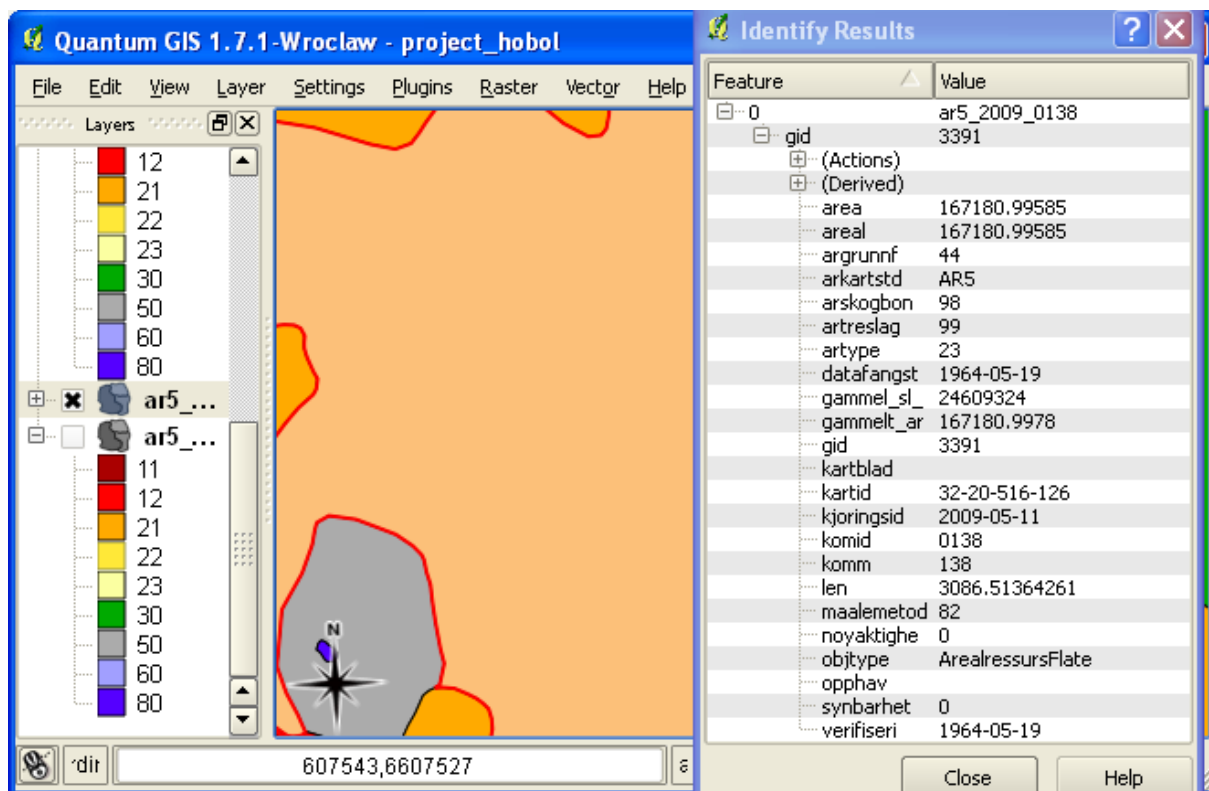
DMK, totalt 8757944 polygoner	
Problem	Antall objekter
Mangler dato	7907
Ulovlig dato	> 1321

## 12.4 Datoer oppdateres ikke eller tilbakestilles

En enkel test av dataene for Hobøl kommune, viser at det har skjedd oppdateringer som ikke er datofestet. Figur 12.2 og 12.3 viser et område som i årsversjon 2008 var fulldyrka (artype 21), i 2009 var innmarksbeite (artype 23), men dato-egenskapene var ikke oppdatert.



Figur 12.2: Data i årsversjon 2008. Datoer i 1964



Figur 12.3: Data i årsversjon 2009. Datoer fortsatt i 1964

Dersom slike feil ikke fikses, vil det føre til at området får to ulike klassifiseringer for samme tidspunkt i rom-tid sammensetning og objektmodell. For å finne alle arealer i Hobøl som var oppdatert, men der datoen ikke var oppdatert mellom AR5-årsversjonen for 2009 og 2010, laget jeg et layer i PostGIS med følgende SQL-spørring:

```
CREATE table ikkeoppdatert as
SELECT ST_Intersection(a.the_geom,b.the_geom)
FROM ar5_2010_0138 a, ar5_2009_0138 b
WHERE ST_Intersects(a.the_geom,b.the_geom)
and (a.datafangst=b.datafangst or a.verifiseri=b.verifiseri)
and (a.artype!=b.artype
or a.arskogbon!=b.arskogbon
or a.artreslag!=b.artreslag
or a.argrunnf!=b.argrunnf)
```

Query returned successfully: 10789 rows affected, 111797 ms execution time.

For å ikke få med skalkepolygoner, slettes alle polygoner der areal dividert på omkrets er mindre enn 0,25:

```
DELETE FROM ikkeoppdatert
WHERE ST_Area(the_geom)>0 and ST_Area(the_geom)/ST_Perimeter(the_geom)<0.25
```

Query returned successfully: 7101 rows affected, 906 ms execution time.

I årsversjon 2010 er ca 1 % av arealet og 3688 polygoner i Hobøl belemret med denne type feil:

```
SELECT (SELECT sum(ST_Area(the_geom)) FROM ikkeoppdatert)
/(SELECT sum(ST_Area(the_geom)) FROM ar5_2010_0138)*100
```



```
1.01037071345279
```

```
SELECT count(*)  
FROM ikkeoppdatert
```

```
3688
```

En annen test, avslører at det forekommer en del tilfeller der dataene i Hobøl tilsynelatende har reist tilbake i tid. Det kommer til syne ved at samme område har tidligere dato i årsversjon 2010 enn i årsversjon 2009. For å finne disse områdene, laget jeg et layer for områder som tilsynelatende hadde gått tilbake i tid med følgende SQL-spørring i PostGIS:

```
CREATE table tilbakeitid as  
SELECT ST_Intersection(a.the_geom,b.the_geom)  
FROM ar5_2010_0138 a, ar5_2009_0138 b  
WHERE ST_Intersects(a.the_geom,b.the_geom)  
and (a.datafangst<b.datafangst or a.verifiseri<b.verifiseri);
```

```
Query returned successfully: 5134 rows affected, 29750 ms execution time.
```

For å ikke få med skalkepolygoner, slettes alle polygoner der areal dividert på omkrets er mindre enn 0,25:

```
DELETE FROM tilbakeitid  
WHERE ST_Area(the_geom)>0 and ST_Area(the_geom)/ST_Perimeter(the_geom)<0.25
```

```
Query returned successfully: 2570 rows affected, 484 ms execution time.
```

Det gjelder ganske store områder i Hobøl, ca 2,9 % av totalarealet, og mer enn 2500 polygoner:

```
SELECT (SELECT sum(ST_Area(the_geom)) FROM tilbakeitid )  
/(SELECT sum(ST_Area(the_geom)) FROM ar5_2010_0138)*100
```

```
2.89203033997992
```

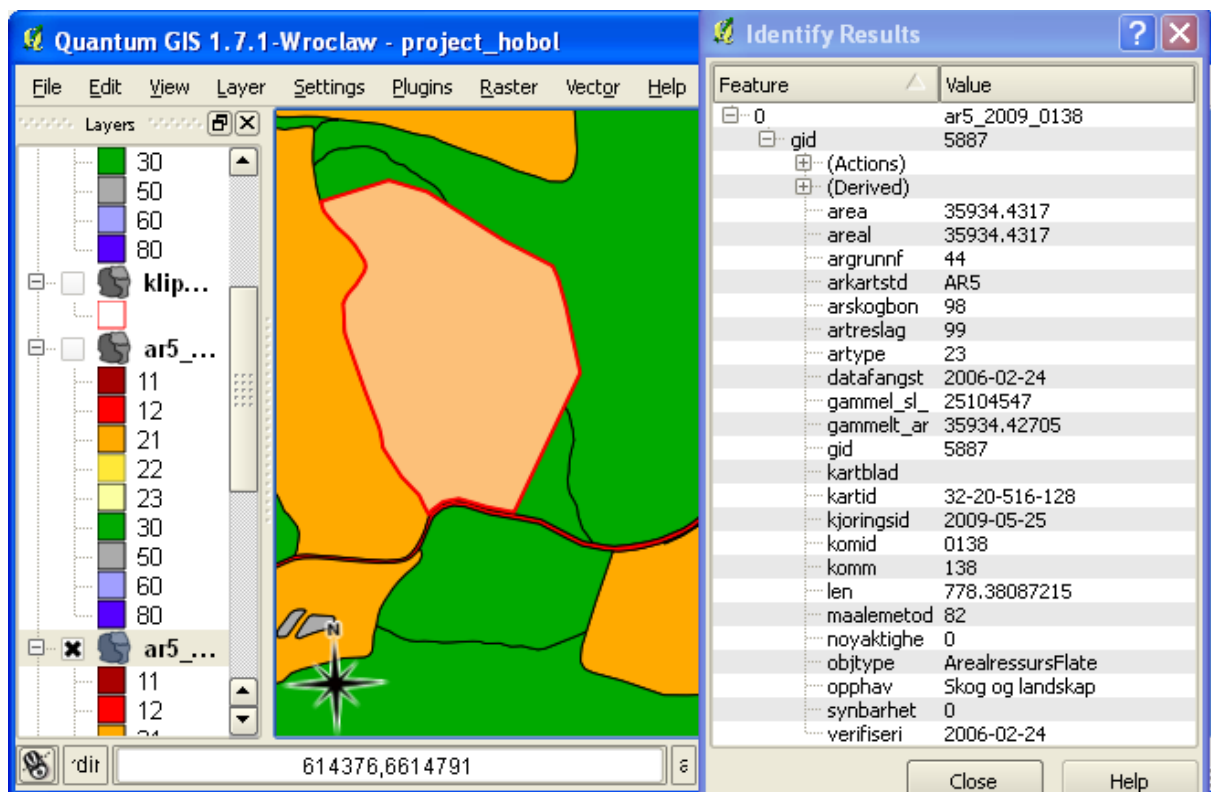
```
SELECT count(*)  
FROM tilbakeitid
```

```
2564
```

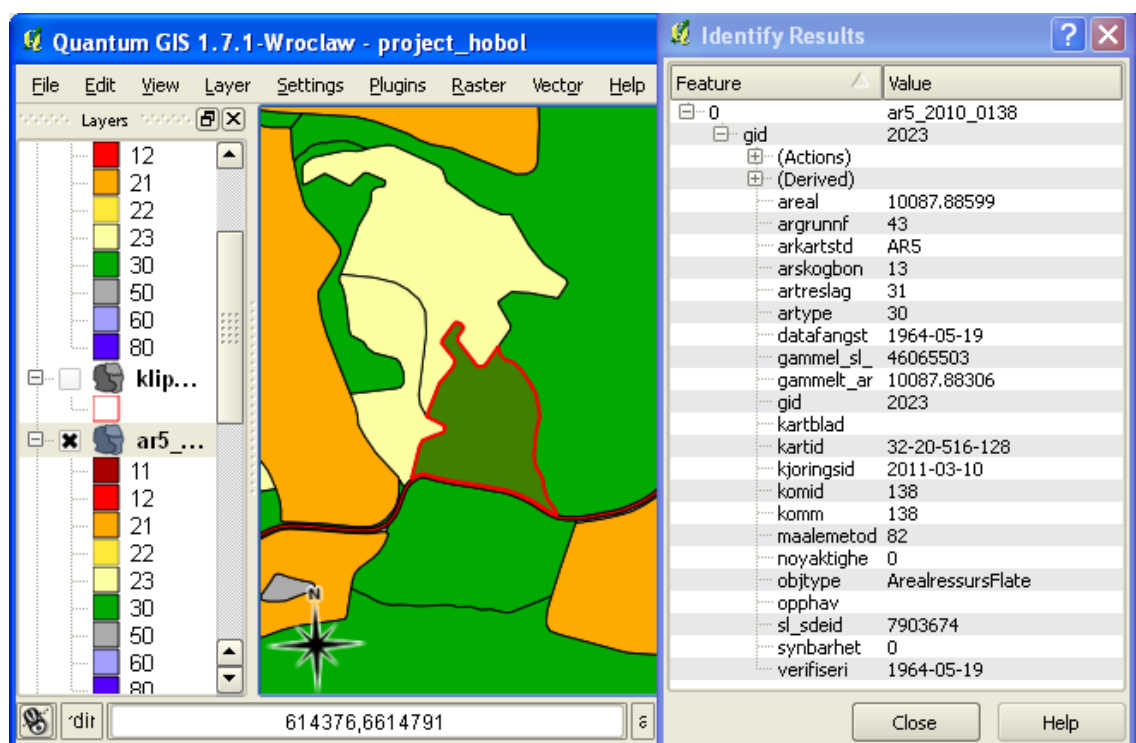
Det ser altså ut som om det har skjedd en alvorlig feil for Hobøl kommune mellom årsversjon 2009 og årsversjon 2010. Mer enn 2500 polygoner har fått en tidligere datafangstdato eller verifiseringsdato i 2010-årsversjonen enn i 2009-versjonen. Dataene har tilsynelatende foretatt en reise tilbake i tid. Det sammenfaller med periodisk ajourføring av datasettet.

Figur 12.4 viser årsversjon fra 2009, og egenskapene til polygonet som inneholder punktet med koordinater 6614791 N og 614376 Ø. Arealtype er innmarksbeite (artype 23), og både datafangstdato og verifiseringsdato er 26.02.2006. I figur 12.5 har polygonet som inneholder det samme punktet arealtype skog (artype 30), og begge datoene er 19.05.1965.

Jeg antar at feilene jeg har funnet i Hobøl forekommer i like stor grad i hele landet, men det bør undersøkes nærmere.



Figur 12.4: Data i årsversjon 2009. Datoer i 2006



Figur 12.5: Data i årsversjon 2010. Datoer i 1964

## 12.5 Konsekvenser av gale datoer i DMK og AR5

Det er generelt så mange objekter som mangler enten verifiseringsdato, datafangstdato eller begge, at det vil kunne skape problemer for informasjonssystemet for tilgjengelig jord. Heldigvis er disse problemene stort sett avtakende, det tyder på at dagens rutiner for sjekk av dette er bedre enn de var før.

Hvor store problemer feilene vil skape i TilgjengeligJord, er avhengig av metoden som velges. Jeg vil diskutere tre ulike metoder: øyeblikksbilder, rom-tid sammensetning og objekter. For øyeblikksbildene skaper manglende og gale datoer ikke så store problemer. Vi kjenner dag og klokkeslett øyeblikksbildene representerer. Fordi historiske data for AR5 er tilgjengelige i årsversjonene, er det naturlig å lage årsversjoner av TilgjengeligJord, og da kan de arve tidfestinga av AR5-øyeblikksbildet det er basert på.

For rom-tid sammensetning og objektmodellen, er det litt mer trøblete av vi mangler datoer, at ajourføring har skjedd uten at datoer er oppdatert, at arealer «reiser tilbake i tid», eller at datoer ligger fram i tid. Det fører til at vi ikke får heldekkende datasett for alle mulige tidspunkt, fordi vi får temporale gap i tidslinja og overlapp.

At verifiseringsdato er tidligere enn datafangstdato, tyder på at «noen» glemmer å oppdatere denne datoen. En SQL-spørring der rader med datafangstdato større enn verifiseringsdato fordeles på opphav, tyder på at «noen» er både Skog og landskap og kommunene. Forglemmelsen ser ut til skje også i dag, så her er en innskjerping av rutiner å anbefale. Effekten av dette problemet er liten for øyeblikksbilder, men for rom-tid sammensetning og objektmodell, vil effekten bli at gyldig tid er senere enn databasetid. Det må forstås som at fremtidige endringer er lagt inn i basen, og det er ikke lov i AR5.

## 12.6 Bruk av verifiseringsdato som «assertion time»

For å kunne si at verifiseringsdato er det samme som «assertion time» slik «assertion time» er definert i [20], må det settes ny verifiseringsdato dersom det rettes opp en feil i kartet uten at datafangstdatoen skal oppdateres.

Gitt at et flybilde brukes ved ajourføring av et område. Dataene ferdigstilles og legges inn i databasen. Senere tas dataene fram igjen, og de sammenliknes med det samme flybildet. Da oppdager tolkeren at det er gjort en feil, som hun ønsker å rette opp. Per i dag vil verken datafangstdatoen eller verifiseringsdatoen bli oppdatert med ny dato. For datafangstdatoen er det både logisk og riktig, siden datafangsten fortsatt er gjort den dagen flybildet er tatt. Imidlertid er det gjort en verifisering som har ført til en oppdatering. Jeg mener derfor at verifiseringsdatoen burde vært oppdatert. Med denne tilleggsregelen for bruk av verifiseringsdato, ville verifiseringsdato være det samme som «assertion time», og vi ville hatt et datoregime i AR5 som tok høyde for feilretting gjennom to ortogonale tidsakser.

## 13 Brukstilfeller

Før datasystemet designes, er det viktig å få klarhet i hvilke spørsmål det skal kunne svare på.

Dyrka jord og dyrkingsjord har et vern i henhold til Jordlova [26]. I §9 står det:

«Dyrka jord må ikkje brukast til føremål som ikkje tek sikte på jordbruksproduksjon. Dyrkbar jord må ikkje disponerast slik at ho ikkje vert eigna til jordbruksproduksjon i framtida.».

Jordvern er de siste årene blitt et viktig politisk tema, og det er et uttalt mål å bremse nedbygging av jordsbruksarealer. Samtidig vet vi at mye av de beste arealene for jordbruk befinner seg i pressområder på det sentrale Østlandet, Jæren og i Trøndelag. Kommunene skal hvert år rapportere inn hvor store arealer jordbruksland og dyrkingsjord de planlegger å bygge ned gjennom rapporteringssystemet Kostra. Man vet imidlertid at tallene som kommer inn kan være mangelfulle og ufullstendige. Kan en veldesignet database for TilgjengeligJord supplere Kostra-tallene, ikke med hva som er planlagt, men med hva som faktisk er blitt nedbygd av jordbruksarealer og dyrkingsjord i norske kommuner? Hvilke konkrete spørsmål må informasjonssystemet da kunne svare på?

Geir-Harald Strand er avdelingsdirektør ved Skog og landskap, og har tidligere arbeidet med å lage et system for oversikt og statistikk over utvikling av jordbruksarealer og dyrkingsjord sammen med min veileder, Knut Bjørkelo. Etter en innledende fase ble prosjektet skrinlagt, da man den gang (i 2005) fant at man verken hadde nødvendig teknologi eller ressurser for å løse det som skulle vise seg å bli en nokså komplisert oppgave. I en samtale om temaet [64], uttrykte Strand at det i første omgang er viktigst med god statistikk for jordbruksareal og dyrkingsjord for ulike geografiske inndelinger. På lang sikt er det aktuelt med et regnskapssystem med tidsserier som gir trender over tid og muligheten til å se mønstre i utviklingen. Strand mener det ikke er så viktig å vite akkurat hva som har skjedd med et bestemt landområde, men at det er helheten som er viktig. Han skisserer i samtalen tre spørsmål om jordbruksarealer og dyrkingsjord som man trenger svar på:

- Hvor mye jordbruksareal og dyrkingsjord har vi?
- Hvor er den?
- Hva har skjedd med det vi hadde av jordbruksareal og dyrkingsjord på et gitt tidspunkt: Er den nedbygd, nydyrket, fortsatt dyrket eller er det ingen vesentlig endring?

I et internt Skog og landskap-notat fra 2005 [10], skriver Knut Bjørkelo følgende liste over spørringer datasystemet bør kunne besvare:

- Finn all dyrkingsjord
- Finn all tidligere dyrkingsjord som er nedgradert etter lokal vurdering
- Finn alt tidligere jordbruksareal som er nedbygd
- Finn all dyrkingsjord som er nedbygd
- Finn all attgrodd tidligere fulldyrka jord
- Finn arealer som er omgradert ved lokal vurdering
- Finn all nydyrking på ikke dyrkingsjord
- Finn all nydyrking på tidligere skog eller myr

I 2005 trodde man at man skulle åpne for omgradering av dyrkingsjord gjort av kommunene (såkalt lokal vurdering). Dette har jeg i denne oppgaven valgt å se bort fra. Dersom det blir aktuelt, vil man kunne se på det som feilretting. De to punktene som omhandler omgradering/nedgradering er derfor ikke så aktuelle lenger, men kan alternativt byttes ut med «Finn all tidligere dyrkingsjord som er rettet til ikke dyrkingsjord» og «Finn arealer som er rettet».

# 14 Systemarkitektur for ny database for TilgjengeligJord

Databasene for DMK og AR5 slik de er implementert hos Skog og landskap i dag, låser tiden, kontrollerer attributt og måler sted. Årsversjonene av AR5 (som er øyeblikksbilder), låser også tiden i hvert enkelt øyeblikksbilde, og kan ikke sies å være «ekte» romlig-temporale databaser. Databasene med endringsdata for AR5 måler både tid og sted, og kan dermed kategoriseres som ekte romlig-temporale data

Det er primært tilstanden man er interessert i i disse databasene, ikke selve hendelsen som førte til endring. Man er for eksempel interessert i at i 1988 var et område skogkledt, mens i 1990 var det bebyggt. Det representerer tilstand. Hendelsene som førte til endringen (for eksempel hogst), er ikke like interessant i arealdekke-perspektivet.

Gjennom samtale med Geir-Harald Strand [64] ble det klart at et hovedproblem med markslagsmodellen er at den passet best analogt, den gangen hovedformålet var å lage det Bertin kalte et lesbart kart [9]. I dag er dataene digitale, og det er krav til flerbruk. De samme dataene skal brukes i jordvernsammenheng, til å fordele tilskudd og lage arealstatistikk. Det er også ønskelig å finne kartografiske framstillinger som gir oss ny informasjon og kunnskap, såkalte sebare kart i Bertins terminologi [9].

## 14.1 Datamodell for TilgjengeligJord – Et nytt datasett laget fra DMK og AR5

Nye krav gjør det nødvendig med nye og bedre datamodeller. Det er på tide å gå bort fra å modellere kartet og over til å modellere verden. Det er en krevende oppgave, og det er ikke sikkert de eksisterende dataene vil passe inn i en ny modell. Derfor fortsetter jeg delvis i gammel tradisjon, men utvider gjeldende romlige modell med attributter for å håndtere tidsdimensjonen, i tillegg til å normalisere informasjonen om dyrkingsjord. Normaliseringen vil gjøre informasjonen om dyrkingsjord mye lettere tilgjengelig enn den er i DMK i dag.

Jordbruksarealer oppdateres i AR5. Dyrkingsjord-informasjon ligger i DMK, men er ikke ajourført. I AR5 ligger også oppdatert informasjon om samferdselsområder, bebyggt areal og vann. Informasjonen om dyrkingsjord kan derfor oppdateres administrativt på bakgrunn av AR5. For å lettere kunne se jordbruksareal og dyrkingsjord i sammenheng, foreslår jeg å legge all informasjonen i samme datasett, som jeg velger å kalle for «TilgjengeligJord».

### 14.1.1 TilgjengeligJord skal inneholde både jordbruksareal og dyrkingsjord

Noen av brukstilfellene fra kapittel 13, kan besvares uten noen spesiell tilpasning av allerede eksisterende datasett og systemarkitektur. Det gjelder spørsmål om jordbruksarealer, hvor disse arealene er, hvor mange dekar det er, og delvis hvordan jordbruksarealene utvikler seg gjennom nedbygging og nydyrking. Denne informasjonen kan hentes fra AR5, årsversjonene av AR5 og endringsanalysene.

Informasjonen om dyrkingsjord kan trekkes ut fra DMK og oppdateres ved hjelp av ajourført AR5. Vi trenger også å se dyrkingsjord i sammenheng med jordbruksareal, siden overflatedyrka jord og innmarksbeite kan være klassifisert som begge deler samtidig. For å unngå å rapportere disse arealene dobbelt i statistikker og kart der begge typene inngår, trenger vi en enkel metode for å identifisere overflatedyrka jord og innmarksbeite som også er dyrkingsjord. Det er tungvint og tidkrevende dersom man alltid, og i alle applikasjoner, må gjøre romlige spørringer for å finne disse arealene. Det vil også medføre stor risiko for å gjøre feil. Da er det bedre å gjøre en kvalitetssikret operasjon en gang, og så gjenbruke resultatet.

Datasettet «TilgjengeligJord» med både jordbruksarealer og dyrkingsjord, kan oppdateres på basis av siste årsversjon av AR5 én gang i året, og gjenbrukes i ulike sammenhenger som Internett-løsninger, statistikkproduksjon og analyser. Dermed får vi et datasett som nyter godt av ajourføringsopplegget for AR5 samtidig som det vil være lett å se jordsbruksarealer og dyrkingsjord i sammenheng.

### 14.1.2 Markslagsegenskaper i TilgjengeligJord

Datamodellen for TilgjengeligJord vil variere etter hvilken modell som velges. Egenskapene som utgjør klassifisering av markslag, vil imidlertid alltid være de samme. Disse egenskapene er:

- Dyrkingsjord (egenskapsnavn dyrkingsjord), spesifisert i tabell 14.1.
- Driftsforhold (egenskapsnavn driftsforhold) spesifisert i tabell 14.2.
- Myr (egenskapsnavn myr) spesifisert i tabell 14.3.
- Vanning og drenering (egenskapsnavn vanningOgDrenering) spesifisert i tabell 14.4.
- Stein og blokk (egenskapsnavn steinOgBlokk) spesifisert i tabell 14.5.
- Arealtype (egenskapsnavn artype). Samme spesifikasjon som for AR5, se tabell 10.6.

Koden for «Ukjent / ikke relevant» for myr, vanningOgDrenering og steinOgBlokk bør vurderes delt i to dersom Skog og landskap velger å etablere TilgjengeligJord som et nasjonalt datasett. I denne oppgaven velger jeg å bruke samme kode for både «ikke relevant» og «ukjent». For driftsforhold velger jeg å dele opp i to koder, fordi man ved oppdatering av TilgjengeligJord kan få nye arealer med dyrkingsjord som ikke

Dyrkingsjord		
Kode	Verdi	Beskrivelse
0	Ukjent	Det er ukjent hvorvidt arealet er dyrkingsjord eller ikke
1	Dyrkingsjord	Arealet tilfredstiller kravene til dyrkingjord
2	Ikke dyrkingsjord	Arealet tilfredsstiller ikke kravene til dyrkingsjord

Tabell 14.1: Koder for egenskap dyrkingsjord

Driftsforhold		
Kode	Verdi	Beskrivelse
0	Ikke relevant	Arealet er ikke dyrkingsjord
1	Lettbrukt	Lettbrukt dyrkingsjord registrert i ØK/DMK
2	Mindre lettbrukt	Mindre lettbrukt dyrkingsjord registrert i ØK/DMK
3	Tungbrukt	Tidligere fulldyrka tungbrukt jord. Har hatt ATIL 21 og JORD=91 i DMK
4	Ukjent	Dyrkingsjord av ukjent driftsforhold. Er/har vært fulldyrka i AR5 (ARTYPE 21), men ikke verken fulldyrka eller dyrkingsjord i ØK/DMK.

Tabell 14.2: Koder for egenskapen driftsforhold

Myr		
Kode	Verdi	Beskrivelse
0	Ukjent/ikke relevant	Det er ukjent hvorvidt arealet er myr eller arealet er ikke dyrkingsjord
1	Myr	Jordbruksareal/dyrkingsjord på myr
2	Ikke myr	Jordbruksareal/dyrkingsjord på fastmark (ikke myr)

Tabell 14.3: Koder for egenskapen myr

har vært kartlagt som dyrkingsjord eller fulldyrka i ØK-kartlegginga. Da vil vi ikke kjenne egenskapen for driftsforhold derfra. Det er neppe aktuelt med nytt feltarbeid for å kartlegge ny dyrkingsjord. Jeg lager derfor en egen kode for ukjent driftsforhold. Den tilsvarer unionen av lettbrukt, mindre lettbrukt og tungbrukt.

Det betyr at for nye arealer dyrkingsjord, vil alle områder som kan fulldyrkes klassifiseres som dyrkingsjord, uansett driftsforhold. Det bør vurderes om arealer som har vært fulldyrka tungbrukt jord i DMK også skal klassifiseres som dyrkingsjord, slik at det blir likt. Det innebærer i så fall en liten omdefinering av begrepet dyrkingsjord i samsvar med figur 3.1. Hvorvidt det skal gjøres er en markslagsfaglig avgjørelse, og



Vanning og drenering		
Kode	Verdi	Beskrivelse
0	Ukjent/Ikke relevant	Det er ikke kjent om arealet må dreneres eller er tørkesvakt eller arealet er ikke dyrkingsjord
1	Ikke selvdrenert	Ikke selvdrenert dyrkingsjord. Areal som ved fulldyrking stort sett må grøftes systematisk.
2	Selvdrenert	Selvdrenert dyrkingsjord. Areal som ved fulldyrking ikke trenger systematisk vanning,
3	Tørkesvak	Tørkesvak dyrkingsjord. Areal med tørkesvak sandjord som er lite egna til dyrkinguten kunstig vanning.

Tabell 14.4: Koder for egenskapen vanningOgDrenering

Stein og blokk		
Kode	Verdi	Beskrivelse
0	Ukjent/Ikke relevant	Blokk- og steinnhold er ukjent eller arealet er ikke dyrkingsjord
1	Ikke blokkrik	Ikke blokkrik dyrkingsjord. Mindre enn 30 m <sup>3</sup> stein og blokker pr. dekar i de øverste 30 cm av mineraljorda.
2	Blokkrik	Blokkrik dyrkingsjord. Mellom 30 og 100 m <sup>3</sup> stein og blokker pr. dekar i de øverste 30 cm av mineraljorda.
3	Svært blokkrik	Svært blokkrik dyrkingsjord. Areal med mer enn 100 m <sup>3</sup> stein og blokker pr. dekar i de øverste 30 cm av mineraljorda.

Tabell 14.5: Koder for egenskapen steinOgBlokk

må tas av det marslagsfaglige miljøet ved Skog og landskap.

### 14.1.3 Konseptuell modell og lagringsmodell

TilgjengeligJord lages med basis i DMK og AR5, som begge er klassifiserte arealdekker og diskrete polygondekker i henhold til ISO 19123. Det skal også TilgjengeligJord være. Polygondekker er en form for feltmodell. I motsetning til AR5, er TilgjengeligJord ikke heldekkende for Norge, det har alle arealer som har vært/er klassifisert som fulldyrka jord eller dyrkingsjord i DMK og AR5 som geografisk domene.

Som for AR5 og DMK, velger jeg å representere TilgjengeligJord som vektordata i form av polygoner i databasen. Avgrensninglinjer lagres ikke. Det er mulig linjene burde vært med, og at de burde arvet kvalitetsinformasjon fra linjene i DMK/AR5. Avgrensninglinjene i AR5 og DMK er ikke lagret i geodatabaser ved Skog og landskap, de er kun tilgjengelige i SOSI-filer. Det gjør det vanskelig å bruke dem. Uansett er det egenskapene ved polygonene som står i sentrum når brukstilfellene skal behandles. Jeg velger derfor å konsentrere meg om polygonene i denne oppgaven, og ser helt bort i fra avgrensninglinjene.

Figur 14.1 viser konseptuell datamodell for TilgjengeligJord. Modellen følger SOSI-standard for modellering av geometri. Håndtering av tid/historikk er ikke tatt med i modellen, da historikk vil håndteres forskjellig avhengig av hvilken romlig-temporal modell som velges.

### 14.1.4 TilgjengeligJord skal håndtere tid/historikk

For å se hva som har skjedd med jordbruksarealene og dyrkingsjorda, trenger man å ta vare på historikken. Da trenger man en datamodell som håndterer tid, og man trenger temporale data å fylle databasen med. TilgjengeligJord må få en datamodell som tar vare på historikk etter hvert som det oppdateres med årsversjoner AR5.

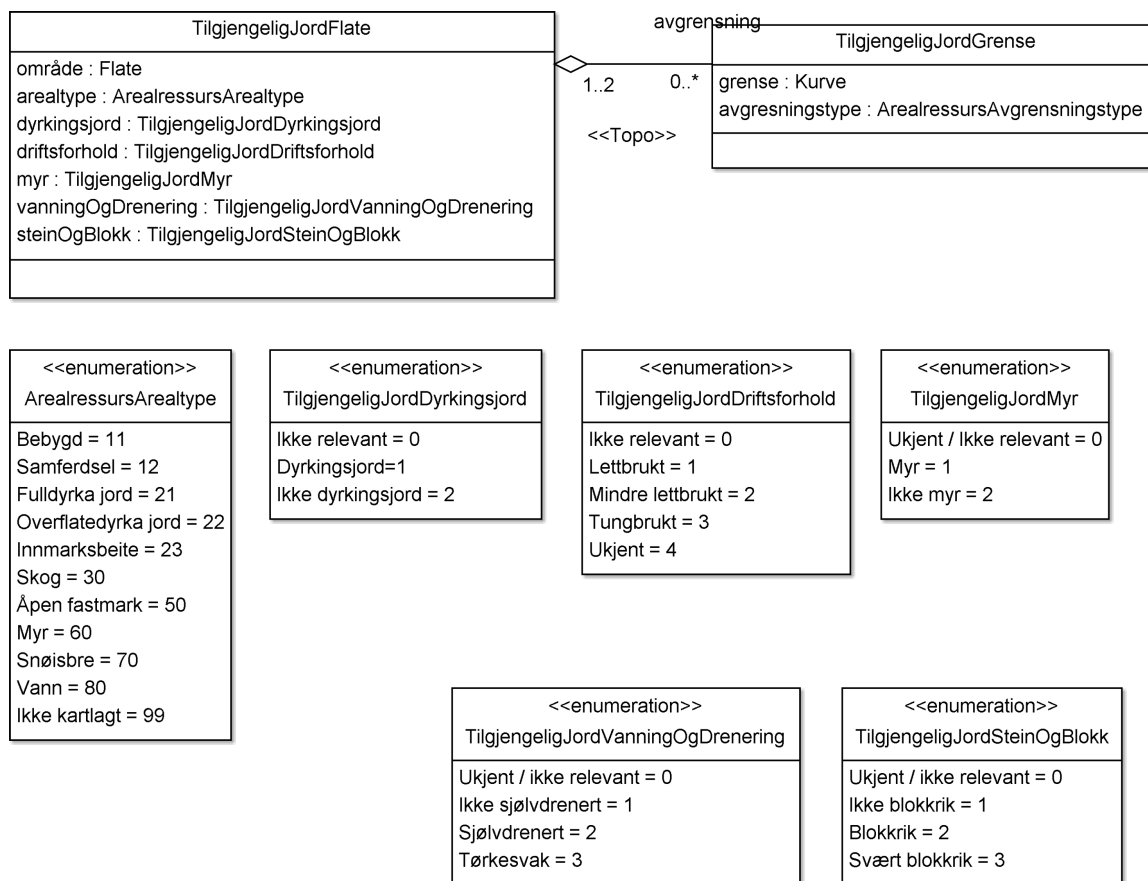
For DMK-databasen, har vi ingen historikk. Datering av informasjonen strekker seg fra 1956 til 2009. Jeg etablerer et omkodet uttrekk av jordbruksareal og dyrkingsjord fra nåværende tilstand av DMK. Uttrekket er å betrakte som null-situasjonen for TilgjengeligJord.

Siden 2008 har Skog og landskap tatt vare på årsversjoner av AR5 som øyeblikksbilder, og siden 2011 er alle endringer mellom to versjoner av AR5 tatt vare på i endringsanalysen.

Det bør skje en innskjerping av dateringen av AR5-polygonene for å øke datasettets evne til å ivareta endring og historikk. Man må også håndtere skillet mellom reelle og ikke-reelle endringer bedre enn i dag. Dateringen av AR5-polygoner ved feilretting må endres, slik at verifiseringsdato kan brukes som «assertion time» i en bitemporal modell. Datafangstdato kan brukes som gyldig tid.

Videre foreslår jeg at verifiseringsdato og datafangstdato for gammelt og nytt polygon tas inn i endringsanalysen. Det vil gi mulighet for en kontroll av datoene for ajourførte arealer for å hindre temporale gap og overlapp. Det er nødvendig å se gammelt og nytt datasett i sammenheng for å finne slike feil, og det er akkurat det endringsanalysen gjør.

Å ta med verifiseringsdato og datafangstdato i endringsanalysen, vil også gjøre det mulig å bruke endringsanalysen mer aktivt i rapportering av endringer i den virkelige verden. I dag er endringsanalysen først og fremst for å identifisere endringer mellom



Figur 14.1: Konseptuell datamodell for TilgjengeligJord (uten tid).

to datasett.

Det er som sagt tilstand som er interessant i TilgjengeligJord. Det modelleres best ved egenskaper med type «TM\_Period» i henhold til ISO 19108, og implementeres som et starttidspunkt og et sluttidspunkt, begge med datatype «date».

### 14.1.5 Prinsipper for lagring og oppdatering av TilgjengeligJord

Dersom TilgjengeligJord skal kunne brukes til å følge utviklingen av jordbruksarealer og dyrkingsjord, er det en del prinsipper som må gjelde ved enhver oppdatering, uavhengig om modellen som brukes er øyeblikksbilder, rom-tid sammensetning eller objektmodell.

- Datasettet TilgjengeligJord skal etableres som et nasjonalt datasett hos Skog og landskap, fortrinnsvis i PostGIS. Det betyr at vi må forholde oss til funksjonalitet i gjeldende versjon av PostGIS, som er versjon 1.5.3. Da er lagring av polygoner i simple feature-standardens geometrimodell det opplagte alternativet. Riktig nok er det mulighet for lagring av polygoner i en topologimodell, men

det er ikke en del av den offisielle dokumentasjonen for PostGIS versjon 1.5.3. Skog og landskap har heller ikke tatt topologimodellen i PostGIS i bruk. Topologi ville sannsynligvis gjort det enklere å oppdatere datasettet uten å få overlapp og gap. Som vi skal se, skaper den enkle simple feature-geometrimodellen en del problemer.

- Det eneste som kan gjøre et areal som er klassifisert som dyrkingsjord til «ikke dyrkingsjord», er hvis området blir bebygd, samferdselsareal, vann eller snøisbre.
- Dyrkingsjord må ikke opphøre å være dyrkingsjord om den dyrkes opp. For fulldyrka jord, betyr det at den får kode for driftsforhold. For overflatedyrka jord og innmarksbeite vil den også merkes som dyrkingsjord (dyrkingsjord = 1)
- Nye områder med overflatedyrka jord eller innmarksbeite på arealer som ikke er dyrkingsjord i TilgjengeligJord fra før, får egenskapen «ikke dyrkingsjord».
- Fulldyrka jord som er registrert som lettbrukt eller mindre lettbrukt i DMK som går over til annen arealtype enn samferdsel, bebygd, vann eller snøisbre, skal klassifiseres som dyrkingsjord med tilhørende driftsforhold.
- Fulldyrka jord som har vært registrert som tungbrukt, og som går over til annen arealtype enn bebygd, samferdsel, vann eller snøisbre, skal registreres som «ikke dyrkingsjord» (med mindre man bestemmer seg eksplisitt for det). Driftsforhold settes til tungbrukt.
- For arealer som er fulldyrka i AR5 (og derfor blir det i TilgjengeligJord), men som ikke var det i DMK, vet vi ikke noe om driftsforhold. Dersom disse går over til annen arealtype, utenom samferdsel, bebygd eller vann, skal den klassifiseres som dyrkingsjord av ukjent driftsforhold (dyrkingsjord=4).
- Koden for grunnforhold i AR5 skal oppdatere egenskapen myr i TilgjengeligJord. Myr settes til 1 når argrunnf er 45, myr settes til 2 når argrunnf er 44. Hvis argrunnf er 98 eller 99, beholder myrkoden sin opprinnelige verdi.
- Egenskapene vanningOgDrenering og steinOgBlokk i TilgjengeligJord oppdateres ikke fra AR5. De settes til «ikke relevant» når et område med dyrkingsjord blir til samferdsel, vei, vann eller snøisbre, ellers beholder de sin opprinnelige verdi.

### 14.1.6 TilgjengeligJord kan implementeres med ulike romlig-temporale modeller

TilgjengeligJord-datasettet kan bygge på ulike romlig-temporale modeller. Øyeblikksbilder er lettest å implementere. Mer komplekse, med tilhørende fordeler og ulemper, er objektmodellen og Langrans rom-tid sammensetning. Jeg vil gå igjennom alle tre modellene, diskutere hva som skal til for å etablere og vedlikeholde TilgjengeligJord-datasettet og hente informasjon gjennom spørringer i de tre ulike alternativene, og se på noen fordeler og ulemper ved hver av dem.

Øyeblikksbilder og rom-tid sammensetning implementeres i PostGIS med data for Hobøl kommune. Objektmodellen diskuteres kun teoretisk. Programkoden for implementering av øyeblikksbilder og rom-tid sammensetning er ikke fullstendig gjennomarbeidet og testet, men er ment som en skisse for hvordan det kan gjøres og et eksperiment i forhold til ytelse og problemstillinger som måtte oppstå.

Jeg velger en bitemporal løsning for rom-tid sammensetning og objektmodell. Selv om verifiseringsdato i AR5 per i dag ikke fyller kravene til bitemporal databasetid, vil jeg allikevel bruke den som det. Verifiseringsdato vil ikke sammenfalle med transaksjonsdato, den vil normalt være tidligere enn transaksjonstiden. Derfor blir det bitemporal tid i form av «assertion time» i rom-tid sammensetning og objektmodell.

## 14.2 TilgjengeligJord som øyeblikksbilder

TilgjengeligJord-datasettet kan lages som øyeblikksbilder i sekvensielle årsversjoner. Første versjon er da et modifisert uttrekk av jordbruksarealer og dyrkingsjord fra DMK. Det er ikke så lett å avgjøre hva som er riktig årstall for dette datasettet, siden noen områder er oppdatert i 2009, mens andre ble oppdatert på 1950-tallet. Jeg foreslår derfor å ikke gi dette datasettet et årstall, men rett og slett kalle det for TilgjengeligJord\_DMK.

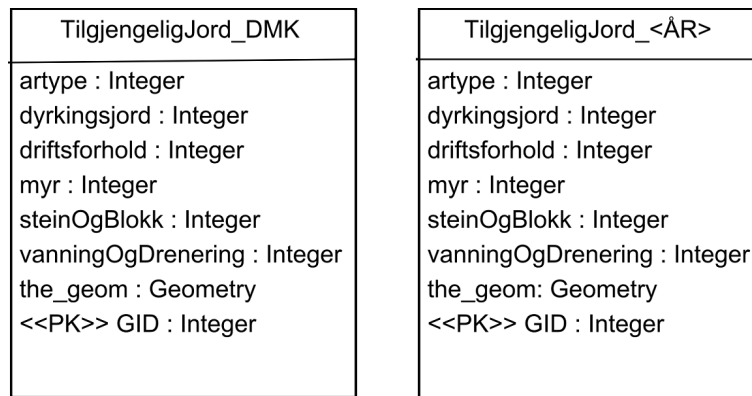
Deretter lages årsversjoner gjennom et overlay (romlig intersect) mellom siste versjon av TilgjengeligJord og årsversjoner av AR5. Ved å ta vare på resultatet fra intersect-operatoren i et eget datasett, kunne man også bevart en oversikt over endringene som ha skjedd. Polygonene i dette datasettet der egenskapene for artype eller grunnforhold i årsversjonen av AR5 er annerledes enn egenskapene for artype eller myr i den forrige versjonen av TilgjengeligJord representerer endring mellom de to datasettene.

TilgjengeligJord\_2008\_test lages ved et overlay mellom TilgjengeligJord\_DMK\_test og AR5\_KOMM\_FLATE\_2008\_test, TilgjengeligJord\_2009\_test lages ved et overlay mellom AR5\_KOMM\_FLATE\_2009\_test osv. Datamodellen for TilgjengeligJord-datasettene finnes i figur 14.2.

For å få et enklere datasett, velger jeg å slå sammen nabopolygoner med de samme kodene for arealtype, dyrkingsjord, driftsforhold, myr, vanningOgDrenering og steinOgBlokk. Datoegenskapene kan være ulike for polygoner som slås sammen. Derfor tas ikke datoene inn i datasettet.

Dersom en velger å ta med dato-egenskapene, vil antallet polygoner i datasettet bli mye større. Det vil også bli mange små polygoner. Det er en del feil bruk av datoer i AR5 og DMK. Det gjør informasjonsverdien til datoene mindre enn den kunne vært, og tapet av informasjon ved å utelate dem er ikke så stort.

Utgangspunktet for den praktiske delen av oppgaven er testdataene som beskrevet i avsnitt 10.6. En oversikt over layer med basisdata i PostGIS finnes i tabell 10.8. Resultatet av arbeidet er en rekke lagere. Disse er presentert i tabell 14.6. I arbeidsprosessen er det brukst en rekke midlertidige layer. Disse finnes i tabell 14.7.



Figur 14.2: Datamodell for TigjengeligJord som øyeblikksbilder

### 14.2.1 Arbeidets gang

Se vedlegg 2 for programkode og spørringer som ble brukt for å gjennomføre arbeidet. Jeg velger en stegvis framgangsmåte. Den krever mange operasjoner og mange mellomlagringer, men gjør det enklere å verifisere hver operasjon.

#### Lager basisversjonen av TilgjengeligJord ut fra jordbruksareal og dyrkingsjord i DMK

1. Lager layeret TilgjengeligJord\_DMK\_test.
2. Henter data fra omkodet DMK i Matjord\_0138\_test-layeret inn i TilgjengeligJord\_DMK\_test. Slår sammen nabopolygoner der artype, dyrkingsjord, driftsforhold, myr, vanningOgDrenering og steinOgBlokk er like.
3. Sletter polygoner med areal = 0. Fjerner små polygoner ved å slå dem sammen med hensiktsmessig nabopolygon.(Det var ingen slike polygoner i TilgjengeligJord\_DMK\_test.)
4. Lager polygoner for ikke klassifisert (fyll-polygoner). Deler dem opp med rutenett så de ikke skal bli for tunge.
5. Fyller inn ikke-klassifisert i TilgjengeligJord\_DMK\_test. Da får jeg et heldekkende datasett. Det gjør det lettere å gjøre romlige intersect-operasjoner på dataene, og datasettenes areal kan sammenliknes med arealet til klippepolygonet. Differansen bør være minimal.
6. Sjekker at polygoner er enkle og gyldige.
7. Sjekker at arealet til TilgjengeligJord\_DMK\_test stemmer overens med arealet til klippepolygonet, og at det ikke er arealer av betydning som overlapper. Spørringene bør gi 0 eller et svært lite tall som resultat.

#### Oppdatering fra årsversjon av AR5

Disse punktene gjentas hver gang man skal lage et nytt øyeblikksbilde med en ny årsversjon av AR5:

1. Lager layeret «TilgjengeligJord\_YYYY\_test».
2. Gjør en romlig snittoperasjon («intersect») mellom siste øyeblikksbilde av TilgjengeligJord og årsversjon av AR5. Om ønskelig, kan vi lage en eget layer med alle endringene mellom de to layerne. Da plukker vi ut de polygonene som har ulik arealtype eller ulik myr/grunnforhold for de to datasettene. Layeret med resultatet av den romlige snittoperatoren, skal ha samme areal som klippepolygonet, og ikke inneholde overlapp.
3. Henter data fra den romlige intersect-en i forrige punkt. Tar med dyrkingsjord fra TilgjengeligJord og jordbruksarealer fra AR5. Slår sammen nabopolygoner der artype, dyrkingsjord, driftsforhold, myr, vanningOgDrenering og steinOgBlokk er like.
4. Sletter polygoner med areal = 0. Fjerner små polygoner ved å slå dem sammen med hensiktsmessig nabopolygon.
5. Lager polygoner for ikke klassifisert (fyll-polygoner). Deler dem opp med rute-nett så de ikke skal bli for tunge.
6. Fyller inn ikke-klassifisert i «TilgjengeligJord\_YYYY».
7. Sjekker at polygoner er enkle og gyldige.
8. Sjekker at arealet til «TilgjengeligJord\_YYYY\_test» stemmer overens med arealet til klippepolygonet, og at det ikke er arealer av betydning som overlapper
9. Oppdaterer med nye polygoner dyrkingsjord og polygoner som ikke er dyrkingsjord lenger. Bruker artype fra AR5 årsversjon til å søke ut polygonene.

LAYERNAVN	BESKRIVELSE
tilgjengeligjord_dmk_test	Basisversjon av TilgjengeligJord-datasettet laget fra omkodet DMK
tilgjengeligjord_2008_test	Årsversjon 2008 av TilgjengeligJord. Laget ved hjelp av tilgjengeligjord_dmk_test og ar5_2008_0138_test.
tilgjengeligjord_2009_test	Årsversjon 2009 av TilgjengeligJord. Laget ved hjelp av tilgjengeligjord_2008_test og ar5_2009_0138_test.
tilgjengeligjord_2010_test	Årsversjon 2010 av TilgjengeligJord. Laget ved hjelp av tilgjengeligjord_2009_test og ar5_2010_0138_test.

Tabell 14.6: Layeret med årsversjonene av TilgjengeligJord for testområdet på 9 km<sup>2</sup>.

Fordelen med denne måten å lage TilgjengeligJord-datasettet på, er at det er konseptuelt enkelt. Det er i prinsippet forholdsvis enkelt å produsere, det er lett å skjønne hva TilgjengeligJord-datasettet er, og hvordan det kan brukes. Ulempene er for det første at det tar mye plass, da alt dupliseres hvert år. En annen ulempe, er at det er vanskelig å finne endringer mellom to øyeblikksbilder som ikke representerer påfølgende årsversjoner, da må man skrive kompliserte og potensielt tunge spørringer. Ved å ta vare på resultatene av snittoperasjonene, lagrer vi alle endringer mellom

LAYERNAVN	BESKRIVELSE
tilgjengeligjord_work_test	Holder overlay mellom forrige versjon av TilgjengeligJord og den versjonen av AR5-årsversjon det jobbes med.
klippepol grid	Layer med ett rektangulært polygon på ca 9 km <sup>2</sup> som brukes til å klippe Hobøl-basisdata for å få mindre data å jobbe med. Rutenett som brukes for å klippe opp polygonene som fyller ut tilgjengeligJord-datasettene til heldekkende datasett.
smaa_dmk_test smaa_2008_test smaa_2009_test smaa_2010_test	Alle polygoner fra tilgjengeligjord_dmk_test der forholdet mellom areal og omkrets er mindre enn 0.25 (såkalte slivers). Alle polygoner fra tilgjengeligjord_2008_test der forholdet mellom areal og omkrets er mindre enn 0.25 (såkalte slivers). Alle polygoner fra tilgjengeligjord_2009_test der forholdet mellom areal og omkrets er mindre enn 0.25 (såkalte slivers). Alle polygoner fra tilgjengeligjord_2010_test der forholdet mellom areal og omkrets er mindre enn 0.25 (såkalte slivers).
fyll_dmk_test fyll_dmk_ruter_test	Brukes for å lage data for å fylle ut tilgjengeligjord_dmk_test der det ikke er jordbruksareal eller dyrkingsjord. Brukes for å klippe layeret over mot layeret grid for å få mindre og enklere polygoner.
fyll_2008_test fyll_2008_ruter_test	Brukes for å lage data for å fylle ut tilgjengeligjord_dmk_test der det ikke er jordbruksareal eller dyrkingsjord. Brukes for å klippe layeret over mot layeret grid for å få mindre og enklere polygoner.
fyll_2009_test fyll_2009_ruter_test	Brukes for å lage data for å fylle ut tilgjengeligjord_dmk_test der det ikke er jordbruksareal eller dyrkingsjord. Brukes for å klippe layeret over mot layeret grid for å få mindre og enklere polygoner.
fyll_2010_test fyll_2010_ruter_test	Brukes for å lage data for å fylle ut tilgjengeligjord_dmk_test der det ikke er jordbruksareal eller dyrkingsjord. Brukes for å klippe layeret over mot layeret grid for å få mindre og enklere polygoner.
overlapp	Brukes for å sjekke om det finnes overlapp i TilgjengeligJord-datasettene.

Tabell 14.7: Layer som brukes underveis i prosessen

to påfølgende øyeblikksbilder. De kan summeres opp for å få antall dekar fulldyrka som ikke er fulldyrka mer, eller antall dekar nydyrka jordbruksareal, men det å følge arealer over tid, og se hva som skjer med dem blir fortsatt tungt og krevende.

## 14.2.2 Problemer underveis

Funksjonen jeg laget for å slå de små polygonene sammen med et nabopolygon, er forholdsvis treg. Det blir også noen merkelige geometrier innimellom.

Jeg hadde ingen store problemer før årsversjon 2010 av AR5 skulle brukes til å lage «TilgjengeligJord\_2010\_test». I 2009 ble dataene for Hobøl periodisk ajourført av Skog og landskap. Mange grenselinjer har fått små forskyvninger, det er mange skalkepolygoner i datasettet. Alle forsøk jeg gjorde, endte med geometrifeil. Det er kjent at det kan hjelpe å gjøre en buffer(0) på problemgeometrier i PostGIS. Da jeg forsøkte det, mistet jeg data. Det oppdaget jeg både ved manuell inspeksjon og fordi arealet på testdatasettet ble for lite.

Hadde jeg brukt en lagringsmodell med topologi, kunne jeg brukt funksjoner tilsvarende «build» og «clean» i ArcInfo til å fjerne små polygoner og gap. Fordi jeg ikke har topologi, må jeg finne andre muligheter. ST\_SnapToGrid laget nye gap i datasettet, og måtte avskrives. Da kunne jeg enten oppløse alle polygoner, og jobbe på linjene (slå sammen punkter nærmere hverandre enn en viss avstand), eller jobbe



på enkelt-geometrier. Jeg forsøkte det første, men selv for små datamengder gikk det veldig sakte. Jeg kom bare så langt som til å node alle linjene etter at polygonene ble oppløst. Etter flere timer krasjet spørrigen med en uspesifisert feilmelding fra GEOS-biblioteket. (GEOS er et bibliotek for tilgang til Java Topology Suite via C++, og brukes av PostGIS for å gjøre operasjoner på geometri [54, 49]).

Derfor måtte jeg velge å jobbe på de enkelte polygon-geometriene. Jeg ser imidlertid at jeg ikke har klart å løse alle problemer med skalepolygoner og små gap. I tillegg ble det som sagt en del merkelige polygoner når skalkepolygonene ble slått sammen med nabopolygoner. Det var ikke alltid rutinen min fant den smarteste måten å gjøre sammenslåingene på.

Jeg antar en eller flere av problemstillingene gap, overlapp eller merkelige polygoner forårsaket både problemene når AR5-årsversjonen fra 2010 skulle tas i bruk, og problemene med å jobbe på linjene fra de oppløste geometriene.

### 14.2.3 Mulig forbedringer

I overlayet mellom siste versjon av TilgjengeligJord og siste versjon av AR5, kan man identifisere arealer som har vært jordbruksareal eller dyrkingsjord, men ikke er det lenger. Disse arealene kan man ta med i øyeblikksbildene, og inkludere en egenskap som angir at det en gang har vært jordbruksareal eller dyrkingsjord. En slik løsning kombinerer øyeblikksbilder for TilgjengeligJord med en løsning skissert i et internt notat ved Skog og landskap i 2005 [10] der det ble lagt opp til å ta var epå den høyeste verdien et polygon noen gang hadde hatt i jordbruks-sammenheng. Den skisserte løsningen ble ikke implementert, og den håndterte tidsdimensjonen dårlig. Kombinert med øyeblikksbilder kan den ha noe for seg.

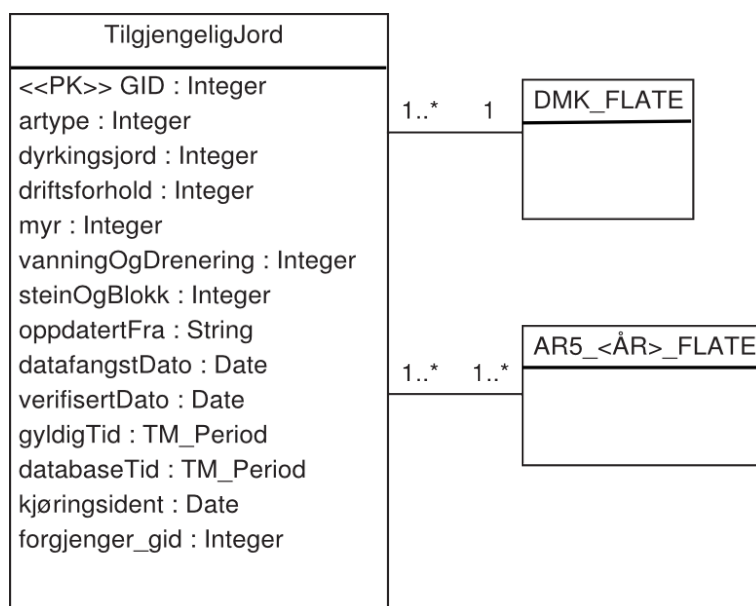
Et alternativ til å lage nye øyeblikksbilder, kunne være å legge dyrkingsjord inn i AR5 årversjonene. Små forskjeller i geometri måtte i intersect-operasjonen i så fall vært sett bort i fra, slik at AR5-geometriene ble beholdt der dyrkingsjord-polygonene var nesten like som AR5. Som vi har sett krever nok det mer logikk en det er hensiktsmessig å lage i enkle SQL-spørringer mot simple feature-data. I mangel av en topologimodell, anbefaler jeg å bruke et programmeringsspråk som for eksempel Java eller PostgreSQL sitt scriptspråk PL/pgSQL.

Romlige øyeblikksbilder kan implementeres som rasterdata. For å bevare kvaliteten i AR5, må en rasterimplementasjon ha liten cellestørrelse. Spørsmålet er om det da blir for tungt å håndtere.

## 14.3 TilgjengeligJord som rom-tid sammensetning

TilgjengeligJord-datasettet kan lages som rom-tid sammensetning. Det er en naturlig metode for å håndtere klassifiserte arealdekker på vektorform. Se figur 14.3 for data-modellen. Figur 14.4 viser et polygon med fulldyrka jord (artype 21) som splittes opp i to nye polygoner, ett polygon med fulldyrka og et annet med bebygd areal (artype 11).

Modellen er bitemporal, og den har tre typer datoer. Datafangstdato oversettes til gyldigTid av type TM\_Period (Se mer om standarden ISO 19118 i kapittel 8). I



Figur 14.3: Datamodell for TilgjengeligJord laget som rom-tid sammensetning

tillegg innføres en databasetid som også har type `TM_Period`. Hvis man endrer litt på bruken av verifiseringsdato i AR5, kan verifiseringsdato brukes som databasetid i form av «assertion time». Med databasetid menes da den perioden man mente at noe var gyldig. I tillegg har man et tidsstempel for når polygonet ble lagret. Der har jeg valgt å bruke egenskapsnavnet «kjøringsident» (implementeres som «kjøringsident»), som har lang tradisjon ved Skog og landskap.

Basisversjonen (null-versjonen) av `TilgjengeligJord` lages fra omkodet DMK i matjord\_0138\_flate på samme måte som for øyeblikksbilder. Det gir en null-versjon med ulik oppdateringsgrad for ulike områder.

### 14.3.1 Arbeidets gang

1. Første versjon av `TilgjengeligJord_rts_test` lages fra `matjord_0138_test`.
2. Områder som ikke er jordbruksareal eller dyrkingjord, fylles ut med `artype=99`, `dyrkingsjord=2`, `driftsforhold=0`, `myr=0`, `vannningOgDrenering=0` og `steinOgBlokk=0`.
3. `TilgjengeligJord_rts_overlay_test` etableres

For hver årversjon gjøres så følgende:

1. `TilgjengeligJord_rts_overlay_test` tømmes og fylles med data fra overlay mellom `TilgjengeligJord_rts_overlay_test` og `AR5_YYYY_test`.
2. `TilgjengeligJord_rts_overlay_test` brukes til å oppdatere `TilgjengeligJord_rts_test`. For fire ulike tilfeller kombinasjoner av egenskaper i `TilgjengeligJord_rts_overlay_test`, gjøres følgende:

Ikke lenger gyldige data i TilgjengeligJord\_rts\_test får nye datoer for gyldig-tid\_til og databasetid\_til ut i fra datafangstdato og verifiseringsdato i AR5-årsversjonen. Deretter settes de nye polygonene fra TilgjengeligJord\_rts\_overlay\_test inn over de med utgåtte datoer i TilgjengeligJord\_rts\_test. Alle nye polygoner får gid-en til forgjengeren i egenskapen «forgjenger\_gid». For et utgått polygon kan endringene kun være i egenskapene (ett polygon erstattes med ett nytt med endrede egenskaper), eller oppsplitting i flere polygoner med ulike egenskaper der noen har endrede egenskaper, andre kanskje ikke.)

De fire tilfellene dette gjøres for:

- a) dyrkingjord/ukjent dyrkingsjord/fulldyrka jord som ikke lenger er dyrkingsjord eller ukjent fordi arealene er blitt bebygd, samferdsel, vann, is

Dette gjøres:

- ikke dyrkingsjord (dyrkingsjord=2)
- driftsforhold=0, myr=0, vanningOgDrenering=0 , steinOgBlokk=0
- ny arealtype
- myr=0

- b) arealer som har blitt fulldyrka

Dette gjøres:

- ikke dyrkingsjord (dyrkingsjord=2)
- driftsforhold som før hvis 1,2,3, ellers kode 4
- vanningOgDrenering og steinOgBlokk som de var før eller ukjent
- arealtype=21
- evt ny myr-kode

- c) fulldyrka områder som har endret arealtype, men ikke blitt bebygd, samferdsel, vann, is.

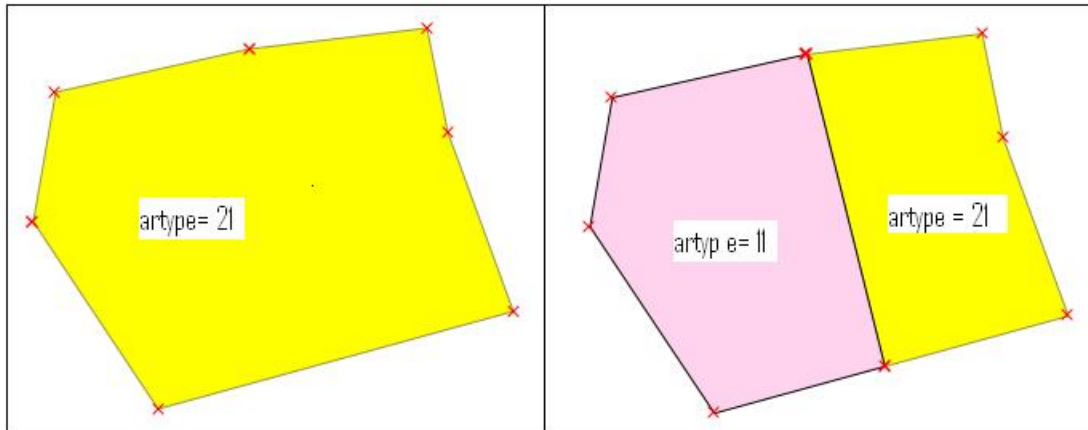
Dette gjøres:

- skal være dyrkingsjord (dyrkingsjord=1)
- driftsforhold som før hvis 1,2,3 ellers kode 4
- vanningOgDrenering og steinOgBlokk som de var før eller ukjent
- ny arealtype
- evt ny myr-kode

- d) arealer som ikke kommer inn under de tre kategoriene over, og som har endret arealtype eller myr

Dette gjøres:

- dyrkingsjord som før
- driftsforhold som før
- vanningOgDrenering og steinOgBlokk som de var før eller ukjent



Figur 14.4: Oppsplitting av en polygon i rom-tid sammensetning

- oppdater arealtype hvis endret
- oppdater myr-kode hvis endret

Kode for arbeidet finnes i vedlegg 3.

### 14.3.2 Skranker som burde vært implementert

Jeg har ikke implementert skranker på dataene. Det er særlig to typer skranker som kunne vært aktuelle. Jeg antar da at lovlige kombinasjoner av markslagsegenskaper i DMK og AR5 er testet fra før. De to typene skranker er:

- Skranker på tiden for å sikre at ethvert areal har en og bare en verdi til enhver tid. Det tilsvarer å sjekke for gap og overlapp i tidsdimensjonen. Disse sjekkene er enkle å implementere ved tester på datoegenskapene i TilgjengeligJord\_rts\_overlay\_test. Å rette opp problemet, innebærer å rette AR5-originalen. Det er en stor fordel om mest mulig feil plukkes opp med en gang AR5 er ajourført, i stedet for når TilgjengeligJord lages. Det kan gjøres gjennom forbedringer av endringsanalysen. Jo tidligere feil oppdages, jo enklere å rette.
- Skranker på geometri, for å hindre gap overlapp mellom polygoner i TilgjengeligJord\_rts\_overlay\_test og AR5-årsversjon. Sjekkene er enkle å implementere, men det å løse problemet kan være vanskelig i en geometrimodell uten topologi. Det er stor sannsynlighet for at slike problemer skyldes PostGIS og geometri-problemer.

### 14.3.3 Fordeler, ulemper og problemstillinger

Denne metoden har flere fordeler: Den tar betydelig mindre plass enn årlige øyeblikksbilder, den har alle data i et og samme layer, så man slipper å forholde seg til flere tabeller når man skriver spørringer, og absolutt alle endringer er tatt vare på og kan

hentes ut ved enkle SQL-er. Det er også en fordel at man slipper å forholde seg til persistente, og tildels kunstige objektidentifikatorer (OID-er) som i objektmodellen.

Ulempen er at polygonene brytes ned i stadig mindre polygoner, slik at et og samme jorde til slutt kan være representert ved svært mange, og også veldig små, polygoner. For testdataene startet jeg med 278 polygoner, men da jeg var ferdig, hadde jeg over 12000 polygoner i det samme området. Det skyldes ikke minst den periodiske ajourføringen, der mange avgrensningslinjer justeres. Her en tabell som viser hvordan romtid sammensetningen TilgjengeligJord\_rts\_test utviklet seg etter hvert som arbeidet skred fram:

	0-versjon	Med AR5 2008	Med AR5 2009	Med AR5 20010
Antall polygoner i overlay-layer		6352	7651	12107
Antall polygoner i TilgjengeligJord_rts_test	278	3332	7651	12310
Antall polygoner mindre enn 0.2 dekar	8	2659	2713	10769

Det er mulig å lage regler for når et polygon er så lite at det skal slås sammen med et nabopolygon, men det krever kompliserte regler. Slike regler må lages både for å bestemme hvilke polygoner som enten er så små eller har en slik form (for eksempel svært lange og smale) at de skal slås sammen med et annet polygon, og det må lages regler for hvilke polygoner de skal slås sammen med. Det er ikke en elementær oppgave å lage disse reglene. Det ble tydelig for øyeblikksbilde-modellen.

### 14.3.4 Mulige forbedringer

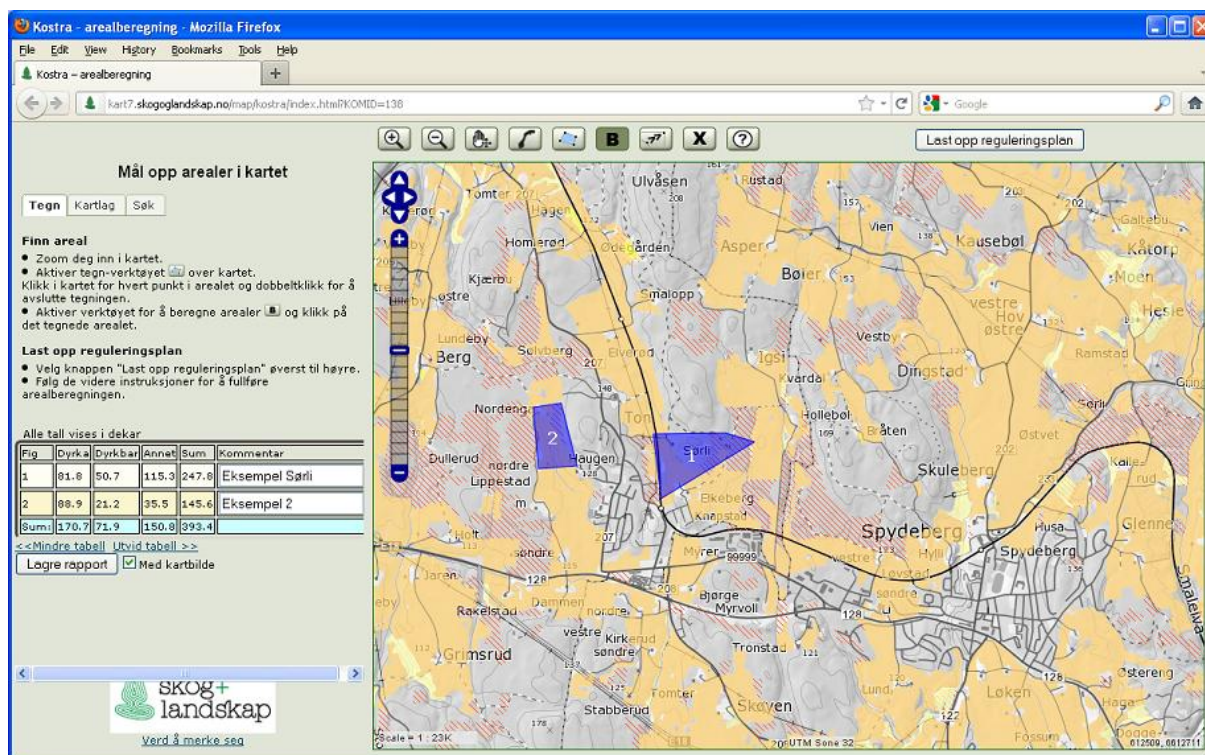
Problemet med alle de små polygnene kan gjøres mindre ved å lage en årsversjon med de mest oppdaterte dataene for TilgjengeligJord som et uttrekk hvert år. En slik årsversjon kan lages med en enkel SQL-spørring som trekker ut data fra TilgjengeligJord-datasettet:

```
CREATE table TilgjengeligJord_2008_Arsversjon as
SELECT *
FROM TilgjengeligJord_rts
WHERE to_date('31122008','ddmmyyyy') BETWEEN gyldigtid_fra AND gyldigtid_til
AND to_date('31122008','ddmmyyyy') BETWEEN dbtid_fra AND dbtid_til;
```

Dersom det er temporal overlapp eller gap, risikerer man å få et uttrekk som har overlappende polygoner eller ikke er heldekkende i det geografiske domenet.

Ved å lage slike årsversjoner, forsvinner fordelene med at dataene tar liten plass. Uansett er ikke plass et så stort problem som det var tidligere, og de datamengdene slike årsversjoner vil bestå av (jeg antar 7-8 GB hvert år), bør ikke by på problemer for Skog og landskap. Årsversjonene kan da brukes i applikasjoner som ikke trenger å håndtere endringer over tid, men kun har behov for de mest oppdaterte dataene. Et eksempel på en slik applikasjon, er Skog og landskaps Internett-løsning som skal hjelpe kommunene å finne ut hvor mye jordbruksareal og dyrkingsjord de planlegger å bygge ned, se figur 14.5. Løsningen er laget i samarbeid med Statens landbruksforvaltning som et hjelpemiddel for kommunene i Kostra-rapporteringen.

Et av brukstilfellene, handler om å finne all nydyrking på myr. Vi vet at kartleggingen av dyrkingsjord kan være mangelfull. Ergo kan vi få nydyrking på myr som ikke er dyrkingsjord. For å slippe å gjøre et overlay med en rekke årsversjoner av AR5 for å finne ut om nydyrka arealer er på slik myr, kan all myr fra DMK og AR5 tas med i TilgjengeligJord i utgangspunktet.



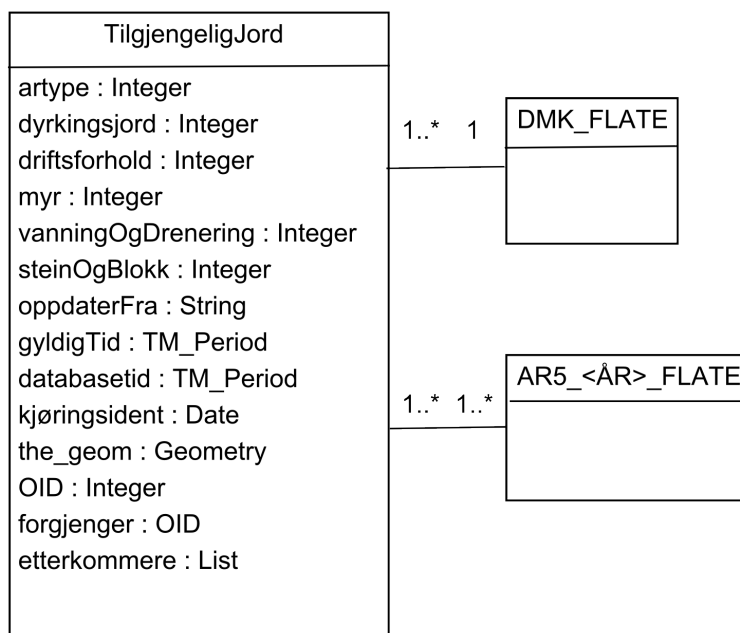
Figur 14.5: Applikasjon for hjelp ved Kostra-rapportering

For å mer effektivt kunne gå både forover og bakover i tid, kan man lage et slektstre for polygonene med pekere både oppover og nedover. Et slikt slektstre kan lages som en tabell i databasen Sørli med en kolonne for «mor» og en for «barn», der en mor kan ha flere barn, men et barn kun ha en mor. Som verdier i kolonnene kan GID-en til polygonene brukes.

## 14.4 TilgjengeligJord som objekter

Hvis man benytter objektmodellen, vil man se på hvert jordbruksareal og areal med dyrkingjord som en diskret enhet med skiftende egenskaper gjennom et livsløp. En objektidentifikator (OID) vil identifisere objektet gjennom hele livsløpet. Datamodellen vil bli den samme som for rom-tid sammensetning, men en persistent OID vil komme i stedet for løpenummeret GID. Primærnøkkel i layeret i rom-tid sammensetningen er en gid for hvert polygon. I objektmodellen, vil primærnøkkel i tabellen være OID sammen med dato for gyldig tid og databasetid. Hvorvidt man skal bruke både fra- og til-datoene, er spørsmål om hvilke skranke man har på dataene. Så sant det ikke tillates å ha overlappende tidsperioder for samme type tid, spiller det egentlig ingen rolle. Her er det naturlig å ha en slik skranke på dataene, så da velger vi den primærnøkkel med færrest attributter: OID, gyldigtid\_fra og databasetid\_fra.

Noen arealer i TilgjengeligJord er naturlige objekter, slik som et tydelig avgrenset jorde. Den store fordelen med denne objektmodellen, er at man bli i stand til å følge utviklingen til slike objekter gjennom den persistente OID-en. Man kan se om jorden



Figur 14.6: Datamodell for TilgjengeligJord som objektmodell

blir større eller mindre, og hva slags arealer som blir lagt til det eller skilt ut fra det.

Ulempen med modellen er at ikke alle arealer i datasettet er naturlige objekter, men vi må allikevel håndtere dem som det. Det betyr at vi må ha et regelverk som avgjør når endringer er så store at et objekt opphører å eksistere og blir erstattet av ny(e) objekt(er), hvilket objekt som skal fortsette å eksistere når to objekter slås sammen, og hvilket objekt som skal beholde OID-en når et objekt splittes opp. Slike regler er kompliserte å lage, særlig for elementer som ikke er naturlige objekter. Implementeringen av slike regler, blir fort enda mer komplisert enn det å lage dem.

Et alternativ er å alltid slette det opprinnelige objektet og lage nye når det skjer oppdateringer. Da får vi en modell som likner veldig på rom-tid sammensetning. Jeg ser ingen fordeler med å bruke en objektmodell på den måten framfor rom-tid sammensetning så lenge implementasjonen skjer i en relasjonsdatabase.

En annen mulighet, er å bruke objekter med skiftende egenskaper gjennom livsløpet bare der det er naturlig. Jordbruksareal er en god kandidat. Så kan man for dyrkingsjord i skog og på myr slette objektet og lage nye ved endringer, så man slipper alle de konseptuelt vanskelige valgene. Jeg anbefaler å vurdere muligheten av en slik modell. Her er noen momenter man kan følge:

- Ved oppsplitting av jordbruksareal når deler av arealet forblir jordbruksareal, lever objektet videre med det arealet som fortsatt er jordbruksareal. Da kan det hende arealet ikke er sammenhengende lenger,
- Hvis deler av arealet til et objekt med jordbruksareal endrer grunnforhold, lever objektet videre med det arealet som har samme myrkode som før. Da kan det hende arealet ikke er sammenhengende lenger.
- Hvis et objekt med jordbruksareal endrer arealtype til en annen kategori jord-

Eksempler på where-betingelser	
Where-betingelse	Hva svarer spørringen på?
dyrkingsjord = 1 or artype in(21,22,23)	areal jordbruksareal og dyrkingsjord
dyrkingsjord = 1	areal dyrkingsjord
artype in(21,22,23)	areal jordbruksareal
artype in(21,22)	areal fulldyrka og overflatedyrka jord
dyrkingsjord = 1 and artype in(22,23)	areal overflatedyrka jord og innmarksbeite som kan fulldyrkes
dyrkingsjord = 0 and artype in(22,23)	areal overflatedyrka jord og innmarksbeite som ikke kan fulldyrkes
dyrkingsjord = 1 and artype not in(22,23)	areal dyrkingsjord som ikke allerede er overflatedyrka eller innmarksbeite

Tabell 14.8: Eksempler på where-betingelsene i spørringer.

bruksareal, får objektet ny arealtype. Endrer myr-klassifikasjonen seg, forblir det også samme objekt, men får ny kode for myr.

- Hvis et objekt med jordbruksareal endrer arealtype til bebygd, samferdsel, vann eller snøisbre, får objektet ny arealtype. Objektet slettes ikke.
- Ved nydyrking av areal som grenser til jordbruksareal, slås det nyoppdyrkede arealet sammen med det eksisterende dersom arealtype og grunnforhold er det samme. Ellers representerer nydyrkingen et nytt objekt

Som for rom-tid sammensetning, kan all myr fra DMK og AR5 tas med selv om den ikke er dyrkingsjord. Da vil man lettere kunne besvare brukstilfellet «finn all nyrking på myr». En myr er ikke et naturlig objekt i informasjonssystemet, og kan håndteres som dyrkingsjord: Ved endring, ødelegges det gamle objektet og erstattes av nye.

## 14.5 De ulike brukstilfellene og øyeblikksbilder

I dette avsnittet går jeg gjennom de ulike modellen og ser om de kan svare på brukstilfellene. Alle brukstilfeller er hentet fra kapittel 13, og er i uthevet skrift.

### Hvor mye jordbruksareal/dyrkingsjord har vi?

Her trenger vi kun en enkel spørring mot et øyeblikksbilde for å få svaret:

```
SELECT sum(ST_Area(the_geom/1000)) as areal_dekar
FROM tilgjengeligJord_2009_test
WHERE dyrkingsjord = 1 or artype in(21,22,23)
```

Where-betingelsen kan enkelt varieres slik at man kun spør etter jordbruksareal eller enkelte klasser jordbruksareal, kun etter dyrkingsjord, kun etter dyrkingsjord som ikke allerede er overflatedyrka eller innmarksbeite også videre. Se figur 14.8 for eksempler.

### Finn all dyrkingsjord.

Dette kan også besvares med en enkel spørring:

```
SELECT *
FROM tilgjengeligJord_2009_test
WHERE dyrkingsjord = 1
```



## Hvor er jordbruksarealet og dyrkingsjorda?

For å fordele på kommuner, trenger vi bare å utvide spørringen over slik at den fordeler arealet på kommuner (i testdataene er ikke kommunenummer tatt med som egenskap, men det må gjøres i en nasjonal database):

```
SELECT komid, sum(ST_Area(the_geom))/1000 as areal_dekar
FROM tilgjengeligJord_2009_test
WHERE dyrkingsjord = 1 or artype in(21,22,23)
group by komid
```

Where-betingelsene kan selvfølgelig varieres på samme måte som over. Dersom en ikke ønsker å fordele på alle landets kommuner, men kun ønsker et utplukk, kan det enkelt legges inn i WHERE-delen av spørringen som en ekstra betingelse: «komid in(213,214)». Det er også enkelt å fordele på fylker i stedet for kommuner ved å gruppere på floor(komid/100):

```
SELECT komid, sum(ST_Area(the_geom/1000)) as areal_dekar
FROM tilgjengeligJord_2009_test
WHERE dyrkingsjord = 1 or artype in(21,22,23)
group by floor(komid/100)
```

Ønsker man andre geografiske fordelinger av arealet, må denne fordelingen finnes i form av et polygonkart med et polygon for hvert areal man ønsker dataene fordelt på. Deretter finner man arealfordelingen ved å gjøre en romlig snittoperasjon mellom de to datasettene. Det kan for eksempel gjøres ved å legge polygonkartet med «fordelingsarealene» inn i PostGIS, og kjøre en SQL-spørring med ST\_Intersection. I eksempelet under, ønsker vi dyrkingsjord fordelt på landskapsregionene i et layer kalt «Landskapsregioner».

```
SELECT l.landskapsregionID, sum(ST_Area(ST_Intersection(t.the_geom,l.the_geom) /1000)) as areal_dekar
FROM tilgjengeligJord_2009_test as t, landskapsregioner as l
WHERE dyrkingsjord = 1
and t.the_geom && l.the_geom
```

(Det er raskere å ikke bruke «ST\_Intersection» i where-betingelsen i spørringen over. Vi får med tomme polygoner ved å kun bruke AND-operatoren «&&» på de omsluttede rektanglene, men siden de ikke har areal, blir resultatet av spørringen allikevel riktig.)

**Hva har skjedd med det vi hadde av jordbruksareal og dyrkingsjord på et gitt tidspunkt. Er den nedbygd, nydyrket, fortsatt dyrket eller er det ingen vesentlig endring?**

**Finn alt tidligere jordbruksareal som er nedbygd.**

**Finn all attgrodd tidligere fulldyrka jord.**

**Finn all nydyrking på ikke dyrkingsjord.**

**Finn all nydyrking på tidligere skog eller myr.**

**Finn all dyrkingsjord som er nedbygd.**

Dette er spørsmål det ikke er så lett å svare på ut fra ett enkelt øyeblikksbilde. Har man tatt vare på layerne som ble laget som et romlig snitt mellom TilgjengeligJord og AR5-årsversjon underveis i prosessen med å lage øyeblikksbildene, kan endringene summeres opp fra disse.

En annen løsning er å gjøre en romlig spørring mellom de øyeblikksbildene som representerer henholdsvis starttidspunktet og sluttidspunktet man er interessert i for å identifisere endringer mellom dem. Da vil man finne jordbruksareal som er nedbygd også hvis det var en periode med skog i mellomtiden. Det kan også være aktuelt å gå tilbake til årsversjonene av AR5 for å se hva nydyrka arealer en gang var dersom man ikke har tatt med all myr i datasettet fra starten av.

For store arealer vil dette bli tidkrevende. Det må også forventes en del geometri-problemer underveis, slik at hele prosessen må antas bli både trøblete og ressurskrevende.

Det er grunn til å tro at de spørsmålene man oftest vil ønske svar på, handler om nedbygging. Det kan derfor være en løsning for mer effektivt å kunne svare på slike spørsmål, å lage en analyse over nedbygd areal mellom to øyeblikksbilder en gang i året, slik at denne kan gjenbrukes ved behov.

Den litt utvidede løsningen for øyeblikksbilder, der man tar vare på informasjon om at et areal en gang har vært jordbruksareal eller dyrkingsjord, vil gjøre det mulig å svare på spørsmål om hva som har skjedd med disse områdene ved hjelp av enkle SQL-spørringer.

### **Finn all tidligere dyrkingsjord som er nedgradert som et resultat av feilretting**

#### **Finn arealer som er omgradert ved feilretting**

Disse spørringene kan ikke besvares gjennom den valgte datamodellen for øyeblikksbilder.

Dersom feilretting var kodet som en egen type endring i et eget attributt, ville det vært lett å finne omfanget av feilrettinger også i øyeblikksbilder.

## **14.6 De ulike brukstilfellene og rom-tid sammensetning**

### **Hvor mye jordbruksareal/dyrkingsjord har vi?**

Her trenger vi kun en enkel spørring som inkluderer tidsaksene for å få svaret. Hvis vi har innført bitemporal tid, må alle spørringer inkludere begge tidsakser. Hvis datafangst dato brukes som gyldig tid, og dagens variant av verifiseringsdato brukes som database-tid, er ikke tidsaksene ortogonale, og det holder å bruke gyldig tid i spørringene. Da er verifiseringsdatoen kun å betrakte som en ekstra opplysning om at vi fortsatt mener at informasjonen er gyldig.

I eksemplene under, bruker jeg begge tidsakser i spørringene. Velger man å ikke innføre bitemporal tid, ser man bort i fra betingelsene på databasetid.

```
SELECT sum(ST_Area(the_geom))/1000 as areal_dekar
FROM tilgjengeligJord_rts_test
WHERE dyrkingsjord = 1 OR artype in(21,22,23)
AND gyldigtid_fra < now() AND gyldigtid_til > now()
AND dbtid_fra < now() AND dbtid_til > now()
```

### **Finn all dyrkingsjord.**

Dette er heller ingen avansert spørring;

```

SELECT *
FROM tilgjengeligJord_rts_test
WHERE dyrkingsjord = 1
  and gyldigtid_fra < now() AND gyldigtid_til > now()
  and dbtid_fra < now() AND dbtid_til > now()

```

Where-betingelsen kan selvfølgelig varieres for arealtype og dyrkingsjord på samme måte som da vi laget spørringer mot øyeblikksbilder. I tillegg har man mulighet til å variere betingelsene på de to tidsaksene, se eksempler i tabell 14.6. Spørringer mot en bitemporal tabell må ha betingelser som inkluderer begge tidsaksene.

Eksempler på where-betingelser på bitemporale tidsakser	
Where-betingelse	Hva svarer spørringen på?
gyldigtid_fra < now() and gyldigtid_til > now() dbtid_fra < now() and dbtid_til > now()	slik datasettet mener verden ser ut i dag slik det er registrert i databasen i dag
gyldigtid_fra < to_date('01012008','ddmmyyyy') and gyldigtid_til > to_date('01012008','ddmmyyyy')	slik datasettet menere verden så ut 1.1.2008
dbtid_fra < to_date('01012010','ddmmyyyy') and dbtid_til > to_date('01012010','ddmmyyyy')	slik det var registrert i databasen per 1.1.2010

Tabell 14.9: Eksempler på betingelser på tidsaksene i TilgjengeligJord som rom-tid sammensetning.

### Hvor er jordbruksarealet og dyrkingsjorda?

Dette gjøres som for øyeblikksbilder, men i tillegg må betingelsene inkludere de to tidsaksene. Det gjelder fordeling per kommune eller fylke, eller ved romlige spørringer mot andre fordelinger. Betingelsene på tidsaksene, kan selvfølgelig varieres som i tabellen over.

**Hva har skjedd med det vi hadde av jordbruksareal og dyrkingsjord på et gitt tidspunkt: Er den nedbygd, nydyrket, fortsatt dyrket eller er det ingen vesentlig endring?**

**Finn alt tidligere jordbruksareal som er nedbygd.**

**Finn all attgrodd tidligere fulldyrka jord.**

**Finn all nydyrking på ikke dyrkingsjord.**

**Finn all nydyrking på tidligere skog eller myr.**

**Finn all dyrkingsjord som er nedbygd.**

For den datamodellen jeg har valgt, er det greit å gå bakover i tid. Å finne skog som før var fulldyrka jord, eller å finne nedbygde arealer som tidligere var fulldyrka, er greit. For sistnevnte tilfelle, finner man da alle samferdselsarealer og bebygde arealer, og følger historien bakover gjennom egenskapen «forgjenger\_gid». I prinsippet går dette an for alle brukstilfellene ovenfor, men flere av dem ville være enklere å svare på dersom man hadde et slektstre i tillegg, slik at man kunne starte med de områdene man er interessert i å se hva har hendt med. Det første brukstilfellet er et typisk eksempel.

Jeg anbefaler scripting i PostgreSQL sitt scriptspråk PL/pgSQL eller at annet script- eller programmeringsspråk framfor bruk av SQL for å svare på disse brukstilfellene.

**Finn all tidligere dyrkingsjord som er nedgradert som et resultat av feilretting**

**Finn arealer som er omgradert ved feilretting**

Feilretting finner vi, dersom bitemporal tid er innført, ved å finne arealer der man har flere databasetid-perioder innenfor samme gyldighetid-periode. Det er da enkelt å finne slike arealer der dyrkingsjord-egenskapen har gått fra 0 til 1 (nedgradering). Det er også enkelt å finne arealene der en eller annen markslagsegenskap har endret seg (omgradert). Enda enklere hadde det vært å finne feilrettinger dersom det var kodet som en egen type endring.

## 14.7 De ulike brukstilfellene og objektmodellen

For en bitemporal objektmodell vil håndtering av brukstilfellene likne veldig på romtid sammensetning. Gitt at objektene er implementert som objekter i en objektdata-base, eller brukerdefinerte datatyper i en relasjonsdatabase, og har assosiasjoner til forgjenger og en liste med etterkommere, skulle alle brukstilfellene la seg besvare. Enda enklere blir det å identifisere feilretting hvis en egenskap med type endring er knyttet til hver nye forekomst av et objekt.

## 14.8 API og views

Det er mer komplekst å jobbe med bitemporale data enn med data uten tidshåndtering eller øyeblikksbilder. For å skåne brukeren, foreslår jeg å gjemme kompleksiteten bak et «application programming interface» (API). Gjør noen en sletting, endring eller innsetting av data, slår API-et øyeblikkelig til og sørger for at data håndteres riktig. Da blir det ikke mulig for eksempel å glemme seg og slette data for godt gjennom en ordinær «DELETE»-operasjon. Boka [20] gir gode oppskrifter på hvordan dette kan gjøres for bitemporale data ved hjelp av triggere.

For mindre avanserte brukere, kan det være lurt å tilrettelegge views som skjuler kompleksiteten i dataene når man skriver spørringer mot dem. Det er oftest de nyeste dataene man er ute etter, det er derfor å anbefale å lage views som henter ut dette, uten at det er nødvendig å angi betingelser på tidsaksene. Andre views bør også tilrettelegges ved behov.

# 15 Konklusjoner

Jeg ønsket å lage en systemarkitektur for et informasjonssystem egnet til å følge med på utviklingen av jordbruksarealer og dyrkingsjord i Norge. For å få til det, har jeg lest aktuell litteratur, jeg har undersøkt om datakildene som er tilgjengelige er gode nok, og jeg har vurdert tre modeller for en databaseløsning. To av modellene er testet ut i praksis, den siste kun diskutert teoretisk. Alle modellene gir bedre og mer oppdatert informasjon om tilgjengelig jordbruksjord enn vi har i dag.

Gjennom litteraturstudiene fant jeg fram til en rekke mulige modeller for å lagre tid, for å lagre geometrier, og for å lagre tid og geometri i samme løsning. Det er nødvendig her, fordi vi ønsker å følge fenomener med geografisk utstrekning over tid.

Bitemporal tid står fram som den absolutt beste metoden for å håndtere historikk, da den også gjør det mulig å håndtere feilretting på en dokumenterbar og konsistent måte. Ved å implementere databasetid som «assertion time», får vi en modell som lar oss registrere databasetider som ikke sammenfaller med transaksjonsdato. For AR5 og TilgjengeligJord vil registreringene normalt skje i ettetid. En bitemporal database krever bedre håndtering av dato-elementene i AR5 enn det som er tilfelle i dag.

Øyeblikksbilder er en enkel romlig-temporal modell som ikke kan være bitemporal. Velges øyeblikksbilder som modell for TilgjengeligJord, er det ikke mulig å svare på alle brukstilfellene. En løsning der man for hvert polygon tar vare på om dette arealet noen gang har vært dyrkingsjord eller jordbruksareal, og i så fall av hvilken kvalitet, vil øke nytten av datasettet, men også øke kompleksiteten betraktelig.

Jeg opplevde en del geometriproblemer på grunn av mitt forsøk på å slå skalkepolygoner sammen med nabopolygoner. Det viste seg også å være komplisert å få det til på en logisk måte, det ble mange polygoner med merkelig form. Alternativet er et datasett med svært mange polygoner. Det er mulig at PosGIS og simple feature-geomtrimodellen ikke var det ideelle valget av geodatabasesystem og geomtrimodell for TilgjengeligJord. En geomtrimodell med topologi, eller muligheter for «å snappe» et datasetts noder til et annet datasett som holdes fast, vil nok være nødvendig for å få effektive og konsistente løsninger. Begge deler kommer i neste versjon av PosGIS.

Rom-tid sammensetningen er kanskje mest tiltalende som romlig-temporal modell for TilgjengeligJord. Fordelene er enkel logikk og gode muligheter til å gå bakover i historien. Vedlikeholder man i tillegg et slektstre, blir det også enkelt å gå framover i tid. Ulempen med rom-tid sammensetning er at antall polygoner etterhvert blir veldig stort, og mange polygoner veldig små. Det ville være en fordel om man fant en metode for å eliminere små polygoner underveis, samtidig som man beholdt full dekning i rom og tid. Det krever gjennomtenkte måter å slå sammen polygoner på. Da jeg arbeidet med øyeblikksbilde-modellen, så jeg at slik sammenslåing ikke var helt elementært.

Objektmodellen er også tiltalende fordi den er en bedre avbildning av virkeligheten enn de to andre. Den krever logiske avklaringer som kan være kompliserte for arealdekke-data.

Gitt at Skog og landskap bestemmer seg for å ta jobben med å heve kvaliteten på dato-egenskapene i AR5, anbefaler jeg å bruke rom-tid sammensetning, gjerne med eliminering av små polygoner og årsversjoner med de mest aktuelle dataene, eller en objektmodell der det kun er jordbruksarealene som håndteres som ekte, persistente objekter. For andre arealer bør objektene ødelegges og erstattes med nye ved enhver endring.

Uansett rom-tid sammensetning eller objektmodell, er det viktig med god kvalitets-sikring av AR5 som en del av ajourføringen. Kontroll av datoer må inngå i kvalitets-sikringen på en bedre måte enn i dag. Hvis datafangstdato og verifiseringsdato fra gammelt og nytt polygon blir en del av endringsanalysen, vil temporale gap og overlapp kunne lukes ut. Imidlertid må data sendes tilbake til den som ajourførte datene for retting. Dersom endringsanalysen kunne bli en del av den kontrollen hver enkelt tolker utførte, ville det bli lettere og raskere å fjerne slike feil. Jeg er usikker på om det er realistisk at kommunene selv gjennomfører en slik test etter ajourføring.

Øyeblikksbilder er ikke like avhengig av riktig og logisk bruk av datoer som rom-tid-modellen og objektmodellen, og bør velges dersom man ikke tror det er mulig å forbedre bruk av datoer i AR5. Dersom man velger å bruke øyeblikksbilder, foreslår jeg å legge dyrkingsjord inn i hver årsversjon av AR5 i stedet for å etablere enda en type øyeblikksbilde. Særlig dersom det i neste versjon av PostGIS blir mulig å holde AR5-grensene faste mens dyrkingsjord tilpasses ved hjelp av den nye topologimodellen eller «snappe»-funksjonalitet, er det en god løsning. Det bør også her vurderes om man skal ta vare på om et polygon noen gang har vært jordbruksareal eller dyrkingsjord. Det øker nytten, men kompliserer det å lage datasettet, og vil nok øke antall polygoner.

Velges en bitemporal modell i form av rom-tid sammensetning eller objektmodell, anbefaler jeg både at det lages et API for endringer og et et view som trekker ut den mest aktuelle versjonen av dataene.

Rom-tid sammensetning og objektmodell lar både tid og sted måles. Hvis en av disse modellene velges for TilgjengeligJord, har man lyktes i å lage en database med en datamodell som lar begge disse komponentene fra Sinton sitt rameverk måles. Da bør behovet for informasjon om utviklingen av jordbruksarealer og dyrkingsjord være dekket, ikke bare i dag, men også for mange år framover.

# 16 Videre arbeid

## 16.1 Arbeid for Skog og landskap for å realisere informasjonssystemet

For å kunne lage et godt bitemporalt informasjonssystem for å følge med på endringer i jordbruksareal og dyrkingsjord, har jeg følgende anbefalinger til Skog og landskap:

- Identifisere så mange ikke-reelle endringer som mulig når AR5 oppdateres, slik at disse kan fjernes fra endrings-rapportene når de ikke bør være der.
- Bruke verifiseringsdato som «assertion time» i AR5 ved å lage nye rutiner for feilretting.
- Innskjerpe at datafangstdata  $\leq$  verifiseringsdato i AR5.
- Ta inn verifiseringsdato og datafangstdata for nytt og gammelt objekt i endringsanalysen. Bruke det for å finne gap og overlapp i tid.
- Kvalitetssikre rutiner for å kjøre endringsanalyser. Det skjer forglemmelser i dag. Det er særlig en utfordring med de kommunene som ajourfører direkte i QMS.
- Om mulig: kvalitetssikre de datoer som er gale i årsversjonene i AR5 i dag.
- Sjekke om feilene med ikke-oppdaterede datoegenskaper og datoegenskaper som «reiser tilbake i tid» i AR5 kun forekommer i Hobøl, eller om fenomenet er like vanlig over hele landet.
- Finne en metode for å fjerne små gap og overlapp i DMK-databasen. Sjekke at det ikke forekommer gap og overlapp i AR5-databasen. Finnes det, må også AR5 rettes.
- Teste topologifunksjonalitet og ny «snappe»-funksjon i PostGIS 2.0 og se hva de kan tilføre.
- Undersøke om rasterformatet og operasjoner på rasterdata kan vær en enklere og mer effektiv metode for forvaltning av TilgjengeligJord. De rasterte datasettene må ha liten celledørrelse, da kvaliteten fra AR5 må ivaretas. PostGIS 2.0 kommer med rasterfunksjonalitet, og ArcGIS har en velfungerende rastermodul.
- Når datamengdene vokser, er det grunn til å se nærmere på hvordan romlig-temporale data kan indekseres.

- Vurdere om man trenger egenskaper på avgrensningslinjer i TilgjengeligJord. I så fall må det også inn i DMK, AR5 og årsversjoner av AR5.
- Designe database med nødvendige skranker, API og views.
- Avgjøre om alt areal som har vært fulldyrka skal klassifiseres som dyrkingsjord når den endrer arealtype til noe annet enn bebygd, samferdsel vann eller snøisbre, også dersom den fulldyrka jorda var tungbrukt.
- Det finnes andre datasett som kanskje kan brukes til å forbedre informasjonen om dyrkingsjord uten at feltarbeid er nødvendig. Det er i så fall et eget forskningsarbeide. Et interessant arbeide, er Åge Nyborgs (ansatt i Jordsmonnsaksjonen ved Skog og landskap) forsøk med tildeling av jordindeks i Skaun kommune. Riktig nok inngår markslagsinformasjon som grunnlagsdata i arbeidet hans, men ikke dyrkingsjordinformasjon. Det er grunn til å tro at områder som i Nyborgs arbeide klassifiseres med høy jordindeks er å betrakte som dyrkingsjord. Kombinert med klimainformasjon, informasjon om avstand til vei og andre oppdyrka arealer etc kan dyrkingsjord-informasjonen muligens forbedres. Dersom klimaendringer gir høyere sommertemperaturer, vil tilfanget av dyrkingsjord øke, og det blir nødvendig å oppdatere informasjonen om hvor vi har dyrkingsjord.
- Velge romlig-temporal modell for datasettet, og ta de konseptuelle avgjørelser som er nødvendig.

## 16.2 Generelt

Visning av romlig-temporale data er en generell problemstilling det med fordel kan sees mer på. Mye av dagens forskning på romlig-temporale modeller og metoder, handler om multimedia-applikasjoner. Visning av data er sentralt i multimedia-løsninger. For å maksimalisere kunnskapen vi kan trekke ut av et romlig-temporalt informasjonssystem, er visning også der helt sentralt. Jeg antar det kan gjenbrukes metoder og teknikker fra multimedia i kartløsninger.

Usikkerhet knyttet til endringer i arealdata, er også et interessant emne for videre undersøkelser. Kan vi benytte statistiske metoder for å si noe om hvor mye av endringene som er reelle?

Bruk av flytende mengder for å modellere uskarpe overganger i geometri, egenskaper og tid er et aktuelt tema. Det hadde vært interessant å se på hvordan flytende mengder kan innlemmes i en database for arealressursdata.

Arbeidet med en forent konseptuell modell for objektdata og felldata kan i framtiden bli nyttig for arealdekke-data som AR5 og TilgjengeligJord. Kanskje kan en forent modell hjelpe oss til å modellere virkeligheten i stedet for dataene?

Bruk av datoegenskaper ved ulike typer oppdatering av geoinformasjon er et mulig tema for videre arbeid. Hvordan kan for eksempel SOSI utvides til å håndtere «assertion time»?



Bruk av fjernmålingsteknikker for automatisk oppdatering av dataene er et spennende tema. Er det mulig å oppdatere AR5 eller dyrkingsjord-informasjonen gjennom automatiske rutiner?

# Referanseliste

- [1] Tamas Abraham og John F. Roddick. Survey of Spatio-Temporal Databases. *Geoinformatica*, 3(1):61–99, 1999.
- [2] James F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983.
- [3] Rolf Bekkhus, Inge Bjørdal, Knut Bjørkelo og Tom J. Kristiansen. *Veileder for administrativt ajourhold av digitalt markslagskart (DMK)*. Norsk institutt for jord- og skogkartlegging, 2003. NIJOS-dokument 1b/2003.
- [4] Inge Bjørdal. *Markslagsklassifikasjon i økonomisk kartverk: 2007-utgåva*. Norsk institutt for skog og landskap, 2007.
- [5] Inge Bjørdal og Knut Bjørkelo. *AR5 klassifikasjonssystem: Klassifikasjon av arealressurser*. Norsk institutt for skog og landskap, 2006. Håndbok fra Skog og landskap 01/2006.
- [6] Inge Bjørdal, Knut Bjørkelo, Frode Flemsæter, Svein Ola Moum og Yngve Rekdal. *Forslag til ny modell for detaljert arealressurskart-AR5*. Norsk institutt for jord- og skogkartlegging, 2005. NIJOS dokument 09/2005.
- [7] Inge Bjørdal, Knut Bjørkelo, Barbi Nilsen, Ingvild Nystuen, Geir-Harald Strand og Kari Thorvaldsen. *Kodeverk og symbolbruk i DMK og avleidde produkt*. Norsk institutt for jord- og skogkartlegging, 2004. NIJOS-dokument 36/2004.
- [8] Jan Terje Bjørke. *Datamodeller, algoritmer og datastrukturer for geografisk informasjonsbehandling*. Geoforum, 2005. URL=[http://www.geoforum.no/forskOgUtdan/kartografisk-litteratur/algoritmer\\_datastrukturer.pdf/at\\_download/file](http://www.geoforum.no/forskOgUtdan/kartografisk-litteratur/algoritmer_datastrukturer.pdf/at_download/file).
- [9] Jan Terje Bjørke. *Kartografisk kommunikasjon*. Geoforum, 2005. URL=[http://www.geoforum.no/forskOgUtdan/kartografisk-litteratur/kartografisk\\_kommunikasjon.pdf/at\\_download/file](http://www.geoforum.no/forskOgUtdan/kartografisk-litteratur/kartografisk_kommunikasjon.pdf/at_download/file).
- [10] Knut Bjørkelo. Database for dyrka og dyrkbar mark: Fastsetting av brukerbehov og funksjonalitet. Internt Skog og landskap-notat, 2005.
- [11] Eliseo Clementini og Paolino Di Felice. A Comparison of Methods for Representing Topological Relationships. *Information Sciences - Applications*, 3(3):149 – 178, 1995.
- [12] Thomas. J. Cova og Michael. F. Goodchild. Extending geographical representations to include fields of spatial objects. *International Journal of Geographical Information Science*, 16(6):509–532, 2002.

- [13] Ramez Elmasri og Shamkant B. Navathe. *Fundamentals of Database Systems*. Addison Wesley, tredje utgave, 2000. ISBN 0-201-43090-2.
- [14] Hector Garcia-Molina, Jeffrey D. Ullman og Jennifer Widom. *Database Systems: The Complete Book*. Prentice Hall, 2002. ISBN 0-13-031995-3.
- [15] Leticia Gómes, Bart Kuijpers og Alejandro Vaisman. A data model and query language for spatio-temporal decision support. *Geoinformatica*, 15:455–496, 2010.
- [16] Antonin Guttman. R-trees—a dynamic index structure for spatial searching. *ACM SIGMOND Intl. Symp. on the Management of Data*, 74:47–57, 1984.
- [17] Gerard W. Hazeu, Arnold K. Bregt, Allard J.W. de Wit og Jan G.P.W. Clevers. A Dutch multi-date land use database: Identification of real and methodological changes. *International Journal of Applied Earth Observation and Geoinformation*, 13:682–689, 2011.
- [18] International Organization for Standardization (ISO). *ISO 19123: Geographic information – Schema for coverage geometry and functions (ISO19123:2005)*, 2005.
- [19] C.S. Jensen, J. Clifford, R. Elmasri, S.K. Gadia, P. Hayes, S. Jajodia, C. Dyreson, F. Grandi, W. Käfer, N. Kline, N. Lorentzos, Y. Mitsopoulos, A. Montanari, D. Nonen, E. Peressi, B. Pernici, J.F. Roddick, N.L. Sarda, M. R. Scalas, A. Segev, R.T. Snodgrass, M.D. Soo, A. Tansel, P. Tiberio og G. Wiederhold. Consensus Glossary of Temporal Database Concepts. *Sigmod Record*, 23(1):52–64, March 1994.
- [20] T. Johnston og R. Weiss. *Managing Time in Relational Databases: How to Design, Update and Query Temporal Data*. Morgan-Kaufmann, 2010.
- [21] Jordregisterinstituttet. *Markslagsklassifikasjon i økonomisk kartverk*, 1980.
- [22] Rose Tshoganetso Khachana, Anne James og Rahat Iqbal. Relaxation of ACID properties in AuTrA, The adaptive user-defined transaction relaxing approach. *Future Generation Computer Systems*, 27:58–68, 27:27–58, 2010.
- [23] K. Kjenstad. On the integration of object-based models and field based models in GIS. *International Journal of Geographical Information Science*, 20(5):491–509, 2006.
- [24] Gail Langran. *Time in Geographic Information Systems*. Taylor & Francis, 1992. ISBN 0-7484-0059-1.
- [25] C. P. Lo og Albert K. W. Yeung. *Concepts and Techniques of Geographic Information Systems*. Prentice Hall, andre utgave, 2007. ISBN 0-13-149502-X.
- [26] Lov 1995-05-12 nr 23: Lov om jord (jordlova). <http://www.lovdatab.no/all/hl-19950512-023.html> besøkt 18.10.2011, 1995.

- [27] Norsk institutt for jord- og skogkartlegging. *Modell for Digitalt MarkslagsKart*, 1992. Rapport nr 2/92.
- [28] Open Geospatial Consortium(OGC). *Simple Feature Access, versjon 1.2.1*.
- [29] Nikos Pelekis, Babis Theodoulidis, Ioannis Kopanakis og Yannis Theodoridis. Literature Review of Spatio-Temporal Database Models. *The Knowledge Engineering Review*, 19(3):235–274, 2004.
- [30] Donna J. Peuquet. *Representations of Space and Time*. The Guilford Press, 2002. ISBN 1-57230-773-0.
- [31] W. Ren, S. Singh, M. Singh og Y.S. Zhu. State-of-the-art on spatio-temporal information-based video retrieval. *Pattern Recognition*, 42(2):267–282, 2009.
- [32] Philippe Rigaux, Michel Scholl og Agn es Voisard. *Spatial Databases: With Application to GIS*. Morgan Kaufmann Publishers, 2002. ISBN 1-55860-588-6.
- [33] Simonas  altenis and Christian S. Jensen. Indexing of now-relative spatio-bitemporal data. *The VLDB Journal*, 11:1–16, 2002.
- [34] David F. Sinton. The Inherent Structure of Information as a Constraint to Analysis: Mapped Thematic Data as a Case Study . I G. Dutton, redakt r, *Harvard Papers on Geographic Information Systems, Volume 7*, side 1–17. Reading MA: Addison Wesley, 1978.
- [35] Richard T. Snodgrass. *Developing Time-Oriented Database Applications in SQL*. Morgan-Kaufmann, 1999. ISBN 1-55860-436-7.
- [36] Standard Norge. *Norsk Standard NS-EN ISO 19208: Geografisk informasjon: Modell for   beskrive tidsaspekter (ISO19108:2002)*, 2005.
- [37] Stortingsmelding nr. 30 Norge digitalt - et felles fundament for verdiskaping, 2002-2003.
- [38] Yufei Tao og Dimitris Papadias. Time-Parameterized Queries in Spatio-Temporal Databases. I *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, SIGMOD '02, side 334–345, New York, NY, USA, 2002. ACM.
- [39] Nectaria Tryfona og Christian S. Jensen. Conceptual Data Modeling for Spatio-temporal Applications. *Geoinformatica*, 3(3):245–268, 1999.
- [40] Michael Vazirgiannis, Yannis Theodoridis og Timos Sellis. Spatio-temporal composition and indexing for large multimedia applications. *Multimedia Systems*, 6:284–298, 1998.
- [41] Michael Worboys og Matt Duckham. *GIS: A Computing Perspective (second edition)*. CRC Press, 2004. ISBN 0-415-28735-2.

- [42] Michael F. Worboys. A Unified Model for Spatial and Temporal Information. *The Computer Journal*, 37(1):26–34, 1994.
- [43] Kristen Øyen. *Kartlegginga av Noregs grønne gull*. Norsk institutt for jord- og skogkartlegging, 2006.
- [44] <http://argouml.tigris.org/> besøkt 04.10.2011.
- [45] <http://www.esri.com> besøkt 28.08.2011.
- [46] [http://webhelp.esri.com/arcgisserver/9.3/java/index.htm#geodatabases/an\\_overv315896832.htm](http://webhelp.esri.com/arcgisserver/9.3/java/index.htm#geodatabases/an_overv315896832.htm) besøkt 05.11.2011".
- [47] <http://edndoc.esri.com/arcscde/9.2/concepts/versioning/versioning.htm> besøkt 05.11.2011.
- [48] [http://www.statkart.no/filestore/Landdivisjonen\\_ny/Fagomrder/lGeovekst/Produktspesifikasjoner/FKB\\_\\_\\_-\\_/4-AR5-2011-01-01.pdf](http://www.statkart.no/filestore/Landdivisjonen_ny/Fagomrder/lGeovekst/Produktspesifikasjoner/FKB___-_/4-AR5-2011-01-01.pdf) besøkt 07.10.2011.
- [49] <http://trac.osgeo.org/geos/> besøkt 04.12.2011.
- [50] <http://www-01.ibm.com/software/data/informix/> besøkt 05.12.2011.
- [51] <http://www.nkgs.no/wip4/quadri-map-server/d.epl?cat=19417> besøkt 19.09.2011.
- [52] [http://en.wikipedia.org/wiki/Object-relational\\_database](http://en.wikipedia.org/wiki/Object-relational_database) besøkt 10.09.2011.
- [53] [http://download.oracle.com/docs/cd/B19306\\_01/server.102/b14200/toc.htm](http://download.oracle.com/docs/cd/B19306_01/server.102/b14200/toc.htm) besøkt 04.11.2011.
- [54] <http://postgis.refrations.net/documentation/manual-1.5/> besøkt 04.10.2011.
- [55] <http://postgis.refrations.net/documentation/manual-svn/> besøkt 21.11.2011.
- [56] <http://www.postgresql.org/about> besøkt 10.09.2011.
- [57] <http://www.postgresql.org/docs/8.4/interactive/datatype-datetime.html> besøkt 04.09.2011.
- [58] <https://www.slf.dep.no/no/miljo-og-okologisk/areal-og-jordvern/jordvern/Kommuner+under+press+for+%C3%A5+bygge+ned+matjorda.15989.cms> besøkt 20.10.2011.
- [59] <http://www.cs.arizona.edu/~rts/sql3.html> besøkt 18.10.2011.
- [60] <http://www.skogoglandskap.no/temaer/markslag> besøkt 24.09.2011.
- [61] [http://www.skogoglandskap.no/filearchive/presentasjon\\_metadata.pdf](http://www.skogoglandskap.no/filearchive/presentasjon_metadata.pdf) besøkt 27.10.2011.

- [62] [http://www.statkart.no/standard/sosi/html\\_33/dmk/dmk.htm](http://www.statkart.no/standard/sosi/html_33/dmk/dmk.htm)  
besøkt 31.08.2011.
- [63] [http://www.statkart.no/filestore/ny/sosi/SOSI\\_pdf/GenerelleTyper.pdf](http://www.statkart.no/filestore/ny/sosi/SOSI_pdf/GenerelleTyper.pdf) besøkt  
28.10.2011.
- [64] Samtale med Geir-Harald Strand (Skog og landskap), juli 2011.
- [65] Samtale med Vidar Svendsen (Skog og landskap), 31. oktober 2011.

# Figurer

1.1	Fra hver mann på sin øy til geografisk infrastruktur. Kilde: ukjent . . .	4
3.1	Terminologien er forvirrende . . . . .	10
3.2	Økonomisk kartverk for et område nær Ås sentrum . . . . .	11
4.1	Liten relasjonsdatabase med to tabeller . . . . .	16
8.1	UML-modell av klassen Geometry i ISO19125 . . . . .	37
8.2	Temporale objekter . . . . .	38
8.3	Temporale geometriske primitiver . . . . .	39
10.1	SOSI UML-modell for DMK . . . . .	45
10.2	DMK har en enkel datamodell i Skog og landskaps nasjonale database .	48
10.3	Datamodell for kodeverket . . . . .	50
10.4	Visning av dyrka jord (gulaktig farge) og dyrkbar jord (skravert) . . . .	51
10.5	SOSI UML-modell for AR5 i FKB . . . . .	53
10.6	Webapplikasjon for innsyn i detaljerte endringer . . . . .	58
10.7	Kart som viser endringer mellom ulike datasett for et område . . . . .	59
12.1	Små skalke-polygoner (slivers) . . . . .	67
12.2	Data i årsversjon 2008. Datoer i 1964 . . . . .	69
12.3	Data i årsversjon 2009. Datoer fortsatt i 1964 . . . . .	70
12.4	Data i årsversjon 2009. Datoer i 2006 . . . . .	72
12.5	Data i årsversjon 2010. Datoer i 1964 . . . . .	72
14.1	Konseptuell datamodell for TilgjengeligJord (uten tid). . . . .	81
14.2	Datamodell for TilgjengeligJord som øyeblikksbilder . . . . .	84
14.3	Datamodell for TilgjengeligJord laget som rom-tid sammensetning . . .	88
14.4	Oppsplitting av en polygon i rom-tid sammensetning . . . . .	90
14.5	Applikasjon for hjelp ved Kostra-rapportering . . . . .	92
14.6	Datamodell for TilgjengeligJord som objektmodell . . . . .	93

# Tabeller

7.1	Representasjon av geografiske data i varierte formater, utvidelse av Sintons eksempler, fra Langrans bok [24] . . . . .	29
9.1	SQL-datatypeer for tid . . . . .	41
10.1	Koder for egenskapen arealtilstand i DMK [7]. . . . .	46
10.2	Koder for egenskapen jordklassifikasjon i DMK [7]. . . . .	47
10.3	Klasser i temakart dyrkingsjord [7]. . . . .	49
10.4	Omkoding av tema dyrkingsjord. Utvidelse av tabell i [7]. . . . .	50
10.5	AR5 egenskaper [5]. . . . .	52
10.6	Verdier for egenskapstypen Arealtype i AR5. Hentet fra [5]. . . . .	55
10.7	Verdier for egenskapstypen Grunnforhold i AR5. Hentet fra [5]. . . . .	55
10.8	Layerer i PostGIS med uttrekk av eksisterende data for Hobøl kommune	61
10.9	Tilgjengelige datasett satt inn i Sintons rammeverk . . . . .	62
14.1	Koder for egenskap dyrkingsjord . . . . .	78
14.2	Koder for egenskapen driftsforhold . . . . .	78
14.3	Koder for egenskapen myr . . . . .	78
14.4	Koder for egenskapen vanningOgDrenering . . . . .	79
14.5	Koder for egenskapen steinOgBlokk . . . . .	79
14.6	Layeret med årsversjonene av TilgjengeligJord for testområdet på 9 km <sup>2</sup> .	85
14.7	Layer som brukes underveis i prosessen . . . . .	86
14.8	Eksempler på where-betingelsene i spørringer. . . . .	94
14.9	Eksempler på betingelser på tidsaksene i TilgjengeligJord som rom-tid sammensetning. . . . .	97



# Vedlegg 1

Kode for å lage testdata i PostGIS. Først tas data ut fra Oracle/PosGIS, deretter legges de inn i en kopibase i PostGIS.

Kode i PLSQL i Oracle for funksjonene som splitter opp (normaliserer) informasjonen om dyrkingsjord i separate egenskaper. Funksjonene var laget av Knut Bjørkelo og Mette Ljosland (tidligere ansatt ved Skog og landskap) fra før.

```
CREATE or replace FUNCTION markslag_jord_drift_fnc(kode NUMBER, atil NUMBER) RETURN NUMBER AS
jord_kode NUMBER(1);
BEGIN
/*Hvis atil = 21 og kode er 0,23 91, er det tungbrukt (dvs 3) */
IF(atil = 21) THEN
  IF(kode = 91) THEN
    RETURN 3;
  ELSIF(kode = 23) THEN
    RETURN 3;
  ELSIF(kode = 0) THEN
    RETURN 3;
  END IF;
END IF;
IF(kode < 31) THEN
  RETURN 0;
/*Hvis kode er mellom 31 og 53, er det lettbrukt (dvs 1)*/
ELSIF(kode < 54) THEN
  RETURN 1;
/*Hvis kode er mellom 61 og 83, er det mindre lettbrukt (dvs 3)*/
ELSIF (kode < 84) THEN
  RETURN 2;
ELSE RETURN 0;
END IF;
END;
```

```
CREATE or replace FUNCTION markslag_myr_fnc(kode NUMBER) RETURN NUMBER AS
jord_kode NUMBER(1);
BEGIN
/*Hvis kode er 23, er det myr (dvs 2)*/
IF(kode = 23) THEN
  RETURN 2;
/*Hvis kode ender på 4, er det myr (dvs 2)*/
ELSIF(substr(kode,2) = 4) THEN
  RETURN 2;
ELSIF(kode = 0) THEN
  RETURN 0;
ELSE RETURN 1;
END IF;
END;
```

```
CREATE or replace FUNCTION markslag_jord_drener_fnc(kode NUMBER) RETURN NUMBER AS
jord_kode NUMBER(1);
BEGIN
/*Hvis kode er 23, er den normal (dvs 1)*/
IF(kode = 23) THEN
  RETURN 1;
/*Hvis kode ender på 3 (bortsett fra 23), er den sjølvdrenerert (dvs 2)*/
ELSIF(substr(kode,2) = 3) THEN
  RETURN 2;
/*Hvis kode ender på 4, er den tørkesvak (dvs 3)*/
ELSIF(substr(kode,2) = 4) THEN
  RETURN 3;
ELSE RETURN 0;
END IF;
END;
```

```

CREATE or replace FUNCTION markslag_jord_stein_blokk_fnc(kode NUMBER) RETURN NUMBER AS
jord_kode NUMBER(1);
BEGIN
/*Hvis kode er mellom 41 og 44, er det blokkrik (dvs 2)*/
IF(kode < 45) THEN
IF(kode > 40) THEN
RETURN 2;
ELSIF(kode = 0) THEN
RETURN 0;
ELSE
RETURN 1;
END IF;
/*Hvis kode er mellom 51 og 53, er det svært blokkrikt (dvs 3)*/
ELSIF(kode < 54) THEN
IF(kode > 50) THEN
RETURN 3;
ELSE
RETURN 1;
END IF;
/*Hvis kode er mellom 71 og 74, er det blokkrik (dvs 2)*/
ELSIF(kode < 75) THEN
IF(kode > 70) THEN
RETURN 2;
ELSE
RETURN 1;
END IF;
/*Hvis kode er mellom 81 og 83, er det svært blokkrikt (dvs 3)*/
ELSIF(kode < 84) THEN
IF(kode > 80) THEN
RETURN 3;
ELSE
RETURN 1;
END IF;
ELSIF(kode = 0) THEN
RETURN 0;
ELSE RETURN 1;
END IF;
END;

```

De overstående funksjonene ble brukt for å lage et layer i ArcSDE med navn matjord\_flate med følgende kode:

Først lages og registreres viewet i ArcSDE med en kommando i et kommandolinjevindu:

```

sdelayer -o CREATE_view -T matjord_flate -T markslag_flate -c geo,sl_sdeid -u sde_markslag
-i nijos_sde -s frigg

```

Deretter oppdateres view-teksten med følgende SQL, for eksempel i SQL\*Plus:

```

CREATE OR REPLACE FORCE VIEW "SDE_MARKSLAG"."MATJORD_FLATE"
("DYRKINGSJORD", "ARTYPE", "DRIFTSFORHOLD", "MYR", "STEINOGBLOKKINNHOLD", "VANNOGDRENERINGSBEHOV",
"OPPDATERT_FRA", "VERIFISERT_DATO", "DATAFANGST_DATO",
"GYLDIGTID_FRA", "GYLDIGTID_TIL", "DATABASETID_FRA", "DATABASETID_TIL", "DMK_DATO",
"DMK_SLSDEID", "GEO") AS
SELECT
case when jord < 31 or jord >84 then 2
when jord between 31 and 84 and atil!=21 then 1
else 0 end,
atil,
markslag_jord_drift_fnc(jord, atil),
markslag_jord_myr_fnc(jord),
markslag_jord_stein_blokk_fnc(jord),
markslag_jord_drener_fnc(jord),
'DMK',null,null,
to_date(dato,'rrrrmdd'), to_date('31129999','ddmmyyyy'),to_date(dato,'rrrrmdd'),
to_date('31129999','ddmmyyyy'), to_date(dato,'rrrrmdd'),
sl_sdeid, geo

```

```
FROM markslag_flate
WHERE atil in(21,22,23) OR jord between 31 AND 84;
```

Deretter ble dette gjort i ArcMap:

Data for Hobøl ble klippet ut fra matjord\_flate ved hjelp av et kommunekart. På grunn av enkelte små glipper mellom ØK-kartblad i DMK-layeret i ArcSDE, ble dataene konvertert til ArcINFO-formatet cover. Operasjonene «build» og «clean» ble kjørt på coveret med en toleransegrense på 15 centimeter for å tette glippene. DMK er lagret i Euref89 geografiske koordinater, så data ble transformert til UTM32. Deretter ble dataene eksportert til en shape-fil med navn matjord\_0138\_test.shp.

Dataene ble lagt inn i et temporært layer i PostGIS, ved hjelp av shp2psql. Da får vi et multipolygon-layer:

```
shp2psql -s 25832 -I matjord_0138_test.shp master.matjord_0138_tmp>lastMatjord.txt
psql -U master -d postgis -f lastMatjord.txt
```

For å lage et layer uten multipolygoner, kjøres følgende kommandoer:

```
CREATE TABLE master.matjord_0138
(
  gid serial NOT NULL,
  dyrkingsjord smallint,
  artype smallint,
  driftsforhold smallint,
  myr smallint,
  steinogblokk smallint,
  vannogdrenering smallint,
  oppdatert_fra character varying(15),
  verifisert_dato date,
  datafangst_dato date,
  gyldigtid_fra date,
  gyldigtid_til date,
  databasetid_fra date,
  databasetid_til date,
  dm_k_dato date,
  dm_k_slsdei integer,
  komid smallint,
  objekttype character varying(16),
  the_geom geometry,
)
WITH (
  OIDS=FALSE
);
ALTER TABLE master.matjord_0138 OWNER TO master;

SELECT AddGeometryColumn('master','matjord_0138','the_geom','25832','POLYGON',2);

CREATE INDEX matjord_0138_the_geom_gist
ON master.matjord_0138
USING gist
(the_geom)
```

Data i layeret ble lagt inn fra matjord\_0138\_tmp. For å splitte opp multipolygoner i matjord\_0138\_tmp, gjøres en ST\_Dump() på geometrikolonnen.

```
INSERT INTO matjord_0138
(dyrkingsjord , artype , driftsforhold, myr, steinogblokk, vannogdrenering,
 oppdatert_fra, verifisert_dato, datafangst_dato,
 gyldigtid_fra,gyldigtid_til, databasetid_fra ,databasetid_til,
 dm_k_dato, dm_k_slsdei, objekttype, the_geom)
SELECT
  dyrkingsjo, artype, driftsforh , myr, steinogblo, vannogdren,
  oppdatert_ , null, null,
```

```
gyldigtid_ , gyldigtid1, databaseti, database_1 ,
dmk_dato,dmk_slsdei, objekttype,(ST_Dump(the_geom)).geom
FROM matjord_0138_tmp;
```

Oppdaterer artype til å være konform med artyper i AR5. Endrer på myrkodene (de er omvendt av hva jeg ønsker).

```
UPDATE matjord_0138 set artype = 60 WHERE artype in(11,12,13);
UPDATE matjord_0138 set artype = 30 WHERE artype in(24,25,26);
UPDATE matjord_0138 set artype = 50 WHERE artype in(27);
```

```
UPDATE matjord_0138 set myr = 3 WHERE myr=2;
UPDATE matjord_0138 set myr = 2 WHERE myr=1;
UPDATE matjord_0138 set myr = 1 WHERE myr=3;
```

Data for AR5 og AR5 årsversjon 2008 og 2009 ble også eksportert fra ArcSDE ved hjelp av ArcMap. For AR5 ble data for Hobøl klippet ut, årsversjonene er allerede klippet mot kommunegrenser og har kommunenummer som attributt, så kun en enkel spørring med komid=138 var nødvendig for å eksportere Hobøl. Deretter ble data eksportert til shape-filene ar5\_0138.shp, 2008\_utm32.shp og AR5\_2009\_utm32 før de ble lagt inn i PostGIS:

```
shp2pgsql -s 25832 -I -S AR5_2008_utm32.shp master.ar5_2008_0138>lastAR5_2008.txt
shp2pgsql -s 25832 -I -S AR5_2009_utm32.shp master.ar5_2009_0138>lastAR5_2009.txt
shp2pgsql -s 25832 -I -S ar5_0138.shp master.ar5_0138>lastAR5.txt
psql -U master -d postgis -f lastAR5_2008.txt
psql -U master -d postgis -f lastAR5_2009.txt
psql -U master -d postgis -f lastAR5.txt
```

Data for AR5-årsversjon 2010 ligger originalt i PostGIS ved Skog og landskp. Data for Hobøl ble eksportert til en shapefil ved hjelp av pgsq2shp. Kommunenummer er et attributt i layeret, så kun en enkel spørring (komid=138) var nødvendig for å eksportere data for Hobøl kommune til shape-fil ar5\_2010\_utm32.shp.

```
shp2pgsql -s 25832 -I -S ar5_2010_utm32.shp master.ar5_2010_0138>lastAR5_2010.txt
psql -U master -d postgis -f lastAR5_2010.txt
```

## Vedlegg 2

Kode for å lage 0-versjon av TilgjengeligJord med testområdet og årlig oppdatering

Del 1 inneholder Spørringeri SQL og PL/pgSQL for PostGIS/PostgreSQL for å etablere 0-versjonen av øyeblikksbilder av TilgjengeligJord fra DMK (matjord-uttrekket fra vedlegg 1).

Del 2 inneholder kode for årlig nytt øyeblikksbilde av layeret ved hjelp av årsversjon for Ar5

### Del 1

Lager basislayeret (0-versjonen) av tilgjengeligJord, basert kun på DMK. Gjør en union på polygoner med felles gresne og like artype/dyrkingsjord-egenskaper («dissolve» i Arc-terminologi).

```
CREATE TABLE master.tilgjengeligjord_dmk_test
(
  dyrkingsjord smallint,
  artype smallint,
  driftsforhold smallint,
  myr smallint,
  steinogblokk smallint,
  vannogdrenering smallint,
  gid serial NOT NULL,
  the_geom geometry,
  CONSTRAINT tj_dmk_test_pk PRIMARY KEY (gid),
  CONSTRAINT enforce_dims_the_geom CHECK (st_ndims(the_geom) = 2),
  CONSTRAINT enforce_geotype_the_geom CHECK (geometrytype(the_geom) = 'POLYGON'::text OR the_geom IS NULL),
  CONSTRAINT enforce_srid_the_geom CHECK (st_srid(the_geom) = 25832)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE master.tilgjengeligjord_dmk_test OWNER TO master;

alter table TilgjengeligJord_dmk_test add constraint tj_dmk_test_pk primary key(gid);

SELECT AddGeometryColumn('master','tilgjengeligjord_dmk_test','the_geom','25832','POLYGON',2);

INSERT into TilgjengeligJord_DMK_test(dyrkingsjord,artype,driftsforhold,myr,steinogblokk,
  vannogdrenering, the_geom)
SELECT dyrkingsjord, artype, driftsforhold, myr, steinogblokk,
  vannogdrenering, (ST_Dump(ST_Union(the_geom))).geom AS the_geom
FROM matjord_0138_test
group by dyrkingsjord, artype, driftsforhold, myr, steinogblokk, vannogdrenering;

CREATE INDEX TilgjengeligJord_DMK_test_the_geom_gist
ON master.TilgjengeligJord_DMK_test
USING gist
(the_geom);
```

Finner og fjerner evt små polygoner og slivers i datasettet. Er ingen i testdatasettet, men har ingen garanti for at det ikke kan være det i andre data. Polygoner uten areal slettes. Småpolygoner slås sammen med nabopolygoner med samme ARTYPE og dyrkingsjord-egenskaper.

```
DELETE FROM tilgjengeligjord_dmk_test WHERE ST_Area(the_geom)=0;
```

```

CREATE table smaa_dmk_test as SELECT * FROM tilgjengeligjord_dmk_test
WHERE ST_Area(the_geom)/ST_Perimeter(the_geom) <0.25;

/* Funksjon som slår slivers sammen med nabopolygon */
CREATE OR REPLACE FUNCTION sma_dmk_func(p VARCHAR) RETURNS varchar as $$
DECLARE
  smapoly CURSOR IS
  SELECT *
  FROM tilgjengeligjord_dmk_test
  WHERE ST_Area(the_geom)/ST_Perimeter(the_geom) <0.25;

  v_gid integer;
BEGIN
  FOR spoly IN smapoly LOOP
    SELECT gid
    INTO v_gid
    FROM tilgjengeligjord_dmk_test wt
    WHERE wt.artype=spoly.artype
    AND ST_Touches(wt.the_geom, spoly.the_geom)
    AND ST_Area(wt.the_geom)/ST_Perimeter(wt.the_geom) >=0.25
    AND NOT EXISTS (SELECT 1 FROM tilgjengeligjord_dmk_test wt2
                     WHERE wt2.artype=spoly.artype
                     AND ST_Touches(wt.the_geom, spoly.the_geom)
                     AND ST_Area(wt.the_geom)/ST_Perimeter(wt.the_geom) >=0.25
                     AND ST_Area(ST_Intersection(spoly.the_geom,wt2.the_geom))>
                           ST_Area(ST_Intersection(spoly.the_geom,wt2.the_geom))
                     AND ST_Area(wt.the_geom)<ST_Area(wt2.the_geom));

    UPDATE tilgjengeligjord_2008_test b set
    the_geom=(ST_Dump(ST_Union(spoly.the_geom, b.the_geom))).geom
    WHERE b.gid=v_gid ;

    DELETE FROM tilgjengeligjord_dmk_test WHERE gid=spoly.gid;

  END LOOP;
  return '1';
END;
$$LANGUAGE plpgsql;

SELECT sma_dmk_func()

```

Lager polygon for å fylle ut TilgjengeligJord-datasettet fullstendig. Greit å ha for å kunne kontrollere areal mot klippepolygonets areal.

```

CREATE table fyll_dmk_test as
SELECT (ST_Dump(ST_Difference(k.the_geom, (SELECT ST_Union(the_geom)
                                         FROM tilgjengeligjord_dmk_test))))).geom AS the_geom
FROM klippepol k

CREATE INDEX fyll_dmkthe_geom_gist
ON master.fyll_dmk_test
USING gist
(the_geom);

```

Legger på rutenett på 'fyllpolygonene' slik at det ikke skal bli for store og tunge. Prøvde først uten, men det skapte store problemer.

```

CREATE table fyll_dmk_ruter_test as
SELECT (ST_Dump(ST_Intersection(f.the_geom, r.the_geom))).geom AS the_geom
FROM fyll_dmk_test f, grid r
WHERE f.the_geom&&r.the_geom

```

Setter fyllpolygonene inn i tilgjengeligjord\_dmk

```

INSERT into tilgjengeligjord_dmk_test
(dyrkingsjord,artype,driftsforhold,myr,steinogblokk,vannogdrenering,the_geom)
SELECT null,null,null,null,null,null,(ST_Dump(the_geom)).geom
FROM fyll_dmk_ruter_test;

```

Sjekker at alle polygoner er enkle og gyldige

```

SELECT count(*) FROM tilgjengeligjord_dmk_test WHERE not isValid(the_geom);
0
SELECT count(*) FROM tilgjengeligjord_dmk_test WHERE not isSimple(the_geom);
0

```

Sjekker arealsum. TilgjengeligJord\_DMK skal ha samme areal som klippepol, dvs differansen skal være tilnærmet 0

```

SELECT (SELECT ST_Area(the_geom) FROM klippepol) -
(SELECT sum(ST_Area(the_geom)) FROM tilgjengeligjord_dmk_test)
0.000732421875

```

Sjekk av overlapp. Resultatet bør enten være ingenting eller geometrier med svært lite areal:

```

SELECT SUM(ST_Area(ST_Intersection(a.the_geom,b.the_geom)))
FROM tilgjengeligjord_dmk_test a
INNER JOIN tilgjengeligjord_dmk_test b
ON (a.the_geom && b.the_geom
AND ST_Relate(a.the_geom, b.the_geom,'2*****'))
WHERE a.gid<>b.gid

0.00341796875

```

## Del 2

Oppdaterer TilgjengeligJord\_DMK\_test med årsversjon 2008 av AR5. 2008 kan enkelt erstattes av 2009 og 2010. Pass på å også erstatte det datasettet som oppdateres. Lager tabell for TilgjengeligJord\_2008

```

CREATE table TilgjengeligJord_2008_test
(dyrkingsjord smallint,
artype smallint,
driftsforhold smallint,
myr smallint,
steinogblokk smallint ,
vannogdrenering smallint,
gid serial);

alter table TilgjengeligJord_2008_test add constraint tj_2008_test_pk primary key(gid);

SELECT AddGeometryColumn('master','tilgjengeligjord_2008_test','the_geom','25832','POLYGON',2);

CREATE INDEX TilgjengeligJord_2008_test_the_geom_gist
ON master.TilgjengeligJord_2008_test
USING gist
(the_geom);

```

Kjører overlay mellom tilgjengeligjord\_dmk\_test og ar5\_2008:

```

CREATE table tilgjengeligjord_work_test
  (tj_dyrkingsjord, tj_artype,tj_driftsforhold, tj_myr,tj_steinogblokk, tj_vannogdrenering,
  ar_artype, ar_grunnf, the_geom) as
SELECT b.dyrkingsjord, b.artype, b.driftsforhold, b.myr, b.steinogblokk, b.vannogdrenering,
  a.artype, a.argrunnf,
(ST_Dump(ST_Intersection(a.the_geom, b.the_geom))).geom AS the_geom
FROM ar5_2008_test a,tilgjengeligjord_dmk_test b
WHERE ST_Relate(a.the_geom,b.the_geom,'2*****')

```

Setter fulldyrka jord til dyrkingsjord for å få mindre glipper i datasettet.

```
UPDATE tilgjengeligjord_work_test set tj_dyrkingsjord=1 WHERE ar_artype=21;
```

Setter inn data ved filtrering fra arbeidslayer i TilgjengeligJord\_2008

```

INSERT into tilgjengeligjord_2008_test
(dyrkingsjord,artype,driftsforhold,myr,steinogblokk,vannogdrenering,the_geom)
SELECT
tj_dyrkingsjord, ar_artype, tj_driftsforhold,
case when ar_grunnf=45 then 1 when ar_grunnf in(98,99) then 0 else 2 end ar_myr,
  tj_steinogblokk,tj_vannogdrenering,
(ST_Dump(ST_Union(the_geom))).geom
FROM tilgjengeligjord_work_test
WHERE ar_artype in(21,22,23)
  or tj_dyrkingsjord =1
group by
tj_dyrkingsjord, ar_artype, tj_driftsforhold, ar_myr,tj_steinogblokk,tj_vannogdrenering

```

Sletter polygoner uten areal. Fjerner små polygoner ved oppdatering («slivers» og annet ræl)

```

DELETE FROM tilgjengeligjord_2008_test WHERE ST_Area(the_geom)=0;

CREATE table smaa_2008_test as SELECT * FROM tilgjengeligjord_2008_test
WHERE ST_Area(the_geom)/ST_Perimeter(the_geom) <0.25

```

Fjerner slivers ved å slå sammen dem sammen med nabopolygon med samme artype med lengst felles grense med sliver-polygonet. Rutinen fungerer, emn tar lang tid, og vil måtte optimaliseres i en nasjonal databaseløsning.

```

CREATE OR REPLACE FUNCTION sma_2008_func() RETURNS varchar as $$
DECLARE
  smapoly CURSOR IS
  SELECT *
  FROM tilgjengeligjord_2008_test
  WHERE ST_Area(the_geom)/ST_Perimeter(the_geom) <0.25;

  v_gid integer;
BEGIN

  FOR spoly IN smapoly LOOP
    SELECT gid
    INTO v_gid
    FROM tilgjengeligjord_2008_test wt
    WHERE wt.artype=spoly.artype
      AND ST_Touches(wt.the_geom, spoly.the_geom)
      AND ST_Area(wt.the_geom)/ST_Perimeter(wt.the_geom) >=0.25
      AND NOT EXISTS (SELECT 1 FROM tilgjengeligjord_2008_test wt2
        WHERE wt2.artype=spoly.artype
          AND ST_Touches(wt2.the_geom, spoly.the_geom)
          AND ST_Area(ST_Intersection(spoly.the_geom,wt2.the_geom))>
            ST_Area(ST_Intersection(spoly.the_geom,wt.the_geom)));

    IF v_gid is not null then

```



```

        UPDATE tilgjengeligjord_2008_test b set
        the_geom=(ST_Dump(ST_Union(spoly.the_geom, b.the_geom))).geom
        WHERE b.gid=v_gid ;

        DELETE FROM tilgjengeligjord_2008_test WHERE gid=spoly.gid;
    END IF;

    END LOOP;
    return '1';
END;
$$LANGUAGE plpgsql;

SELECT sma_2008_func()

```

Sjekker at polygoner er enkle og gyldige

```

SELECT count(*) FROM tilgjengeligjord_2008_test WHERE not isValid(the_geom);
SELECT count(*) FROM tilgjengeligjord_2008_test WHERE not isSimple(the_geom);

```

Fyller ut alt areal

```

CREATE table fyll_2008_test as
SELECT (ST_Dump(ST_Difference(k.the_geom, (SELECT ST_Union(the_geom)
FROM tilgjengeligjord_2008_test))))).geom AS the_geom
FROM klippepol k

CREATE INDEX fyll_2008the_geom_gist
ON master.fyll_2008_test
USING gist
(the_geom);

```

Legger på rutenett på 'fyllpolygonene' slik at det ikke skal bli for store og tunge

```

CREATE table fyll_2008_ruter_test as SELECT ST_Intersection(f.the_geom, r.the_geom) AS the_geom
FROM fyll_2008_test f, grid r
WHERE f.the_geom&&r.the_geom

```

Setter fyllpolygonene inn i TilgjengeligJord\_2008\_test

```

INSERT into tilgjengeligjord_2008_test
(dyrkingsjord,artype,driftsforhold,myr,steinogblokk,vannogdrenering,the_geom)
SELECT null,null,null,null,null,null,(ST_Dump(the_geom)).geom
FROM fyll_2008_ruter_test;

```

Sjekker at polygoner er enkle og gyldige

```

SELECT count(*) FROM tilgjengeligjord_2008_test WHERE not isValid(the_geom);
SELECT count(*) FROM tilgjengeligjord_2008_test WHERE not isSimple(the_geom);

```

Hvis feil kan mna prøve buffer(0). Vær OBS på at arealer kan forsvinne.

```

UPDATE tilgjengeligjord_2008_test set the_geom = ST_Buffer(the_geom,0) WHERE not isValid(the_geom);

```

Sjekker arealtallet og overlapp

```

SELECT (SELECT ST_Area(the_geom) FROM klippepol) -
(SELECT sum(ST_Area(the_geom)) FROM tilgjengeligjord_2008_test)

SELECT SUM(ST_Area(ST_Intersection(a.the_geom,b.the_geom)))
FROM tilgjengeligjord_2008_test a
INNER JOIN tilgjengeligjord_2008_test b
ON (a.the_geom && b.the_geom
AND ST_Relate(a.the_geom, b.the_geom,'2*****'))
WHERE a.gid<>b.gid
AND a.ARTYPE IS NOT NULL
AND b.ARTYPE IS NOT NULL;

```

Oppdaterer kolonner for dyrkingsjord og driftsforhold for nye arealer dyrkingsjord

```
UPDATE tilgjengeligjord_2008_test
set dyrkingsjord=1,
    driftsforhold=4
WHERE artype =21 and dyrkingsjord is null;
```

Oppdaterer kolonner for dyrkingsjord og driftsforhold for tidligere arealer dyrkingsjord

```
UPDATE tilgjengeligjord_2008_test
set dyrkingsjord=2,
    driftsforhold=0,
    myr=0,
    vannogdrenering=0,
    steinogblokk=0
WHERE artype in (11,12,70,80);
```

Omkoder dyrkingjord på fulldyrka igjen

```
UPDATE tilgjengeligjord_2008_test set dyrkingsjord=2 WHERE artype=21;
```

# Vedlegg 3

Kode for å lage romlig-temporal sammensetning av TilgjengeligJord-datasettet og årlig oppdatering

Lager layeret TilgjengeligJord\_rts\_test og legger inn data fra matjord\_0138\_test

```
CREATE table TilgjengeligJord_rts_test (  
gid serial primary key,  
artype smallint,  
dyrkingsjord smallint,  
driftsforhold smallint,  
myr smallint,  
vanningogdrenering smallint,  
steinogblokk smallint,  
oppdatertfra character varying(30),  
gyldigtid_fra date,  
gyldigtid_til date,  
dbtid_fra date,  
dbtid_til date,  
kjoringsident timestamp,  
forgjenger_gid smallint);  
  
ALTER TABLE master.TilgjengeligJord_rts_test OWNER TO master;  
SELECT AddGeometryColumn('master','tilgjengeligjord_rts_test','the_geom','25832','POLYGON',2);  
  
CREATE INDEX tj_rts_test_the_geom_gist  
ON master.TilgjengeligJord_rts_test  
USING gist  
(the_geom);  
  
INSERT into TilgjengeligJord_rts_test  
(artype,dyrkingsjord,driftsforhold,myr,vanningogdrenering,steinogblokk,  
oppdatertfra,gyldigtid_fra,gyldigtid_til,dbtid_fra,dbtid_til,  
kjoringsident, forgjenger_gid, the_geom)  
SELECT artype,dyrkingsjord,driftsforhold,myr, vannogdrenering,steinogblokk,  
oppdatert_Fra,gyldigtid_fra,gyldigtid_til,gyldigtid_fra,databasetid_til,  
now(), 0, the_geom  
FROM matjord_0138_test;
```

Lager polygoner for å fylle ut datasettet (letter join-operasjoner). Klipper polygonene mot et grid så de ikke skal bli for tunge.

```
CREATE table fyll_rts_test as  
SELECT (ST_Dump(ST_Difference(k.the_geom, (SELECT ST_Union(the_geom)  
FROM tilgjengeligjord_rts_test))))).geom AS the_geom  
FROM klippepol k  
  
CREATE INDEX fyll_rts_the_geom_gist  
ON master.fyll_rts_test  
USING gist  
(the_geom);  
  
CREATE table fyll_rts_ruter_test as SELECT ST_Intersection(f.the_geom, r.the_geom) AS the_geom  
FROM fyll_rts_test f, grid r  
WHERE f.the_geom&&r.the_geom
```

Setter «fyllpolygonene» inn i TilgjengeligJord\_rts\_test

```
INSERT into tilgjengeligjord_rts_test (dyrkingsjord,artype,driftsforhold,myr,steinogblokk,vanningogdrenering,  
gyldigtid_fra,gyldigtid_til, dbtid_fra, dbtid_til,  
the_geom)  
SELECT 2,99,0,0,0,0,  
now(),to_date('31129999','ddmmyyyy'),now(),to_date('31129999','ddmmyyyy'),  
(ST_Dump(the_geom)).geom  
FROM fyll_rts_ruter_test;
```

Sjekker at alle polygoner er gyldige og enkle. Er det feil, gjør en buffer(0) på tilgjengeligjord\_rts\_test. Husk å sjekke før og etter. Det var feil ved join på årsversjon 2010. Buffer(0) fiksa problemet.

```
SELECT count(*) FROM tilgjengeligjord_rts_test WHERE not IsSimple(the_geom)
SELECT count(*) FROM tilgjengeligjord_rts_test WHERE not IsValid(the_geom)
```

Lager layer for overlay mellom tilgjengeligjord\_rts\_test og årsversjoner av AR5

```
CREATE table TilgjengeligJord_rts_overlay_Test (
gid serial primary key,
tj_gid smallint,
tj_artype smallint,
tj_dyrkingsjord smallint,
tj_driftsforhold smallint,
tj_myr smallint,
tj_vanningogdrenering smallint,
tj_steinogblokk smallint,
tj_oppdatertfra character varying(40),
tj_gyldigtid_fra date,
tj_gyldigtid_til date,
tj_dbtid_fra date,
tj_dbtid_til date,
tj_kjoringsident date,
tj_forgjengergid smallint,
ar_artype smallint,
ar_argrunnf smallint,
ar_verifiseringsdato date,
ar_datafangstdato date);

ALTER TABLE master.TilgjengeligJord_rts_overlay_test OWNER TO master;
SELECT AddGeometryColumn('master','tilgjengeligjord_rts_overlay_test','the_geom','25832','POLYGON',2);

CREATE INDEX tj_rts_ovealay_test_the_geom_gist
ON master.TilgjengeligJord_rts_overlay_test
USING gist
(the_geom);
```

Setter inn data. Dataene er ovelay mellom nåværende polygoner i TilgjengeligJord\_rts\_test og den årsversjonen av AR5 man jobber med. Herfra og ut gjentas altså det hele med nye årsversjoner.

```
INSERT into TilgjengeligJord_rts_overlay_test (tj_gid,tj_artype,
tj_dyrkingsjord,tj_driftsforhold,tj_myr,tj_vanningogdrenering, tj_steinogblokk,
tj_oppdatertfra,tj_gyldigtid_fra, tj_gyldigtid_til,
tj_dbtid_fra,tj_dbtid_til,tj_kjoringsident,tj_forgjengergid,ar_artype,ar_argrunnf,
ar_verifiseringsdato, ar_datafangstdato, the_geom)

SELECT
tj.gid,
tj.artype,
tj.dyrkingsjord ,
tj.driftsforhold ,
tj.myr ,
tj.vanningogdrenering,
tj.steinogblokk ,
tj.oppdatertfra,
tj.gyldigtid_fra,
tj.gyldigtid_til,
tj.dbtid_fra,
tj.dbtid_til,
tj.kjoringsident,
tj.forgjenger_gid,
a.artype,
a.argrunnf,
a.verifiseri,
a.datafangst,
(ST_Dump(ST_Intersection(tj.the_geom,a.the_geom))).geom
FROM
```

```
TilgjengeligJord_rts_test tj, ar5_2008_test a
WHERE tj.the_geom&&a.the_geom
and now() between tj.gyldigtid_fra and tj.gyldigtid_til
and now() between tj.dbtid_fra and tj.dbtid_til
and ST_Relate(a.the_geom, tj.the_geom,'2*****')
```

Fikser gale datoer. Får ikke sjekka gap og overlapp i datoer så enkelt:

```
UPDATE TilgjengeligJord_rts_overlay_test
set ar_verifiseringsdato= ar_datafangstdato
WHERE ar_verifiseringsdato < ar_datafangstdato;
```

```
UPDATE TilgjengeligJord_rts_overlay_test
set ar_verifiseringsdato= ar_datafangstdato
WHERE ar_verifiseringsdato is null;
```

```
UPDATE TilgjengeligJord_rts_overlay_test
set ar_datafangstdato =ar_verifiseringsdato
WHERE ar_datafangstdato is null;
```

Setter nye gyldigtid\_til og dbdato\_til til i TilgjengeligJord\_rts\_test der det har skjedd en av de fire definerte endringene.

Dyrkingjord/ukjent dyrkingsjord/fulldyrka jord som ikke lenger er det fordi arealene er blitt bebygd, samferdsel, vann, is:

```
UPDATE TilgjengeligJord_rts_test tj
set dbtid_til = a.veridato,
    gyldigtid_til = a.datadato
FROM (SELECT tj_gid, ar_verifiseringsdato as veridato, ar_datafangstdato datadato
      FROM TilgjengeligJord_rts_overlay_test
      (tj_dyrkingsjord in (0,1) OR ar_artype=21) AND ar_artype in(11,12,70,80)
      and ST_Area(the_geom)/ST_Perimeter(the_geom) >=0.25) as a
WHERE a.tj_gid = tj.gid;
```

Fulldyrka områder som har endret arealtype, men ikke blitt bebygd, samferdsel, vann, is, fulldyrka:

```
UPDATE TilgjengeligJord_rts_test tj
set dbtid_til = a.veridato,
    gyldigtid_til = a.datadato
FROM (SELECT tj_gid, ar_verifiseringsdato as veridato, ar_datafangstdato datadato
      FROM TilgjengeligJord_rts_overlay_test
      WHERE (tj_artype=21
            AND (ar_artype not in(21,11,12,70,80))
            and ST_Area(the_geom)/ST_Perimeter(the_geom) >=0.25) as a
WHERE a.tj_gid = tj.gid
```

arealer som har blitt fulldyrka

```
UPDATE TilgjengeligJord_rts_test tj
set dbtid_til = a.veridato,
    gyldigtid_til = a.datadato
FROM (SELECT tj_gid, ar_verifiseringsdato as veridato, ar_datafangstdato datadato
      FROM TilgjengeligJord_rts_overlay_test
      WHERE tj_artype != 21 AND ar_artype in(21))
      and ST_Area(the_geom)/ST_Perimeter(the_geom) >=0.25) as a
WHERE a.tj_gid = tj.gid
```

Arealer som ikke kommer inn under de tre kategoriene over som har endre arealtype eller myr

```

UPDATE TilgjengeligJord_rts_test tj
set dbtid_til = a.veridato,
  gyldigtid_til = a.datadato
FROM (SELECT tj_gid, ar_verifiseringsdato as veridato, ar_datafangstdato datadato
      FROM TilgjengeligJord_rts_overlay_test
      WHERE not (((tj_dyrkingsjord in (0,1) OR tj_artype=21) AND ar_artype IN(21,11,12,70,80))
      OR (tj_artype != 21 AND ar_artype = 21)
      OR (tj_artype=21 AND ar_artype in(11,12,70,80)))
      AND tj_artype!=ar_artype
AND ST_Area(the_geom)/ST_Perimeter(the_geom) >=0.25)as a
WHERE a.tj_gid = tj_gid

```

Sletter polygoner uten areal:

```
DELETE FROM TilgjengeligJord_rts_overlay_test WHERE ST_Area(the_geom)=0;
```

Setter inn de oppsplittede polygonene som skal erstatte de som har gått ute på dato. Dette gjøres for fire ulike tilfeller av kombinasjoner av egenskaper:

```

CREATE OR REPLACE FUNCTION omkod () RETURNS varchar as $$
DECLARE
  case1 CURSOR IS
  SELECT distinct tj_gid
  FROM TilgjengeligJord_rts_overlay_test
  WHERE (tj_dyrkingsjord in (0,1) OR ar_artype=21) AND ar_artype in(11,12,70,80)
  and ST_Area(the_geom)/ST_Perimeter(the_geom) >=0.25;

  case2 CURSOR IS
  SELECT distinct tj_gid
  FROM TilgjengeligJord_rts_overlay_test
  WHERE tj_artype != 21 AND ar_artype in(21)
  AND ST_Area(the_geom)/ST_Perimeter(the_geom) >=0.25;

  case3 CURSOR IS
  SELECT distinct tj_gid
  FROM TilgjengeligJord_rts_overlay_test
  WHERE tj_artype=21
  AND (ar_artype not in(21,11,12,70,80))
  AND ST_Area(the_geom)/ST_Perimeter(the_geom) >=0.25;

  case4 CURSOR IS
  SELECT distinct tj_gid
  FROM TilgjengeligJord_rts_overlay_test
  WHERE NOT (((tj_dyrkingsjord in (0,1) OR tj_artype=21) AND ar_artype IN(21,11,12,70,80))
  OR (tj_artype != 21 AND ar_artype = 21)
  OR (tj_artype=21 AND ar_artype in(11,12,70,80)))
  AND tj_artype!=ar_artype
  AND ST_Area(the_geom)/ST_Perimeter(the_geom) >=0.25;

BEGIN
  FOR rad IN case1 LOOP
    INSERT INTO TilgjengeligJord_rts_test
      (artype,dyrkingsjord,driftsforhold,myr,vanningogdrenering,steinogblokk,
      oppdatertfra,gyldigtid_fra,gyldigtid_til,
      dbtid_fra,dbtid_til,
      kjoringsident, forgjenger_gid, the_geom)
    SELECT ar_artype, 2,0,0,0,0,
      'AR5_2008',ar_datafangstdato, to_date('99991231','yyyymmdd'),
      ar_verifiseringsdato,to_date('99991231','yyyymmdd'),
      now(), tj_gid,the_geom
    FROM TilgjengeligJord_rts_overlay_test
    WHERE tj_gid=rad.tj_gid;
  END LOOP;
  FOR rad IN case2 LOOP
    INSERT INTO TilgjengeligJord_rts_test
      (artype,dyrkingsjord,driftsforhold,myr,vanningogdrenering,steinogblokk,
      oppdatertfra,gyldigtid_fra,gyldigtid_til,dbtid_fra,dbtid_til,

```

```

        kjouringsident, forgjenger_gid, the_geom)
SELECT ar_artype, 2,
       case when tj_driftsforhold=1 or tj_driftsforhold=2 or tj_driftsforhold=3 then tj_driftsforhold
           else 4 end,
       case when ar_argrunnf=45 then 1
           when ar_argrunnf=44 then 2
           else 0 end,
       coalesce(tj_vanningogdrenering,0),
       coalesce(tj_steinogblokk,0),
       'AR5_2008',ar_datafangstdato, to_date('99991231','yyyymmdd'),
       ar_verifiseringsdato,to_date('99991231','yyyymmdd'),
       now(), tj_gid,the_geom
FROM TilgjengeligJord_rts_overlay_test
WHERE tj_gid=rad.tj_gid;
END LOOP;

FOR rad IN case3 LOOP
INSERT INTO TilgjengeligJord_rts_test
(artype,dyrkingsjord,driftsforhold,myr,vanningogdrenering,steinogblokk,
oppdatertfra,gyldigtid_fra,gyldigtid_til,dbtid_fra,dbtid_til,
kjouringsident, forgjenger_gid, the_geom)
SELECT ar_artype, 1,
       case when tj_driftsforhold=1 or tj_driftsforhold=2 or tj_driftsforhold=3 then tj_driftsforhold
           else 4 end,
       case when ar_argrunnf=45 then 1
           when ar_argrunnf=44 then 2
           else 0 end,
       coalesce(tj_vanningogdrenering,0),
       coalesce(tj_steinogblokk,0),
       'AR5_2008',ar_datafangstdato, to_date('99991231','yyyymmdd'),
       ar_verifiseringsdato,to_date('99991231','yyyymmdd'),
       now(), tj_gid,the_geom
FROM TilgjengeligJord_rts_overlay_test
WHERE tj_gid=rad.tj_gid;
END LOOP;

FOR rad IN case4 LOOP
INSERT INTO TilgjengeligJord_rts_test
(artype,dyrkingsjord,driftsforhold,myr,vanningogdrenering,steinogblokk,
oppdatertfra,gyldigtid_fra,gyldigtid_til,dbtid_fra,dbtid_til,
kjouringsident, forgjenger_gid, the_geom)
SELECT ar_artype, tj_dyrkingsjord, tj_driftsforhold,
       case when ar_argrunnf=45 then 1
           when ar_argrunnf=44 then 2
           else 0 end,
       coalesce(tj_vanningogdrenering,0),
       coalesce(tj_steinogblokk,0),
       'AR5_2008',ar_datafangstdato, to_date('99991231','yyyymmdd'),
       ar_verifiseringsdato,to_date('99991231','yyyymmdd'),
       now(), tj_gid,the_geom
FROM TilgjengeligJord_rts_overlay_test
WHERE tj_gid=rad.tj_gid;
END LOOP;

return '1';
END;
$$LANGUAGE plpgsql;

SELECT omkod()

```