

ANVENDELSER AV GEOGRAFISKE DATABASER I BIM FOR INFRASTRUKTURPROSJEKTER

APPLICATIONS OF GEOGRAPHIC DATABASES IN BIM FOR INFRASTRUCTURE
PROJECTS

EIRIK OULIE ROSBACH

UNIVERSITETET FOR MILJØ- OG BIOVITENSKAP
INSTITUTT FOR MATEMATISKE REALFAG OG TEKNOLOGI
MASTEROPPGAVE 30 STP. 2013



Sammendrag

BIM, bygningsinformasjonsmodellering er i utstrakt grad benyttet i byggebransjen for bygg de siste årene og har bidratt til å redusere antall byggefeil og gitt bedre planlegging-, simulering- og visualiseringsmuligheter av bygget. Bygningsmodellene i 3D er tenkt å brukes gjennom hele byggets levetid, fra planleggings-, konstruksjons- og driftsfasen til bygget.

Høsten 2012 kom Statens Vegvesen *Håndbok 138 - "Modellgrunnlag. Krav til grunnlagsdata og modeller"*, som legger føringer og krav til modeller og grunnlagsdata i vei- og infrastrukturprosjekter som bruker digitale 3D modeller til prosjektering.

Denne oppgaven ser på hvordan en geografisk database som PostgreSQL med den romlige utvidelsen PostGIS som har støtte for 3D-geometri, topologi, koordinatsystemer og romlige operasjoner, kan brukes til å håndtere modell- og innmålingsdata i et infrastrukturprosjekt. Prosjektet *Bjørvika Etappe 2 - Dronning Eufemias gate* bruker slike modeller og ble valgt ut for å studere og avdekke hvordan modeller brukes i planlegging, til å lage stikningsdata og innmålinger av objekter som er bygget slik at geometrisk kontroll kan gjennomføres, og "som utført" dokumentasjon lages. Det er gjennomført intervjuer med de stikningsansvarlige og feltarbeid sommeren 2012 på prosjektet. Fra intervjuene og krav i Håndbok 138 er det laget kravspesifikasjoner for fagmodell, innmålingsdata og funksjonalitet til et system. Kravspesifikasjonene er brukt som grunnlag i arbeidet med å utvikle konseptuelle og logiske modeller før implementering i databasen. Programmer for å lese innmålingsdata og gjennomføre geometrisk kontroll er utviklet som en del av oppgaven.

Resultater fra prototypen viser at en geografisk database kan håndtere modell- og innmålingsdata i et infrastrukturprosjekt sammen slik at geometrisk kontroll kan gjennomføres. Objektstatus endres til "som utført" hvis objektet er bygget innenfor toleransegrensen eller så skrives det ut en avviksmelding. Eksisterende programvare kan brukes til å visualisere data i databasen. Prototypen har dessuten funksjonalitet for å beregne lengder og antall som kan brukes i planlegging- og dokumentasjonsfasen av et infrastrukturprosjekt.

Abstract

BIM, building information modelling, is used extensively in construction projects of buildings in the recent years with great success in reducing building faults, better planning, simulation and visualisation capabilities. 3D-models of the buildings are intended to be used throughout the building's life, from planning, construction and operation.

During autumn 2012 the Norwegian Public Roads Administration released *Guide 138 - "Basic models. Requirements for basic data and models"* (nor: Håndbok 138 - "Modellgrunnlag. Krav til grunnlagsdata og modeller"). Guide 138 provides guidelines and requirements for models and work process in road and infrastructure projects using digital models.

This paper look at the possibility a geographic database such as PostgreSQL with the spatial extension PostGIS which have support for 3D geometry, topology, coordinate systems and spatial operations, may be used to handle such models and surveyed data from the field in an infrastructure project. *Bjørsvika Stage 2 - Dronning Eufemias street* is one of several projects in Norway that are using such models today and are selected to identify use and stream of data from models, to stake out data in the field, surveying of built objects, geometric control and production of "as-built" documentation. Interviews with leaders of the surveyors and field work during summer 2012 have been conducted. From the interviews and based on requirements in Guide 138, requirement specifications for models, surveyed data and functionality to a system was developed. These specifications were used during developing conceptual and logical models ahead of implementation in the database. Programs to read surveyed data and to perform geometric control were developed as a part of this thesis.

Results from the prototype show that a geographic database is capable of handling 3D model data and surveyed data in an infrastructure project. These may be used together so that a geometric control may be conducted. The prototype do also have functionality which may be used during planning and documentation phases of similar projects.

Innledning

Denne masteroppgaven er avslutningen på min utdanning i Geomatikk ved Universitetet for miljø- og biovitenskap (UMB). Geografiske informasjonssystemer (GIS) har fascinert meg lenge under studietiden og i den siste tiden også bygningsinformasjonsmodellering (BIM).

Oppgaven er et resultat av sommerjobb i anleggsbransjen og utfordringene som er knyttet til bruk og håndtering av modell- (BIM) og innmålingsdata i anleggsprosjekter.

Jeg vil rette en takk til hovedveileder, Håvard Tveite ved UMB og medveileder Anders Avlesen ved Skanska, for god støtte og hjelp underveis. Jeg vil også takke alle på prosjektet i Dronning Eufemias gate som har bidratt til denne oppgaven, og spesielt Lars Petter Lingjærde og Bezim Zekaj.

Til slutt vil jeg takke familie, venner og studiekamerater for god støtte og hjelp gjennom studietiden ved UMB og for hjelp med gjennomlesing av oppgaven.

UMB, Ås, 7. mai 2013

Eirik Oulie Rosbach

Innhold

1	Oppgavens rammer & metoder	2
1.1	Problemstilling	2
1.2	Metode & oppbygging	2
1.3	Bakgrunn	3
1.4	Data & verktøy	4
1.5	Forkortelser & begreper	5
2	Introduksjon	7
2.1	GIS	7
2.2	BIM	7
2.3	Statens Vegvesen Håndbok 138	8
3	Teori	12
3.1	Geografiske data & geometri	12
3.2	Koordinatsystemer	14
3.3	Topologi	18
3.4	GIS-standarder	21
3.5	CAD- & BIM-standarder	26
3.6	Standarder for overføring av geografiske data	28
3.7	Datamodellering	29
3.8	Databasesystem	30
3.9	PostgreSQL & PostGIS	31
3.10	Tidligere arbeid	36
4	Studie av Dronning Eufemias gate	37
4.1	Planlegging	38
4.2	Produksjon	38
4.3	Dokumentasjon	40
4.4	Etterbruk & “som utført” modell	44
5	Prototype på system & resultater	46
5.1	Kravspesifikasjon	46
5.2	Konseptuelle & logiske modeller	49
5.3	Prototype på system	54
5.4	Resultater	55

6	Diskusjon & konklusjon	61
6.1	Diskusjon	61
6.2	Konklusjon	64
6.3	Videre arbeid	65
	Bibliografi	66
A	Intervju med entreprenør	71
B	Program	75
B.1	geo2pgsql	75
B.2	Geometrisk kontrollrapport	84

Figurer

2.1	Stikingsdata VA.	10
2.2	Gravegrop for konstruksjoner.	11
2.3	Innmålt geometri og “som utført” modell.	11
3.1	Vektormodellen for geometri med topologisk dimensjon og primitiv.	13
3.2	Regulær og irregulær representasjon i feltmodellen	13
3.3	Triangulering og TIN	14
3.4	Kartesisk høyrehåndssystem for lokale koordinater.	15
3.5	Koordinatsystem av type kilometrering.	15
3.6	Geodetisk lengde- og breddegrad og ellipsoidisk høyde.	16
3.7	Lokalt geodetisk koordinatsystem (ENU)	17
3.8	ECEF kartesisk koordinatsystem	17
3.9	Kartprojeksjon: kjegleprojeksjon og transversal sylinder.	18
3.10	Linjer med ulike topologiske egenskaper.	19
3.11	Spaghetti og topologisk modell.	20
3.12	Topologiske forhold mellom 3D objekter	20
3.13	Geometriprimitiver i GML 3.2.1.	21
3.14	5 “Level of Details” i CityGML	22
3.15	Kjernemodell i SOSI ledning.	24
3.16	SOSI ledning vann og avløp.	25
3.17	Oppbygging i ifc-formatet for å definere en vegg.	26
3.18	Skjermdump av Kartverkets karttjeneste Norgeskart.	28
3.19	UML spesialisering av klasse og forhold.	30
3.20	Databasesystem.	31
3.21	OpenGIS Simple Features.	32
3.22	Geometrityper i PostGIS.	33
3.23	3D-modell laget fra databasen.	36
4.1	Dataflyt under produksjon.	39
4.2	Revidering vist i modell.	41
4.3	Temkodeliste.	42
4.4	Betongkonstruksjon plate.	42
4.5	Oversikt over innmålinger i geometrisk rapport i arbeidstegning.	43
4.6	Utsnitt av loggskjema for geometrisk rapport.	44
4.7	Dataflyt dokumentasjon.	45
5.1	Konseptuell modell for vann og avløp.	51
5.2	Logisk modell for vann og avløp.	52

5.3	Konseptuell modell for innmåling.	53
5.4	Generell logisk modell for innmåling.	54
5.5	Prototypens oppbygging av ulike komponenter.	54
5.6	Oversiktsbilde over geometri i prototype.	56
5.7	Utsnitt fra fagmodell for va-ledning og innmålinger.	57
5.8	Utsnitt fra fagmodell for kummer og innmålinger.	58

Tabeller

2.1	Modelltyper i håndbok 138	9
-----	-------------------------------------	---

Kapittel 1

Oppgavens rammer & metoder

BIM, bygningsinformasjonsmodellering, er et begrep vi stadig omgås i faglitteratur og fagtidsskrifter, men også etterhvert i vanlige mediekanaler. BIM blir beskrevet som en avansert 3D-modell bestående av intelligente objekter som skal gjøre byggeprosessen billigere, mer effektiv og gi færre feil. I utgangspunktet er BIM modelldata for bygninger, men metoden er overført til infrastrukturprosjekter og kan fremdeles sies å være i startgropa.

I denne oppgaven vil jeg undersøke hvordan mulighetene i et databasesystem med støtte for 3D-geometri og geografisk data, et geografisk databasesystem, kan benyttes for modell- og innmålingsdata i infrastrukturprosjekter. En geografisk database hører i utgangspunktet til et geografisk informasjonssystem (GIS).

1.1 Problemstilling

Problemstillingen i denne oppgaven er konkretisert til følgende:

Hvordan kan et geografisk databasehåndteringssystem anvendes for modelldata (BIM), innmålingsdata og "as-built" dokumentasjon, samt være et verktøy for brukere under planlegging-, produksjon- og dokumentasjonsfasen i et infrastrukturprosjekt.

For å undersøke dette vil jeg se på et prosjekt i Skanska som bruker modelldata for å undersøke og avdekke behov for data, egenskaper og funksjonalitet som et system bygget på en geometrisk database må ha. Et studie av formater som brukes i GIS og BIM som brukes i dag eller kan brukes for modelldata inngår også.

1.2 Metode & oppbygging

I denne oppgaven blir prosjektet *Bjørvika Etappe 2 - Dronning Eufemias gate*, heretter kalt Dronning Eufemias gate, studert. Dette prosjektet bruker modelldata etter kravene i *Statens Vegvesens Håndbok 138 - Modellgrunnlag. Krav til grunnlagsdata og modeller* [1]. Denne kalles heretter Håndbok 138.

For å kunne svare på problemstillingen i denne oppgaven er metoden delt opp i fire deler:

- Studere et prosjekt gjennom intervjuer og feltarbeid for å undersøke dataflyt fra digitale modeller (BIM) til bruk i planlegging, som stikningsdata i produksjon og “som utført” dokumentasjon.
- Studere formater som brukes i BIM (modelldata) og GIS (geografiske databaser) for å gi en bakgrunn for formater som kan eller blir brukt for modelldata
- Utvikle en prototype på et system i databasen som kan håndtere modell- og innmålingsdata for en fagmodell i et infrastrukturprosjekt gjennom prosessen med datamodellering (konseptuelle og logiske datamodeller). Fagmodellen vann og avløp fra Håndbok 138 avgrenses i denne prototypen til kun kum og ledning.
- Utvikle et program som benytter data i databasen til å gjennomføre geometrisk kontroll og produsere “som utført” dokumentasjon.

GIS, BIM og Håndbok 138 behandles i kapittel 2 for å gi en introduksjon til sentrale temaer i oppgaven. Teori om geografiske data og geometri, koordinatystemer, topologi, GIS- og BIM-standarder, datamodellering og det geografiske databasesystemet som er valgt for prototypen behandles i kapittel 3. Tidligere arbeid inngår også her. Kapittel 4 er en gjennomgang av dataflyt og modellbruk i prosjektet Dronning Eufemias gate. Kapittel 5 tar for seg kravspesifikasjoner, konseptuelle og logiske datamodeller for fagmodell og innmålinger. Implementering i databasesystemet og resultater behandles også. Til slutt i kapittel 6 diskuteres resultatene opp mot problemstilling og konklusjoner trekkes på bakgrunn av arbeid og diskusjon. Muligheter for videre arbeid presenteres også.

1.3 Bakgrunn

Bygg- og anleggsbransjen har gjennom de to siste tiårene endret seg mye gjennom utvikling og standardisering av arbeidsprosesser, objekter, datamodeller og formater for utveksling av data. Bruk av 3D modeller for analyse og visualiseringsformål har blitt viktig i komplekse prosjekter, men også til bruk i navigering i bil og telefon med GPS og kart. Hovedsakelig har 3D-modeller vært brukt til visualisering og presentasjon, men det er nå et økende fokus på å bruke 3D-modeller som en integrert del av design og konstruksjonsprosessen [2, s. 2].

I mai 2007 gikk Statsbygg ut og sa at de skulle gå over fra ukoordinerte 2D-tegninger til 3D/4D/5D-informasjonsmodeller med intelligente informasjonsobjekter og at satsningen på BIM skal basere seg på den internasjonale industristandarden IFC [3]. Tilsvarende uttalelser om bruk av 3D informasjonsmodeller har både Statens Vegvesen (SVV) og Jernbaneverket (JBV) kommet med under et nettverkstreff i BA-Nettverket i 2012 [4, s. 13]. Når store aktører som Statsbygg, SVV og JBV velger å satse på dette får det betydninger for hele bransjen.

buildingSMART står bak den åpne standarden *Industry Foundation Classes* (IFC) for *Building Information Modelling* (BIM). Byggbransjen (bygg, blokk etc) har valgt å følge denne standarden og denne er blitt svært utbredt. Anleggsbransjen (vei, bane, vann og avløp etc) har ligget lenger etter og fulgte ISO/TC 211-arbeidet

og ISO1900-serien av standarder. I dag framstår bygg- og anleggsbransjen som to litt forskjellige verdener som jobber med det samme, informasjonsmodellering, men refereres til som henholdsvis BIM- og GIS-verdenen [5, vedlegg 5]. Dette har med valg av forskjellige standarder (ISO/TC211 er GIS-standarder), se kapittel 3.4 og 3.5. Store aktører i bransjen er pådrivere for å innføre BIM i infrastrukturprosjekter med grunnlag i Håndbok 138 [6]. Programvareleverandøren *Vianova Systems* er en av disse og som lager en programvare som håndterer modeller etter Håndbok 138, og bruker begrepet “BIM for infrastruktur” om denne GIS-verdenen og sin programpakke *Novapoint 19DCM* [7].

I dag er det store utfordringer å flytte data mellom disse verdenene fordi BIM-verdenen i utgangspunktet modellerer det som skal bygges gjennom detaljerte 3D-objekter med mye informasjon mens GIS-verdenen modellerer det som er bygget gjennom en forenkling av verden med enkle geometriske primitiver pluss informasjon ved siden av.

En tilleggsutfordring er at alle benytter IKT-løsninger som er spesialutviklet for eget formål, men ingen kan i særlig stor grad benytte data fra andre. Alle må produsere data på nytt som andre har laget tidligere. Modellene ble sett på som private for rådgivere og arkitekter og ble i liten eller ingen grad delt med entreprenøren [5, vedlegg 5, s. 19].

Anleggsbransjen og programvareleverandørene har gjennom *BA-Nettverket*, som ble etablert i 2003 av Inger Hokstad, gått sammen for å løse utfordringene med BIM- og GIS-verdenene og deling av modelldata. BA-Nettverket er en arena for aktører i bygg- og anleggsbransjen for spredning av kunnskap, avdekke hvordan bransjen kan utnytte ny teknologi, utvikle standarder, programvare og tilpasse seg styrende dokumenter. I felleskap gjennom BA-nettverket ble det utarbeidet en ønskeliste for hvordan bransjen kunne la data flyte mellom aktører og de ulike fasene i prosjektene. I 2007 tok Vegdirektoratet over arbeidet som var gjort med å lage en ny håndbok og resultatet var Håndbok 138 som ble publisert oktober 2012 [1][4][8].

Det er skrevet flere oppgaver om BIM de siste årene med temaer som økonomiske fordeler, bedre muligheter for planlegging, simulering og visualisering. Det er skrevet lite eller ingenting om lagring og versjonshåndtering av modeller på objektnivå, og lite eller ingenting for infrastruktur- og samferdselsprosjekter. Gjennom sommerjobb og samtaler med ansatte hos entreprenøren Skanska er det konstantert utfordringer knyttet til dette og dette danner utgangspunktet for oppgaven. Det er naturlig å trekke inn Håndbok 138 som setter standard for hvordan modeller og arbeidsprosesser skal foregå i anleggsbransjen og studere utfordringer og muligheter som arbeid etter denne håndboka skaper.

1.4 Data & verktøy

Et fokus gjennom hele oppgaven er bruk av åpen kildekode (eng: open source) og gratis programvare for å løse de enkelte oppgavene i metoden. Data som er benyttet i oppgaven er .dwg-filer for fagområdet vann og avløp fra prosjektet Dronning Eufemias gate og .ifc-filer fra to andre prosjekter i Skanska, Statoils kontorbygg på Sandsli i Stavanger og Veitvet skole i Oslo. Innmålingene er fra feltarbeid sommeren 2012 i Dronning Eufemias gate eller laget for dette formålet i ettertid. Programvare

som er benyttet er:

- OpenGeo Suite, en programpakke bestående av blant annet:
 - PostgreSQL 9.1, objekt-relasjonell databasehåndteringssystem
 - PostGIS 2.0, romlig utvidelse til PostgreSQL
 - pgAdmin3 for spørringer mot databasen
 - GeoServer for å lage karttjenester som WFS og WMS
- AutoCad Map 3D for å lage .dwg-filer av innmålingsdata. Visualisering av data fra databasen
- AutoCAD 2013
- Quantum GIS 1.8.0 'Lisboa' (QGIS) for visualisering av data i PostGIS
- Solibri Model Viewer for å se på modeller i IFC
- Dia modelleringsprogramvare for å lage konseptuelle modeller
- pgModeler for å modellere logiske datamodeller til PostGIS og PostgreSQL
- Ipe for å lage figurer
- SBG Geo, oppmålingsprogram i Skanska
- Python versjon 2.7 for å programmere egenutviklede programmer. Database-drivern *psycopg2* for å koble seg på PostgreSQL database

1.5 Forkortelser & begreper

Det er en del begreper som brukes litt om hverandre og jeg vil prøve å klargjøre noen av disse i dette kapitlet.

“Som bygget” og “som utført” modell er to begreper som brukes om hverandre, men beskriver egentlig det samme. Håndbok 138 bruker begrepet “som utført” [1]. Skal begrepene skilles kan “som bygget” brukes om innmålingene fra entreprenør før objektene i modellen oppgraderes til “som utført” av byggherren. I engelsk språk brukes begrepet “as-built” for dette og “som bygget” er den direkte oversettelsen av dette og brukes på prosjektet som er studert i denne oppgaven.

Med modelldata og modell menes en digital 3D-modell av en bygning, vei, vann og avløp eller lignende. En fagmodell er en fagspesifikk modell som for eksempel vann og avløp eller elektriske systemer. En modell er en forenklet representasjon av noe og ulikt et format som beskriver regler for hvordan noe skal lagres [9][10].

1.5.1 Forkortelser

- GIS - Geografisk informasjonssystem. Se definisjon i kapittel 2.1
- BIM - Bygningsinformasjonsmodellering. Se definisjon i kapittel 2.2
- FDV - Forvaltning, drift og vedlikehold
- SVV - Statens Vegvesen
- VA - Vann og avløp
- DBMS - databasehåndteringssystem. Se beskrivelse i kapittel 3.8.
- CAD/DAK - Computer Aided Drawing/Drafting. Dataassistert konstruksjon.
- 4D/5D - 4D: tidsdimensjon. 5D: kostnadsdimensjon
- SB/SU - "Som bygget/utført"
- SOSI - Samordnet Opplegg for Stedfestet Informasjon
- Solid - Volumobjekt
- OGC - Open Geospatial Consortium
- NVDB - Nasjonal Vegdatabank

Kapittel 2

Introduksjon

CAD og GIS er tradisjonelt to forskjellige verdener der førstnevnte representerer det som er bygget og sistnevnte representerer det som skal bygges. BIM hører ikke direkte til CAD-verdenen, men kan sees på som en videreutvikling av denne. Definisjon av GIS og BIM gjennomgås. Håndbok 138 legger føringer for modelldata i infrastrukturprosjekter slik at det gis en innføring i denne med vekt på fagmodellen vann og avløp.

2.1 GIS

GIS betyr geografisk informasjonssystem og anvendes på flere områder i et samfunn i dag. For eksempel i arealplanlegging, forvaltning, miljøovervåkning, flåtestyring, infrastrukturplanlegging og stedbaserte tjenester [11]. Representasjonsmetoder for geografiske data er behandlet i kapittel 3.1.

Det er flere definisjoner på GIS som varierer etter hvilken kontekst det defineres i, for eksempel i en analyse eller databehandlingskontekst. Worboys [12] definerer GIS ut i fra den sistnevnte konteksten:

A geographic information system (GIS) is a computer-based information system that enables capture, modelling, manipulation, retrieval, analysis and presentation of geographically referenced data.

2.2 BIM

BIM defineres av Eastman i *BIM Handbook* [13, s. 16] som “en modelleringsteknologi med tilhørende prosesser for å produsere, kommunisere og analysere bygningsmodeller.” En bygningsmodell er videre “kjennetegnet av: bygningskomponenter [...] som er digitalt representert (objekter) og parametriske regler som tillater intelligent manipulering, komponenter som inkluderer data som beskriver hvordan de oppfører seg etter behov ved analyse og arbeidsprosess [...]” *BIM Handbook* gir også en beskrivelse av hva BIM ikke er, nemlig statisk 3D-geometri med få eller ingen egenskaper til objektet som ikke har støtte for parametriske oppførsel (forandrer seg hvis omkringliggende objekter endrer seg) og en modell som er bygget opp med flere 2D CAD/DAK referansefiler for å definere modellen [13, kap. 1.5].

Begrepene bygningsinformasjonsmodell og bygningsinformasjonsmodellering brukes om hverandre i bransjen selv om det er en forskjell. Bygningsinformasjonsmodellering er en arbeidsprosess for å bygge opp en bygningsinformasjonsmodell. Modellen representerer anlegget som skal bygges eller er bygget og prosessen skal støtte opp under planlegging, bygging og drift gjennom hele anleggets levetid. Dette betyr at BIM er og blir en viktig del av arbeidsprosessen for byggherre, konsulenter og rådgivere, entreprenører, eiere av anlegg, leverandører av programvare og produkter til anlegget.

buildingSMART, tidligere IAI (den Internasjonale Alliansen for Interoperabilitet), ble etablert i 1994 og står bak en åpen, internasjonal standard for BIM som benyttes i dag. BIM er i følge buildingSmart bygget opp av 3 komponenter [14]:

- buildingSMART Datamodell (IFC - Industry Foundation Classes)
- buildingSMART Dataordbok (IFD - Industry Foundation Dictionaries)
- buildingSMART Prosess (IDM - Information Delivery Manuals)

Prosessen er mindre interessant i denne oppgaven enn datamodellen og dataordboken. En gjennomgang av datamodellen og dataordboken er foretatt i kapittel 3.5.1. Typisk for BIM er bruk av et lokalt ingeniør koordinatsystem (eng: engineer coordinate system), se kapittel 3.2.3 om ENU. Videre brukes typisk volumbasert geometri for oppbygging av de enkelte komponentene i bygningsmodellen. Data i modellen har også romlig struktur [15].

2.3 Statens Vegvesen Håndbok 138

“BIM for infrastruktur” er å anvende BIM-prosessen i infrastrukturprosjekter. B’en i BIM utgjør ikke bare bygninger, men alle typer konstruksjoner og anlegg [7]. IFC-modellen dekker primært bygninger, slik at andre formater brukes i disse prosjektene. I veiprojekter brukes Statens Vegvesens håndbøker og for prosjekter som bruker 3D-modeller brukes Håndbok 138.

Håndbok 138 stiller en rekke krav til partene i et veiprojekt hvordan data og modeller skal bestilles, utarbeides og leveres. Det spesifiseres hva slags modeller, geometri, formater, egenskapsdata, koordinatsystemer og lignende som skal benyttes i prosjektet [1]. Under følger en kort gjennomgang av Håndbok 138 med de delene som anses som viktige for denne oppgaven.

2.3.1 Modelltyper

Håndbok 138 lister opp to modelltyper som danner basis for flere undermodeller:

1. Grunnlagsmodeller
2. Fagmodeller

Grunnlagsmodellene beskriver eksisterende situasjon. Fagmodellene beskriver planlagt situasjon og prosjekteres av det enkelte fagmiljø. Se tabell 2.1 for en oversikt over de enkelte grunnlags- og fagmodeller. Ulike kombinasjoner av de to foregående modelltypene danner grunnlag for 3 nye modeller i tillegg til eksisterende og planlagt situasjon. Disse tre modellene er henholdsvis: 1) tverrfaglig modell (alle fagmodeller + grunnlagsdata), 2) presentasjonsmodell med tekstur (skilt/lyssetting, bilder etc) og 3) “som utført” modell. Den sistnevnte modellen representerer det som er bygget og danner grunnlag for eksport til forvaltning og driftssystemer [1, kap. 3].

Modelltype	Undermodell	Modellnavn
Grunnlag	Grunnlagsm. 1	Terrengoverflate
	Grunnlagsm. 2	Grunnforholdsmodell
	Grunnlagsm. 3	Eksisternde objekter
	Grunnlagsm. 4	Administrative forhold
Fagmodell	Fagmodell 1	Veg
	Fagmodell 2	Konstruksjoner
	Fagmodell 3	Tunnel
	Fagmodell 4	VA, grøft og drenering
	Fagmodell 5	Bergsikring og geotekniske konstruksjoner
	Fagmodell 6	Skilt, signal, oppmerkning
	Fagmodell 7	Vegutstyr
	Fagmodell 8	Kabelføringsanlegg
	Fagmodell 9	Tekniske installasjoner
	Fagmodell 10	Landskapstiltak
	Fagmodell 11	Reguleringsflater
	Fagmodell 12	Ytre miljø
	Fagmodell 13	Nye eiendoms- og rettighetsforhold, grunnerverv

Tabell 2.1: Modelltyper i Håndbok 138. Basert på tabell 5 [1].

2.3.2 Geometri, objektkoding & referanseramme

Objektene i en modell skal kodes etter en objektliste og det skal legges inn metadata for det enkelte objekt eller modell, enten direkte i objektet eller som et lagnavn til grupperte objekter.

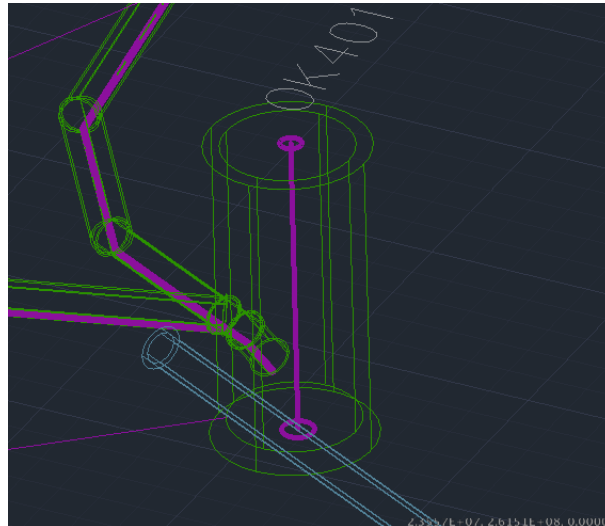
Prosjekterte objekter i fagmodeller skal leveres i geodetisk referanseramme EUREF89 med NTM projeksjon i grunnriss (X, Y). Høyden, Z, følger offisielt høydesystem på stedet, NN1954 eller NN2000. Se kapittel 3.2 for en gjennomgang av ulike koordinatsystemer.

De enkelte objekter kan være bygget opp av følgende geometrityper:

- Punkt (x, y, z)
- Kurve med jevn krumming og rette linjer
- Kurver med varierende krumming (klotoider etc)

- Flater
- Volumobjekter

Fagmodellene danner grunnlag for utsettings- og maskinstyringsdata. I enkelte fagmodeller hvor objekter er modellert som volumobjekter skal det være en egen geometri i form av en referanselinje eller referansepunkt som skal kunne benyttes til utsetting. Dette gjelder særlig kummer, rør og peler hvor dette brukes av stikkeren på prosjektet og er vist i figur 2.1 for kum og rør.



Figur 2.1: Modell og stikningsdata for overvannskum og rør i henholdsvis grønn og magenta farge. Skjermdump fra AutoCAD.

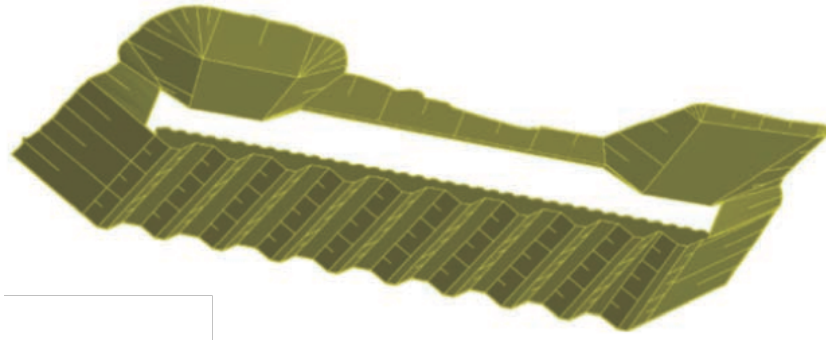
2.3.3 Grunnlagsmodell og grunnlagsdata

Grunnlagsdata skal beskrive eksisterende objekter på, under og over bakken i planleggingsområdet som representerer dagens situasjon. Det betyr data som blant annet viser høyder på terreng, vann (vassdrag, kanaler, grøfter), grenser, konstruksjoner i grunn samt VA og kabler, markslag, arealbruk, vegetasjon, samferdselsanlegg (vei, jernbane etc), kulturminner og verneområder utgjør grunnlagsdata. Fastmerker og grunnlagsnett hører også med inn under denne kategorien. I stor grad er dette FKB-data, *Felles Kart dataBase*, fra *Norge Digitalt*. Norge Digitalt er et bredt samarbeid mellom store virksomheter og nesten alle kommuner i Norge, som bruker eller fremskaffer stedfestet informasjon. Leveranser av grunnlagsdata gjøres hovedsakelig på SOSI-formatet. Terrengoverflatemodell skal leveres som en triangelmmodell på LandXML-format [1, kap. 2] [16].

2.3.4 Fagmodell vann og avløp

Fagmodellen VA dekker grøfter og rørledning i tillegg til vann og avløp. Det betyr at objekter som kummer og ledninger for gass, fjernvarme, vann, overvann, fundamenter og grøfter (byggegrop) inngår i denne modellen. Som vist i figur 2.1 skal

volumobjekter som kummer og rør ha referanselinje for bruk til utsetting. Figur 2.2 viser en byggegrep for konstruksjoner. Tilsvarende byggegrep brukes for grøfter.



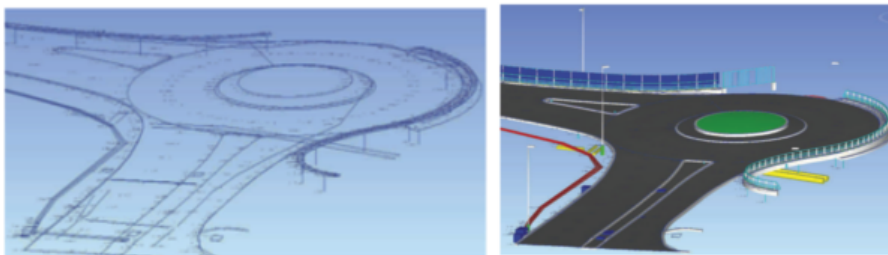
Figur 2.2: Gravegrop for konstruksjoner. Tilsvarende modell for grøft til VA. Figur hentet fra fra [1, figur 27].

2.3.5 “Som utført” & dokumentasjon

“Som utført” modell dokumenterer hva som ble bygget og eventuelle endringer fra prosjektert data er modellert opp på nytt. Det kan for eksempel være ny plassering av objektet eller oppdatert informasjon over grunnforhold (avdekket fjell, løsmasser etc). “Som utført” modell baserer seg på innmålinger fra entreprenør. Innmålt geometri sammen med modelldata er vist i figur 2.3.

Entreprenør kan levere innmålinger eller oppdatere en modell på tre måter:

1. Oppdatere prosjektert modell med reviderte objekter og levere til godkjenning av byggherre.
2. Levere nok innmålinger i modellens originalformat til rådgiver/konsulent som kan modellere opp og endret statuskode på objekter til “som utført” for byggherre.
3. Bearbeide data fra skanning (laser) og objekter bygget opp fra punktsky.



Figur 2.3: Innmålt geometri (venstre) og som “utført modell” med innmålt geometri vist sammen (høyre). Figur hentet fra [1, kap. 3.6].

Kapittel 3

Teori

Formater som typisk brukes i GIS, CAD og BIM gjennomgås for å gi en bakgrunn i formater som brukes i dag. Ulike representasjonsmodeller for geografiske data, geometri, koordinatsystemer og topologi er sentrale emner i både GIS og BIM og gis derfor en behandling. Databasesystemet som er valgt for prototypen behandles for å gi en oversikt over muligheter og begrensninger. Standarder for overføring av geografisk data er knyttet til GIS og behandles for å kunne gi en bakgrunn for tjenester som kan bygges på en geografisk database. Prosessen med datamodellering behandles også.

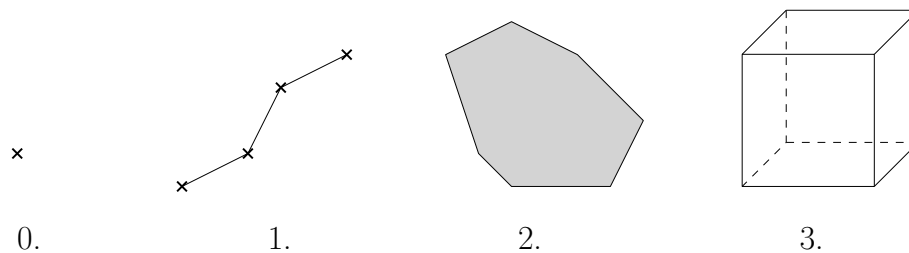
3.1 Geografiske data & geometri

Begrepet romlig data (eng: spatial) benyttes også om geografiske data. Geografiske data kan for eksempel være en vei, by, bygning, land, adresse, elver, høyde (terreng) eller skog. Dette er det geografiske temaet og består av flere geografiske objekter som igjen er bygget opp av to deler [17, kap. 1.2]:

- beskrivende egenskaper
- romlig komponent (geometri eller topologi)

Denne tankegangen brukes i GIS, men den gjelder også til en viss grad for BIM. Egenskaper kan være navn på by eller vei, antall innbyggere, nedbørsmengde, tema-kode eller lignende. Den romlige komponenten til et geografisk tema som en by kan være representert som et punkt eller som et polygon for å vise en mer nøyaktig avgrensning på byen. For å representere geografiske data benyttes det to modeller for å beskrive og modellere geografiske data; objektbasert og feltbasert.

I den objektbaserte modellen sees verden på som en samling av objekter hvor geometrien fyller (avgrenser) rommet og tilhørende egenskaper på objektet. Geometrien er bygget opp av punkter, linjer, flater og volumobjekter [17, kap. 2.1]. Geometrien kalles for vektormodellen og er vist i figur 3.1 sammen med den topologiske dimensjonen på det tilhørende topologiske primitivet. Det engelske ordet *boundary representation* (b-rep) er også mye brukt om vektormodellen. Den objektbaserte modellen er brukt i GIS, CAD og BIM.

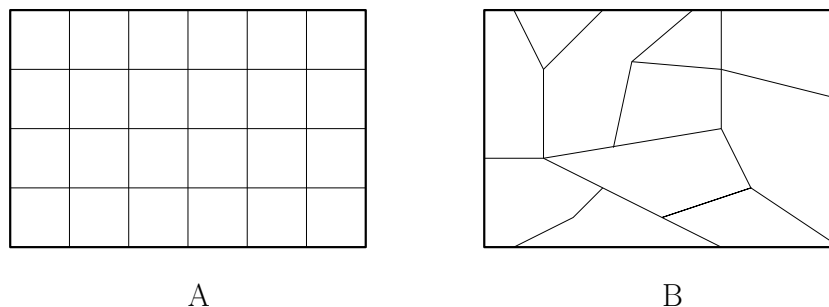


Figur 3.1: Vektormodellen og tilhørende topologisk dimensjon for punkt (node), linje (kant), polygon (flate) og volumobjekt (volum). Topologisk primitiv i parentes.

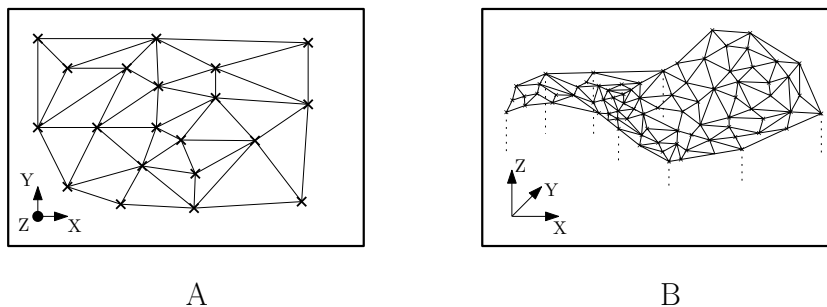
I den feltbaserte modellen er verden en kontinuerlig funksjon i x og y og verdien i ett gitt punkt, (x, y) , er egenskapen ved feltet på dette punktet. For eksempel kan høyden i ett gitt punkt sees på som en funksjon av det gitte punktet:

$$H = f(x, y)$$

Temperatur og nedbør er andre eksempler på data som kan uttrykkes gjennom feltmodellen. Feltmodellen bruker en oppdeling av planet i celler og kommer som oftest i form av et regulært rutenett. Dette kalles også for et raster hvor hver enkelt celle kalles en pixel. Dette er den vanlige lagringsmåten for felldata. Dette er en form for regulær tesselering, men planet kan også ha en irregulær tesselering. Regulær og irregulær tesselering av et plan er vist i figur 3.2. Felldata kan representeres i objektmodellen som TIN, triangulert irregulært nettverk (eng: Triangulated Irregular Network) [17, kap. 2.1, 2.2]. TIN og raster i form av et bilde brukes ofte i digitale terrengmodeller. Triangulering og TIN er vist i figur 3.3.



Figur 3.2: A) Regulær og B) irregulær tesselering av felldata. Figur basert på [17, figur 2.3, 2.4].



Figur 3.3: A) Triangulering. B) TIN. Figur basert på [17, figur 2.9].

3.2 Koordinatsystemer

GIS-verdenen er opptatt av å lokalisere objektet til dets fysiske plassering på jorda og bruker derfor en jordfast referanseramme. CAD- og BIM-verdenen er ikke opptatt av den fysiske plasseringen i første omgang og bruker derfor en referanseramme med lokale modellkoordinater i første omgang. Modellen kan i ettertid gis en fysisk plassering.

Et koordinatsystem har en unik SRID, *Spatial Reference System Identifier*, som brukes for å identifisere hvilket koordinatsystem som brukes. EPSG, *European Petroleum Survey Group*, er en organisasjon som har laget en katalog med definisjoner av mange ulike koordinatsystemer og er den mest brukte katalogen for å identifisere koordinatsystemer. SRID- og EPSG-koden blir ofte brukt om hverandre for å identifisere koordinatsystemer [18][19].

I geodesien brukes referanseramme om en realisering av et referansesystem gjennom observasjoner og definerer blant annet origo og aksenes retning ved et gitt tidspunkt [20, kap. 2][21, kap. 5.4]. Det kan være horisontale og vertikale referanserammer hver for seg, eller en samlet 3D-referanseramme [21, kap. 5.4]. Betegnelsen datum brukes også om en referanseramme. Et koordinatsystem i en geodetisk sammenheng er en del av en referanseramme for å kunne angi en posisjon i den gjeldende referanserammen og kan være kartesiske, polar-, kule- eller sylindervektorkoordinater. Det finnes flere referanserammer som kan være lokale, regionale eller globale. ETRS89/EUREF89 og WGS84 er et eksempel på henholdsvis en regional og global referanseramme.

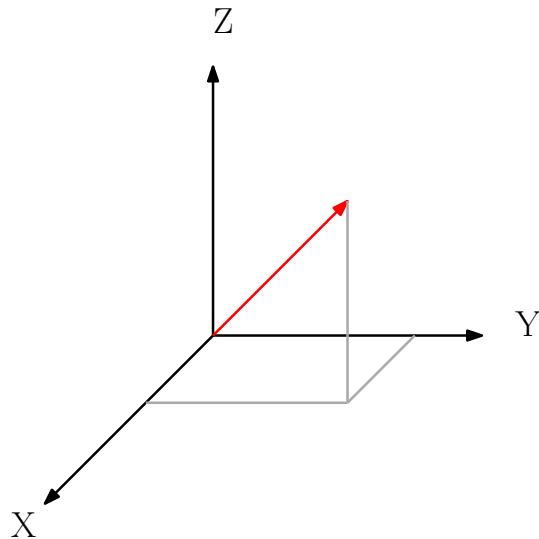
En grov oppdeling av koordinatsystemer kan gjøres i fire typer:

1. Lokale koordinater (modellkoordinater)
2. Kilometrering
3. Geodetiske koordinater
4. Projiserte koordinater

3.2.1 Lokale koordinater

Lokale koordinater er kartesiske koordinater i et høyre- eller venstrehåndssystem. Dette er typiske koordinater som brukes i 3D modelleringsprogramvare. Et lokalt

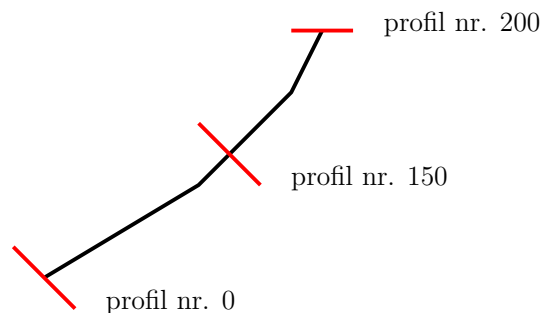
koordinatsystem er ikke tenkt å kunne kobles til en jordfast referanseramme slik som et lokalt geodetisk system (ENU). Dette behandles i kapittel 3.2.3. Et lokalt system er vist i figur 3.4.



Figur 3.4: Kartesisk høyrehåndssystem for lokale koordinater.

3.2.2 Kilometrering

Kilometrering er et 1-dimensjonalt koordinatsystem som brukes i vei- og jernbane-prosjekter. Det kan også brukes i prosjekter med ledninger. Profilnummeret forteller hvor langt langs en linje, typisk en senterlinje, en er fra starten eller slutten på linja [22]. En slik linje og profilnummer 0 til 200 er vist i figur 3.5.



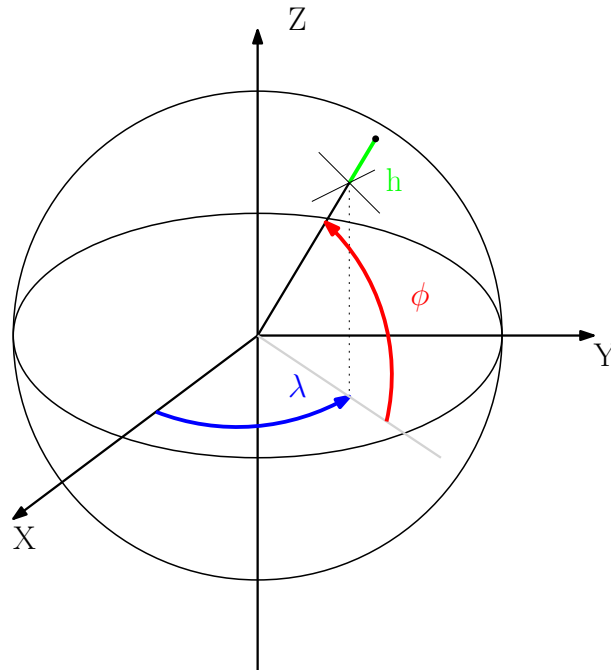
Figur 3.5: Profilnummer og linje i koordinatsystem av type kilometrering.

3.2.3 Geodetiske koordinater

Geodetiske koordinater er lengde- og breddegrad og ellipsoidisk høyde (ϕ, λ, h) gitt i en bestemt ellipsoide. En ellipsoide er en matematisk beskrivelse av jorda [21, kap.

5.2]. Satellitnavigasjonssystemet *GPS* (Global Positioning System) bruker WGS84-ellipsoiden som er nesten identisk med den globale GRS80-ellipsoiden. Ellipsoidisk koordinatsystem er vist i figur 3.6. Geografiske koordinater er geodetiske koordinater i 2D (ϕ, λ) og mye brukt i GIS [21, kap. 4.3][23, kap. 8].

Høyden vi får i et geodetisk system med en ellipsoide er ellipsoidisk høyde og denne er ulik den ortometriske høyden, høyde over havet, som er den høyden vi er mest interessert i. For å gå fra ellipsoidisk høyde til ortometrisk høyde trenger vi en geoidmodell av jorda [23, kap. 8]. Det er utenfor denne oppgavens rammer å gå videre inn på dette.

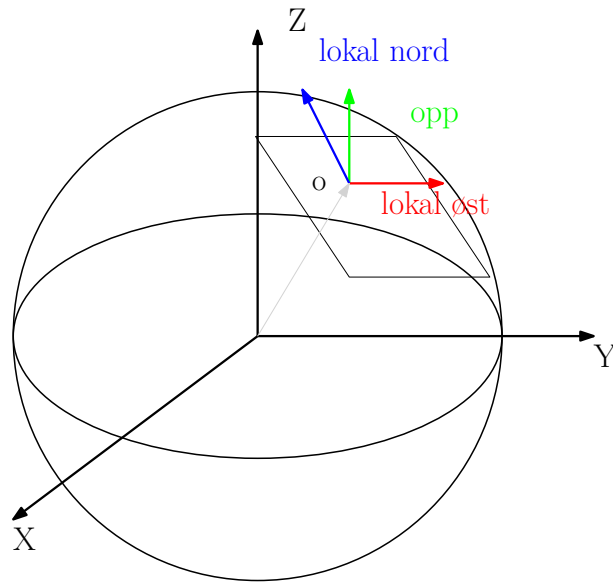


Figur 3.6: Geodetisk lengde- og breddegrad og ellipsoidisk høyde. Figur basert på [21, fig. 5.3].

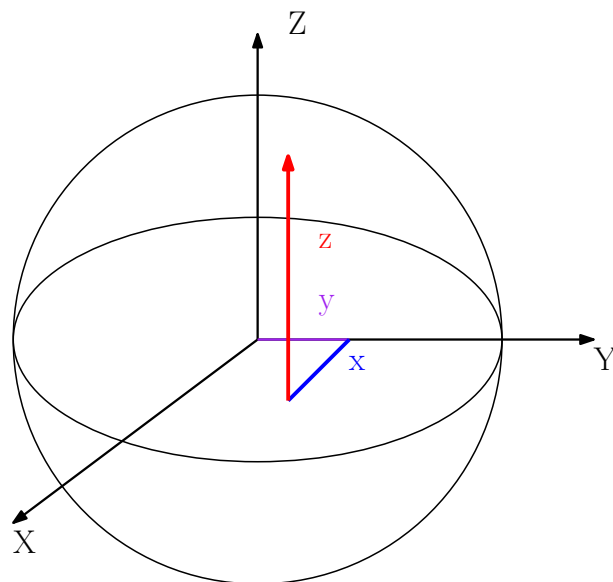
Et lokalt geodetisk koordinatsystem er av typen ENU, *East, North, Up*, og med tilhørende translasjons-, rotasjons- og målestokkparametre kan dette lokale systemet transformeres til et globalt geodetisk koordinatsystem eller vice versa [21, kap. 5.3]. Dette kalles ofte i engelsk språk for *engineering coordinate system* og er vist i figur 3.7. Et slikt lokalt system er ofte brukt i CAD- og BIM-modeller på bygg- og anleggsprosjekter.

Geosentriske koordinater er kartesiske koordinater (X, Y, Z) med origo $(0, 0, 0)$ i jordas massesentrum. Dette kalles ofte også for ECEF, *earth-center earth-fixed*, eller jordsentrisk-jordfaste koordinater [21, kap. 5.1]. Et ECEF-koordinatsystem er vist i figur 3.8.

Det finnes metoder i dag for å konvertere fra et koordinatsystem til et annet og mellom ulike referanserammer med ulike transformasjonsparametre. For eksempel bruker satellitnavigasjonssystemet *GPS* geosentriske koordinater, men mottakeren viser ofte geografiske koordinater (ϕ, λ) og høyde (ellipsoidisk eller ortometrisk). Mottakeren kan også vise projiserte koordinater [21, kap. 5][23, kap. 9.2].



Figur 3.7: Lokalt geodetisk koordinatsystem: East (Øst), North (Nord) og Up (opp/høyde). Figur basert på [21, fig. 5.1].

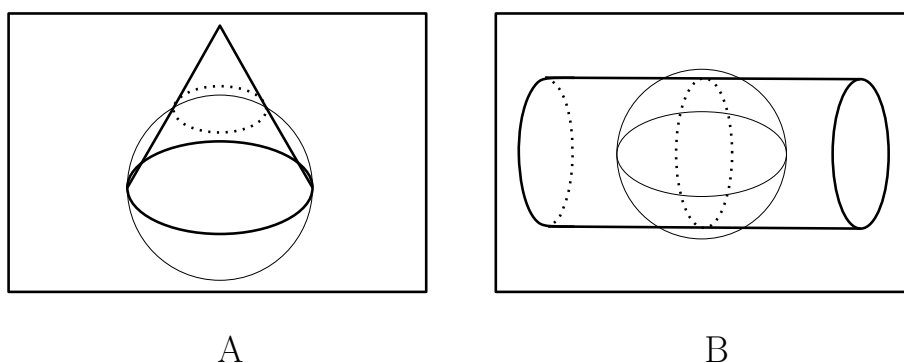


Figur 3.8: ECEF-“Earth Centered Earth Fixed” kartesisk koordinatsystem. Figur basert på [21, fig. 5.1].

3.2.4 Projiserte koordinater

Projiserte koordinater knyttes til en spesifikk kartprojeksjon hvor visse egenskaper er valgt til å bli bevart under transformasjon fra geodetiske koordinater til kartplanet. En konform kartprojeksjon bevarer vinklene, ekvidistante projeksjoner bevarer avstander og ekvivalente projeksjoner bevarer arealet. Projeksjonsflaten kan blant annet være stående eller liggende sylinder, kjegle eller et plan som tangerer overflaten på jorda i et punkt. Projeksjonsflaten “brettes ut” til et kartplan. Valg av projeksjonsflate har sammenheng med type projeksjon og hvilke egenskaper (konform, ekvidistant, ekvivalent) som ønskes å bevares [1, vedlegg, kap. 1.3][21, kap. 8.1, 8.2][23, kap. 8].

I Europa og Norge er ETRS/EUREF89 og UTM-projeksjonen dominerende for offentlige kart [24]. Kartplanet i UTM-projeksjonen er skalert med en faktor 0.9996 i sentralmeridianen og gir en maks feil på $400ppm$ (4 cm på hundre meter) i grunnriss ved utsetting sammenlignet med beregnet verdi. Innenfor en UTM-sone varierer denne feilen etter hvor langt du er fra sentralmeridianen. Bygg- og anleggsbransjen i Norge har tatt i bruk NTM, *Norsk Transversal Merkator*, for prosjektering og som ikke bruker en målestokk i kartplanet (i motsetning til 0.9996). Det gir en teoretisk maks feil på 11 ppm ($\approx 1mm$ på 100m) [1, vedlegg kap. 1.3]. En kjegle og liggende sylinder som projeksjonsflate er vist i figur 3.9. UTM og NTM bruker begge en liggende sylinder. Projiserte koordinater er vanlig i GIS, men modeller i CAD og BIM kan også bruke denne type koordinater, se kapittel 3.5.



Figur 3.9: A) Kjegleprojeksjon. B) Transversal sylinder. Figurer basert på [21, fig. 8.13, 8.14].

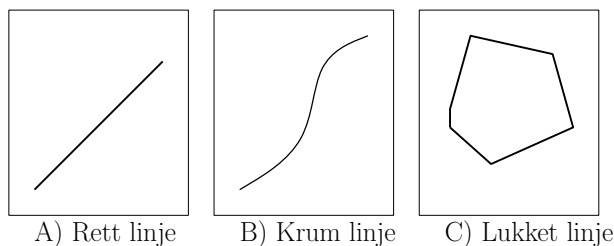
3.3 Topologi

Topologi og mengdelære er store og sentrale områder innenfor matematikken og kommer til anvendelse i relasjonelle databaser og GIS. Topologi har også stor betydning for oppbygging av modeller i BIM. Mengdelæren danner det teoretiske grunnlaget for relasjonsdatabaser. En mengde er en samling objekter og kan for eksempel være punkter, linjer, kuber eller observasjoner av et fenomen eller noe annet.

Topologi baserer seg på mengdelæren og er et uttrykk for kontinuitet og konnektivitet/forhold mellom to mengder. Kontinuitet betyr i denne sammenheng at konnek-

tive egenskaper som naboskap og forhold bevares gjennom kontinuerlig forandring som skalering, strekk/krymp, rotasjon og forskyvning (translasjon) [25, kap. 3.5]. Det vil si at endring i målestokken i kartet ikke endrer på forholdene mellom objektene. Egenskaper som hvor lang, bredt, høyt, areal, volum, form er ikke topologiske egenskaper, men geometriske egenskaper.

Figur 3.10 viser tre linjer: A) en rett, B) en krum og C) en lukket linje. Det er ingen topologisk forskjell på den rette (A) og krumme linja (B), men en lukket linje (C) er forskjellig fra en åpen linje (A og B). Linjene i A og B kalles også homeomorfe ved at det ene objektet kan avledes av det andre uten å splitte eller klippe objektet. Egenskapen åpen/lukket er dermed en topologisk egenskap [25, kap. 3.2].



Figur 3.10: Linjer med ulike topologiske egenskaper.

3.3.1 Spaghetti-modellen

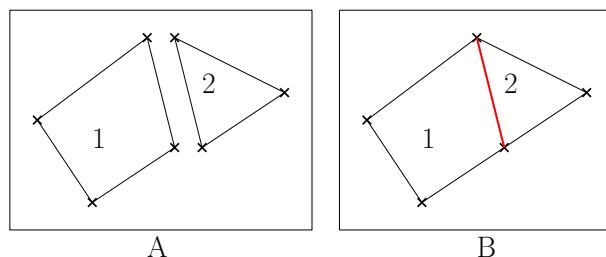
Spaghetti-modellen er en betegnelse for data hvor hvert objekt er individuelt og felles geometri er duplisert. Dette er data som ikke er bygget opp med topologi. Figur 3.11.A er en spaghetti-modell hvor eiendommene vises som to separate objekter (polygoner). På figuren er disse skilt fra hverandre for å vise at polygonene er separate.

3.3.2 Topologisk modell

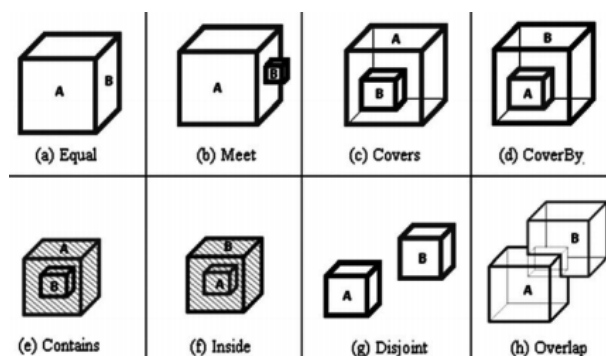
I en topologisk modell for data er geometri bygget opp av noder, kanter og flater for å unngå duplisering av felles geometri. Dette gjelder både for GIS- og BIM-standarder som henholdsvis GML og IFC [26, kap. 8.20.1][27, kap. 13]. I GIS uttrykker topologi naboskap, forhold og nærhet mellom objekter. For nettverk som for eksempel VA eller ruteplanlegging er topologi nødvendig for å gjøre nettverksanalyser og korteste-vei problemer [28]. I BIM og IFC-formatet uttrykker topologi også det samme, slik at et vindu er koblet til riktig vegg (eng: relate). Veggene er videre relatert til en gitt etasje i bygget.

Topologi kan avledes og bygges opp fra geometri med verktøy. En modell for topologi som benyttes i for eksempel nettverk i dag er to kolonner, *fraNode* og *tilNode* for en ledning/vei (linje) i tabellen over linjene [28].

Figur 3.11.B er en topologisk modell for en eiendom og den felles eiendomsgrensen mellom eiendommene er farget rødt. Noder, kanter og flater som bygger opp eiendommene er lagret separat og ikke som to enkelte polygoner som i figur 3.11.A.



Figur 3.11: A) Spaghetti-modell. B) Topologisk modell med felles geometri uthevet i rødt. Noder vist som kryss.



Figur 3.12: Topologiske forhold mellom 3D objekter. Figur hentet fra [32].

3.3.3 Topologisk dimensjon & 9-snittsmodellen

Topologisk dimensjon til et objekt er vist i figur 3.1 for vektormodellen. En node (eng: node) er dimensjon 0, en kant (eng: edge) er i dimensjon 1, en flate (eng: face) er i dimensjon 2 og et volumobjekt er i dimensjon 3. Forhold mellom objekter brukes for å finne objekter som er koblet/relatert til hverandre, overlapper eller inne i. Dette er en form for kollisjonskontroll (eng: clash detection) som brukes i BIM-modeller for å unngå at for eksempel rør og ventilasjonskanaler ikke får nok plass og kolliderer i modellen. Krasjtesting brukes også om dette.

Egenhofer og Franzosa's 9-snittsmodell (eng: the dimensionally extended nine-intersection model, DE-9IM) er et rammeverk som gir en systematisk beskrivelse av topologiske relasjoner mellom objekter (geografiske og ikke-geografiske) og tar utgangspunkt i omriss (grense), indre og ytre til to objekter, A og B [25, kap. 3.6]. *OpenGIS Simple Feature* spesifikasjonen bruker 9-snittsmodellen for å bestemme relasjoner mellom objektene A og B [29, s. 80].

3D-objekter (volumobjektet) byr på en del utfordringer når topologiske forhold skal utledes. Det er forsøkt med flere modeller for 3D topologi (3D FDS, TEN, SSS, UDM, SOMAS, OO3D) og de har hver for seg styrker og svakheter [30][31]. 9-snittsmodellen beskriver $2^9 = 512$ mulige relasjoner for to 3D-objekter hvorav 8 er mulige og vist i figur 3.12. De 8 mulige relasjonene er *equal*, (lik), *meet* (berører), *covers* (dekker), *coveredBy* (dekket av), *contains* (inneholder), *inside* (inne i), *disjoint* (disjunkte/atskilt) og *overlap* (overlapp) [17][30][32].

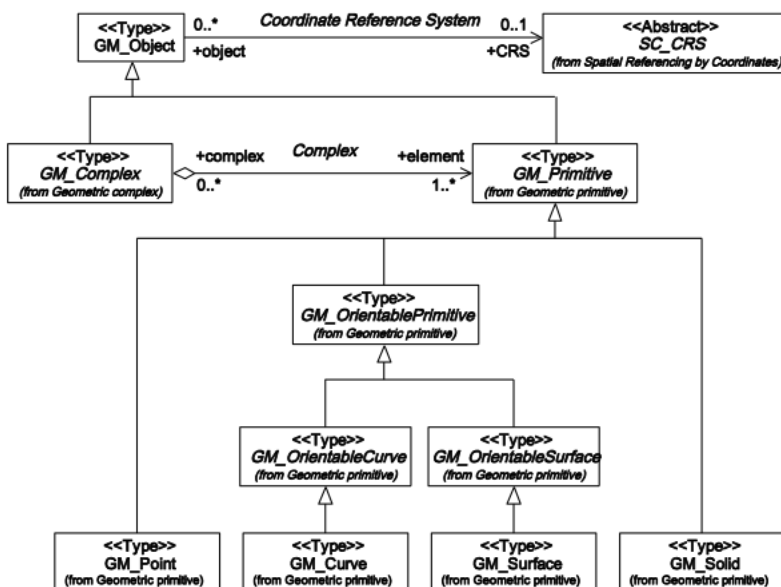
3.4 GIS-standarder

3.4.1 GML

Geography Markup Language (GML), er en standard som brukes til å uttrykke geografiske objekter og er et åpent format for å utveksle og lagre geografisk data. Det er lagt opp til at brukergrupper kan lage egne applikasjonsskjemaer (eng: application schemas) som en utvidelse av GML-standarden for å dekke spesielle behov. CityGML er et eksempel på en slik utvidelse med fokus på 3D bymodeller. *Open Geospatial Consortium* (OGC) står bak denne standarden som modellerer ISO 19100-serien som er utarbeidet av ISO/TC211. Siste versjon er 3.3, men det er små forskjeller slik at versjon 3.2.1 er gjennomgått her [33].

GML har en rik geometrimodell i bunn basert på vektormodellen, se figur 3.1. Primitiver for geometri er vist i figur 3.13. En `GM_Curve` kan ha interpolasjonstyper som blant annet lineær, geodetisk, splines (polynomisk og kubisk, bsplines og bezier), klotoid og sirkulær for å lage sirkler. For `GM_Surface` er undertyper polygoner med og uten hull, polyhedral surface (bygget opp av polygoner), triangulert overflate og triangler, kjegle, sylinder og kule. `GM_Solid` er bygget opp av sine respektive flater (eng: shell) som avgrenser objektet [27, kap. 10, annex D.2.3].

GML har en topologimodell bestående av noder, kanter, flater og “*TopoSolid*” som et 3-dimensjonalt topologisk primitiv. I GML er det også egne skjemaer (eng: schema) for blant annet observasjoner, felt (eng: coverages), målinger og temporal informasjon. Objektene kan ha egenskaper og metadata knyttet til seg. Det er støtte for mange ulike koordinatsystemer, både geodetiske, projiserte, lokale (eng: engineering coordinate system), vertikale og temporale. Det er ikke funnet noe om kilometrering [27].



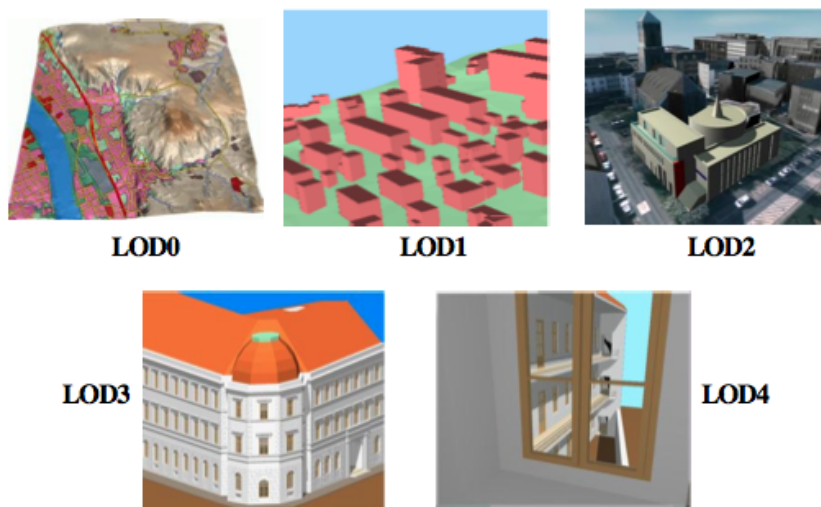
Figur 3.13: Geometriprimitiver i GML 3.2.1. Figur D.2 gjengitt fra [27].

3.4.2 CityGML

CityGML er en åpen standard basert på GML versjon 3.1.1 for representasjon og utveksling av virtuelle 3D-bymodeller og er en standard under OGC. *CityGML* er både en modell og et format. Siste versjon er 2.0 fra 2012 [34][35].

I *CityGML* er det fire aspekter som ligger til grunn for en 3D-bymodell; geometri, topologi, semantikk og utseende. Semantikk i *CityGML* betyr at et objekt i en bymodell tilhører en bestemt klasse, som en bygning, transport, vann, eller vegetasjon. Utseende til en 3D-modell er gitt ved tekstur fra bilder eller materialdefinisjoner. Egenskaper til objektet er lagt opp til å hentes eksternt for å sikre oppdaterte data. [34].

Et enkelt objekt i *CityGML* kan modelleres i 5 ulike nivåer, *Level of Details* (LOD), med høyere nøyaktighet og detaljering. *LOD0* er en 2.5-dimensjonal terrenngmodell (DTM), *LOD1* er blokker uten tak, *LOD2* er blokker med noe takkonstruksjoner og større bygningsinstallasjoner som balkonger og trapper. *LOD3* bringer inn detaljerte vegger og takkonstruksjoner. En modell i *LOD4* har med indre strukturer som rom, innvendige trapper og møbler. De ulike nivåene er vist i figur 3.14. Til de ulike nivåene er også en absolutt posisjonsnøyaktighet angitt i 3D, for eksempel 0.2 meter for *LOD4*. Bruksområde er navigasjon, turisme, byplanlegging, simulator eller lignende [34] [35, kap 6.2].



Figur 3.14: 5 “Level of Details” i *CityGML*. Figur 3 gjengitt fra [34].

Geometrien i *CityGML* 2.0 er basert på et utdrag av GML 3.1.1 sin geometri-modell. En kurve er begrenset til å være en rett linje og ingen andre interpolasjonsmetoder. Polygoner for overflater og punkt er støttet. Triangulert overflate og solids (volumobjekter) er også støttet. Støtten for å håndtere koordinatsystemer er arvet fra GML, men i tillegg kan lokale koordinatsystemer for objekter som går igjen, som for eksempel skilt i en bymodell, brukes. Disse objektene kalles for “*Implicit Geometries*” og består av et forankringspunkt i det jordfaste koordinatsystemet og en transformasjonsmatrise [35, kap. 8].

CityGML bruker ikke GML sin topologi-modell, men har en eksplisitt støtte

for topologi gjennom at to objekter kan dele felles geometri. For eksempel kan en garasjevegg og en husvegg dele på geometrien som definerer begge veggene. Det er valgfritt om en vil bruke topologi i en modell i CityGML [35, kap. 8].

CityGML er et applikasjonsskjema til GML og det er lagt opp til at brukere selv kan lage egne utvidelser til CityGML, såkalte *Application Domain Extensions* (ADE). For eksempel kan en utvidelse være for å beregne støy i bymodeller [34].

3.4.3 SOSI ledning 4.5

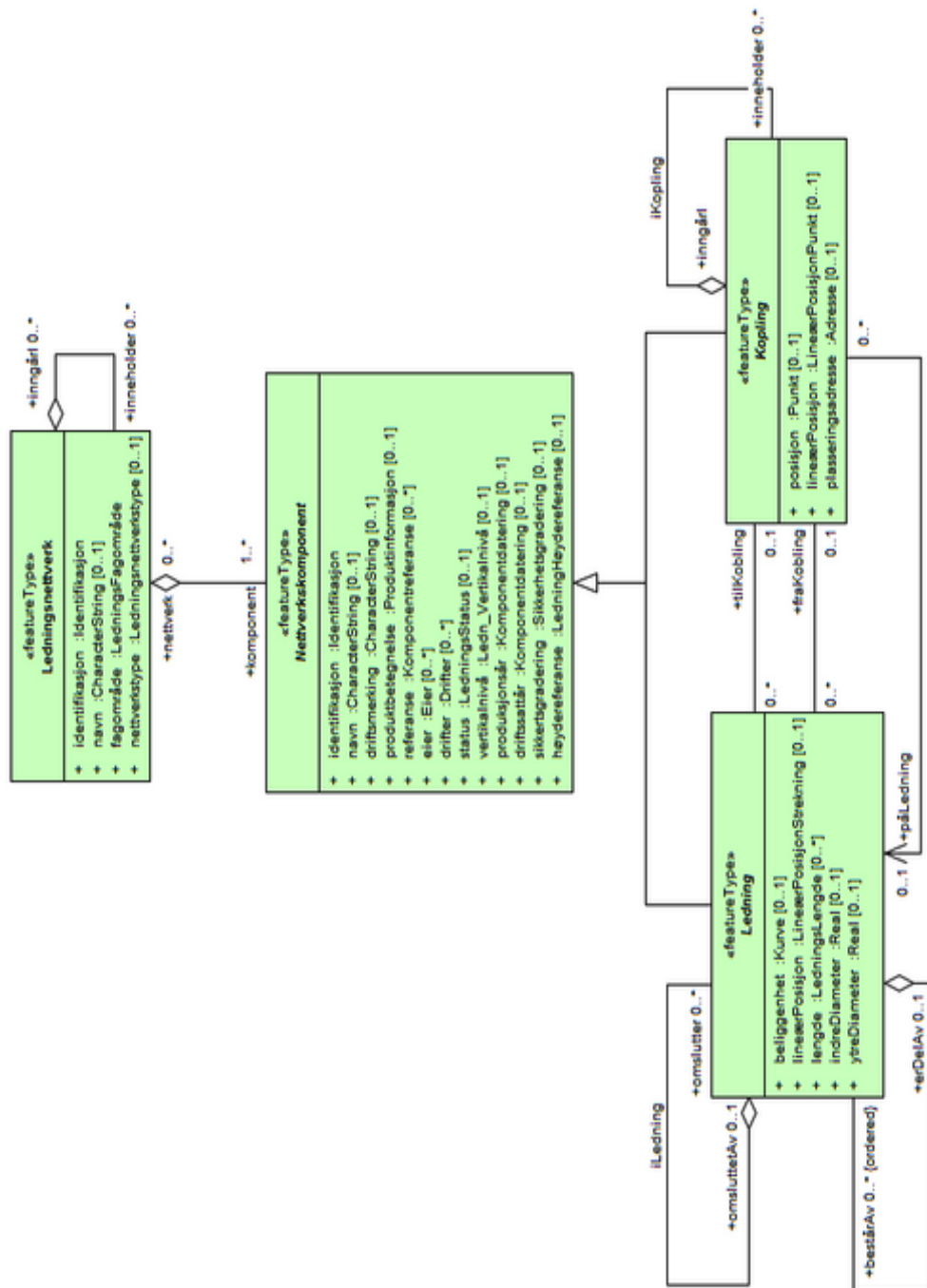
SOSI ledning versjon 4.5 var ute til høring fram til 15. februar 2013 og UML-modellen ble stengt for endringer 7. mars 2013 [36]. Det er denne modellen som er brukt som bakgrunn for den forenklede fagmodellen vann og avløp i kapittel 5.2.1 og som er gjengitt under. Versjon 4.5 er en total revidering av hele modellen for ulike ledningsnett [37, kap. 2, 3]:

- vann/avløp
- telekommunikasjon/signalanlegg
- elektrisitet
- fjernvarme, olje og gass (ikke i standarden på nåværende tidspunkt)

SOSI ledning går bort fra SOSI-formatet for lagring og blir realisert i GML-formatet. GML er gjennomgått i kapittel 3.4.1. I standarden er det spesifisert bruksområder som overføring fra landmåling/felt til ledningsdatabase og overføre ledningsdata til planleggingssystemer som GIS og BIM. Objektene i dagens utgave inneholder senterpunkt/senterlinjer og noen egenskaper for ulike dimensjoner på enkelte objekter, som for eksempel indre og ytre radius på rør, men det er ønskelig at modellen for ledningsnett på et senere tidspunkt kan håndtere volumobjekter [37, kap. 3, 8].

SOSI ledning har en *kjernemodell* bestående av fire klasser som er felles for alle fagområdene. Et hvilket som helst ledningsnettverk, enten det er vann og avløp eller elektrisitet, består av nettverksobjekter som en *kopling* eller *ledning*. Topologi i ledningsnettet håndteres i kjernemodellen gjennom *fraKopling* og *tilKopling* eller på ledningen direkte. Dette kalles en anbring. Kjernemodellen er vist i figur 3.15.

For vann og avløp er *VA-kopling* en spesialisering og kan blant annet være en kum, hydrant eller en ventil. *VA-ledning* er en spesialisering av ledning og kan være en overvannsledning, avløpsledning eller vannledning. Modellen for vann og avløp er vist i figur 3.16.



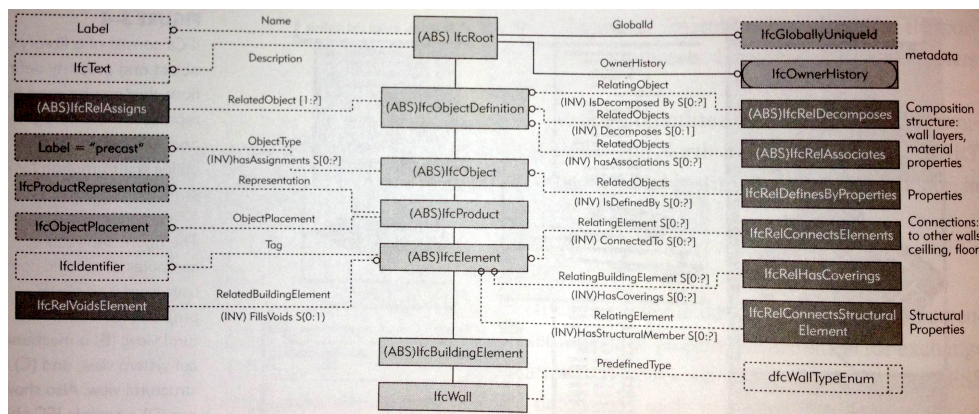
Figur 3.15: Kjernemodell i SOSI ledning. Skjermdump fra UML-modell.

3.5 CAD- & BIM-standarder

3.5.1 IFC

Industry Foundation Classes (IFC) er datamodellen i BIM og et lagringsformat. Siste versjon er IFC4 og ble lansert 12. mars 2013, mens forrige versjon som er mye brukt er IFC2x3 [38]. IFC er tenkt å dekke alle behov en modell av en bygning har fra den planlegges og designes inkludert analyser og simulering, bygging, bruk og drift, kort sagt hele bygningens levetid [13, kap. 3.3.3].

Datamodellen i IFC er objektorientert og hierarkisk bygget opp. Dette betyr at et hvilket som helst objekt kan spesialiseres ut i fra et mer generelt objekt. I bunnen ligger de mest generelle entitetene som geometri, materiale, topologi, geometrisk modell, geometriske betingelser, målinger, egenskaper og representasjon, totalt 26 ulike entiteter. Disse entitetene brukes for å bygge opp bygningsobjekter som en vegg, gulv, bjelker, rør eller et vindu, fasilitets- eller forvaltningsobjekter. På toppen i datamodeller er de spesifikke fagområder eller domener som blant annet brannsikring, varme-, ventilasjons- og sanitærteknikk (VVS), vann, elektrisitet, arkitektur, forvaltning og drift. Figur 3.17 viser hvor omfattende hierarkiet og oppbyggingen av en .ifc-fil er bare for en enkelt vegg. Hvert nivå i figuren viser ulike egenskaper og relasjoner vegg har med andre objekter. *IfcObjectDefinition* har en oversikt over hvilke andre objekter som denne vegg består av, som for eksempel en dør eller et vindu. *IfcObject* inneholder blant annet egenskaper til vegg basert på hva slags type vegg og *IfcElement* har de topologiske egenskapene som hvilke andre objekter er relatert til denne vegg, som gulv, vinduer og dører. Plassering og form er definert i *IfcObject*. En .ifc-fil er også organisert hierarkisk etter *Project - Site - Building - BuildingStorey - Space* hvor høyere nivåer består av nivåene under [13, kap. 3.3.3].



Figur 3.17: Oppbygging i ifc-formatet for å definere en vegg. Figur hentet fra [13, fig. 3.6].

I IFC4 er geometrien i hovedsak bygget opp av volumobjekter som kan være representert med vektormodell (se figur 3.1), geometriprimitiver som kuler og kjegler, “Non-Uniform Rational B-Spline” (NURBS) og “Constructive Solid Geometry” (CSG) som er resultatet fra boolske operasjoner (union, snitt, differanse) på geometriske primitiver. Punkter, linjer og flater er også støttet [26, kap. 4.][39].

I IFCx3 kan *IfcSite* gis en jordfast posisjon i form av lengde-, breddegrad og høyde for å kunne vise modellen med sin jordfaste plassering for visualiseringsformål. Nord-retning angis i *IfcGeometricRepresentationContext* [40]. I IFC4 er det lagt opp til å kunne velge mellom et lokalt system (“local engineering coordinate system”) eller et annet spesifisert system. Det angis ikke direkte en EPSG kode (se kapittel 3.2), men det legges opp til at et av disse koordinatsystemene skal brukes. Et projisert koordinatsystem (“IfcProjectedCRS”) kan også brukes for å transformere det lokale systemet til en fysisk plassering [26, kap. 8.18].

IFD er en dataordbok for å skape en felles terminologi på tvers av landegrenser. I dag brukes begrepet *buildingSMART Dataorbok* i stedet for IFD. Et eksempel er hva som legges i begrepet “dør” på norsk og “door” på engelsk. På norsk vil vi snakke om hele døra, men på engelsk er det snakk om dørflaten. Det riktige begrepet er “door set” i dataorboken når vi snakker om en dør [14][41].

3.5.2 DWG

AutoCAD er et CAD-program som bruker .dwg-formatet for å lagre 2D- og 3D-data og metadata [42]. Formatet er proprietært slik at en beskrivelse ikke er tilgjengelig, slik at det som framkommer om AutoCAD og dermed DWG er tilegnet gjennom bruk av AutoCAD.

AutoCAD har mulighet for å tegne mye ulik geometri, både splines, linjer, flater og volumobjekter. Geometrien er statisk. Det er ikke støtte for topologi gjennom for eksempel å dele felles geometri med noder, kanter eller flater. Flere enkeltstående geometriobjekter kan grupperes i et lag. Laget kan gis en bestemt farge og lagnavn. Det er ikke mulig å legge til flere egenskaper på den grupperte geometrien enn lagnavnet. AutoCAD bruker et lokalt koordinatsystem og modellen kan gis koordinater i jordfast koordinatsystem. Det er ikke undersøkt hvordan dette gjøres.

3.5.3 LandXML

LandXML er et utvekslingsformat for stikningsdata til oppmålings- og maskinstyrringsutstyr i samferdsel- og infrastrukturprosjekter. De fleste verktøyene for prosjekterings kan importere og eksportere LandXML. Formatet er xml-basert. Formatet trekkes fram i Håndbok 138 som et åpent utvekslingsformat for modell- og stikningsdata i tillegg til originalformatet. Det er noen utfordringer med ulike “dialekter” av formatet og hvordan et program håndterer alle egenskaper på et objekt [1][43].

LandXML har støtte for punkter og linjer. For vei-modellen i LandXML er senterlinje (alignment), profiler (profiles) og tverrprofiler (cross-sections) støttet. Det er også en egen va-modell i formatet. Overflate som TIN er også støttet. Det er ikke støtte for volumobjekter og ikke funnet noe om topologi (node, kant, flate). Det er mulig å knytte egenskaper til objektene i formatet [44].

3.5.4 DXF

Drawing Interchange Format (DXF) er et format som brukes for å utveksle data fra AutoCAD til for eksempel GIS-programvare eller andre programmer. Formatet

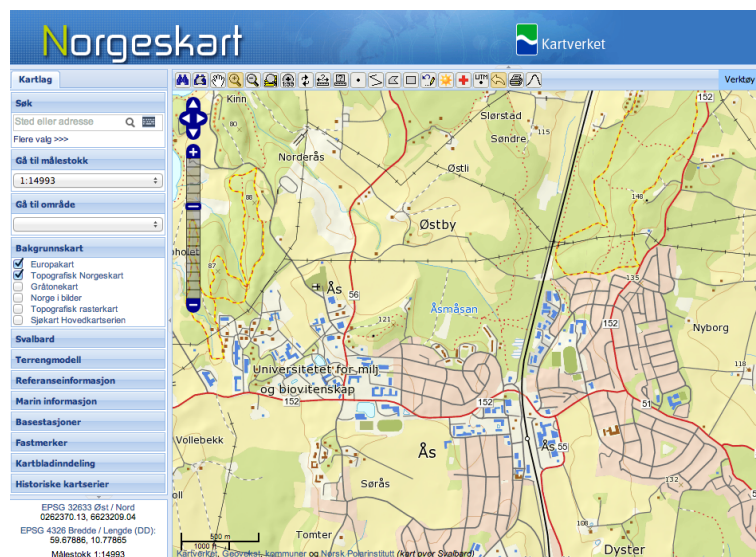
kommer i en tekst- og binærutgave, men tekst-utgaven (ASCII) er den mest utbredte. En .dxf-fil er en komplett representasjon av en tilsvarende AutoCAD-tegning (.dwg) [45].

3.6 Standarder for overføring av geografiske data

WMS og WFS er standarder for overføring av geografisk data over Internett fra OGC, Open Geospatial Consortium. Felles for disse standardene er at en nettleser henter og presenterer data. Disse standardene, særlig WMS, brukes i dag i karttjenester på Internett og mobile enheter. Programmet *GeoServer* kan brukes til å lage karttjenester basert på begge disse standardene [46]. Web 3D Service er en standard for overføring av 3D-modeller over nettet som foreløpig ikke er vedtatt av OGC. Formatet som er valgt for overføring er X3D [47]. WCS er en tilsvarende tjeneste for rom/tid-domene [48].

3.6.1 WMS

Web Map Service (WMS) produserer et statisk kart som et digitalt bilde i formater som JPEG, PNG eller GIF. Kartbildet lages av geografiske data. Det er begrenset støtte for egenskapsdata. WMS har mulighet for å endre koordinatsystemer på data som overføres. Siste versjonsnummer er 1.3 fra 2006 [49]. WMS er vanlig å bruke som bakgrunnskart bestående av flyfoto eller satellittbilder. Vektordata blir konvertert til et digitalt bilde. *Norgeskart* er Kartverket sin karttjeneste på nett og er vist i figur 3.18.



Figur 3.18: Skjermdump av Kartverkets karttjeneste Norgeskart som viser kartdata gjennom WMS-standard.

3.6.2 WFS

Web Feature Service (WFS) produserer kart som vektorer og har full egenskapsstøtte i motsetning til WMS. Projeksjonstøtten er tilsvarende som for WMS. WFS har mulighet for å gjennomføre spørre-operasjoner (eng: query operations) mot databasen og har mulighet til å låse data og gjennomføre transaksjoner for å kunne lage nye, endre eksisterende eller slette data i databasen. En spørring kan være å koble to (eng: join) datasett basert på egenskaper eller plassering til objektene (eng: spatial join). Siste versjon av WFS-standarden er 2.0 [50].

3.7 Datamodellering

Datamodellering kan defineres som “prosessen med å definere fenomener i den virkelige verden eller geografiske objekter som er interessante på bakgrunn av karakteristik og forhold mellom hverandre” [51, s. 425]. Modellering av den ustrukturerte verden foregår gjennom 3 trinn og ender opp i en strukturert modell av virkeligheten som er implementert i et databasesystem. De tre trinnene er:

1. Konseptuell modell/miniverden
2. Logisk modell/domenemodell
3. Fysisk modell/implementering

Den konseptuelle modellen er plattformuavhengig og beskriver hvem, hva, hvor og forhold mellom entiteter i den virkelige verden. Konseptuelle modeller gjelder for en begrenset del av virkeligheten, eksemplevis anleggsverdenen i denne oppgaven, og i noen sammenhenger kalles denne også for miniverden.

E/R-modellen, entitet/forhold (eng: entity/relationship), er en mye brukt modell i den konseptuelle verden for å modellere tabeller og forhold. UML er en mer brukt modell i dag [51, kap. 3.3] [52, kap. 7].

En logisk modell er i motsetning til den konseptuelle modellen ikke plattformuavhengig og det skyldes at denne modellen er knyttet til et bestemt databasesystem. Databasesystemene kan støtte forskjellige datatyper og den konseptuelle modellen må overføres til en logisk modell som er mulig å implementere i det valgte databasesystemet. En logisk modell kalles også for en domenemodell ettersom den gjelder innenfor et bestemt system, et domene.

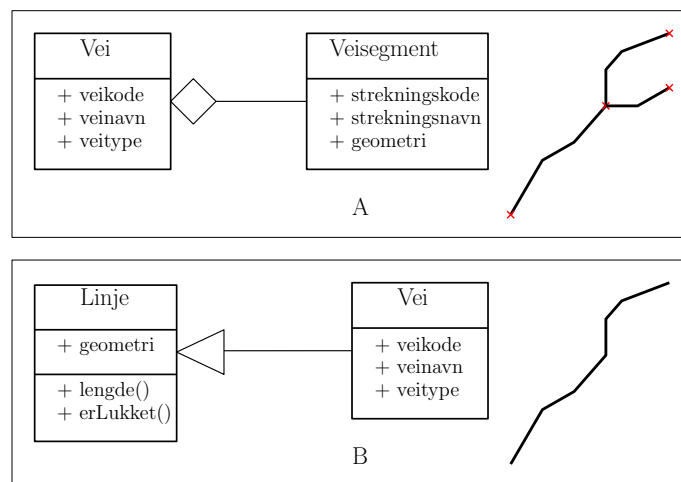
Den fysiske modellen er når den logiske modellen er implementert i en database og beskriver den faktiske datalagringen [51, kap. 3.3]. Se også figur 3.20 om oppbygging av et databasesystem i tre deler.

3.7.1 UML

Unified Modelling Language, UML, er et modelleringspråk som henter mye av tankegangen fra objektorientert tankegang med klasser, attributter, metoder, arv og forhold/assosiasjoner mellom klasser. UML kan brukes til å modellere mye mer enn bare klasser, som for eksempel prosesser, aktiviteter, roller og logiske komponenter. Klassediagram kan benyttes i datamodellering [53][54].

En klasse representerer noe eller noen, for eksempel en vei eller et veisegment som i figur 3.19.A. Forholdet mellom klassene i denne figuren er av typen *aggregering* (eng: aggregation) og brukes om en klasse som fungerer som en beholder for andre klasser. Andre forhold mellom klasser kan være *assosiasjon* (eng: association) eller *sammensetning* (eng: composition). Multiplisitet av forholdet mellom klassen er ikke angitt på figuren, men en vei kan tenkes å bestå av minimum en og maksimum uendelig veisegmenter, eller *1..** i UML-notasjon. Tilsvarende kan et veisegment tilhøre en eller flere veier hvis to eller flere veier går felles. [17, kap. 3][54].

Klassen *Vei* kan være en spesialisering av klassen *Linje* slik det er vist i figur 3.19.B. Klassen *Vei* arver alle egenskapene fra klassen *Linje* i tillegg til sine egenskaper og metodene som veikode, veinavn og veitype [52, kap. 7]. Metodene til klassen *Linje* er *lengde()* og *erLukket()*. + foran egenskapene betyr at denne egenskapen er offentlig (eng: public).

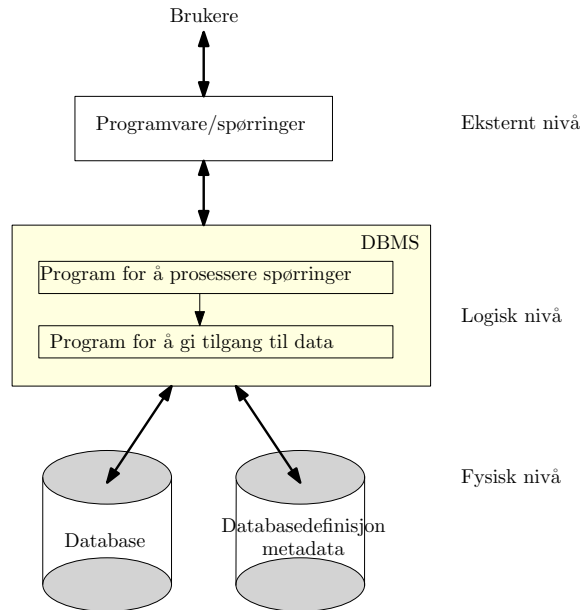


Figur 3.19: A) Klassene *vei* og *veisegment* med aggregering som forhold. B) Klassen *vei* som en spesialisering av en *linje*. Figur A basert på [17, fig 3.2].

3.8 Databasesystem

Et databasesystem består av en database og et databasehåndteringssystem, DBMS. Databasehåndteringssystemet (eng: database management system) håndterer brukertilgang, transaksjoner som å skrive, lese og endre data for flere samtidige brukere, indeksering av data og definering av databaser med egne tabeller og datatyper. *RDBMS*, relasjonell DBMS er den mest vanlige og utbredte varianten av DBMS i dag. *ORDMS* er en objektrelationell utgave og støtter abstrakte datatyper, klasser, arv og metoder på objektene som er lagret i en tabell [55][56].

En forenklet utgave av et databasesystem er gjengitt i figur 3.20 og beskriver tre nivåer av abstraksjon: 1) eksternt (program/bruker), 2) logisk og 3) fysisk. Det fysiske nivået er lagringsstrukturen, selve databasen. Det logiske nivået beskriver hvordan dataene er definert og bygget opp gjennom tabeller og forhold mellom tabeller. Et DBMS utgjør det logiske nivået. Det eksterne nivået er de faktiske



Figur 3.20: Databasesystem. Basert på figur 1.1 i [17].

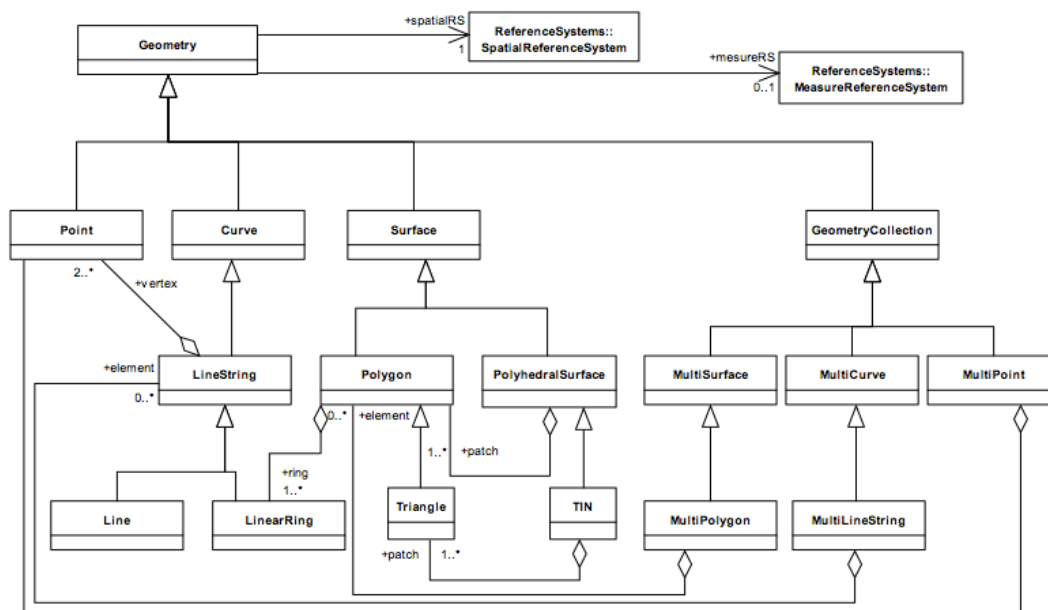
dataene en bruker eller program viser eller ser [17, kap. 1]. Begrepet database brukes ofte om hele databasesystemet.

Dagens geografiske databaser har begrenset støtte for 3D-geometri i form av volumobjekter. En nærmere gjennomgang av geometristøtten i PostGIS gjøres i kapittel 3.9. “Free form” linjer og overflater som for eksempel Bézier, splines og NURBS er det ikke innebygget støtte for i dag. Dette er mye brukt i CAD/BIM-verdenen og hvis det på et tidspunkt skal være mulig å lagre hele fagmodeller som benytter seg av en slik avansert geometri er det nødvendig med støtte for dette. Det er gjort en prototyp av en implementering av Bézier, B-spline og NURBS i *Oracle Spatial* som en egendefinert datatype [57][58]. Dette er så langt ikke innebygget i hverken PostGIS eller Oracle Spatial. Egendefinerte datatyper vil ikke kunne indekseres av DBMS og det vil kunne påvirke hastigheten ved spørringer [59][60].

3.9 PostgreSQL & PostGIS

PostgreSQL er et åpent kildekode objektrelasjonel databasehåndteringssystem som ble lansert i 1995 og er en avansert database med støtte for mange ulike datatyper. Egne datatyper og nye funksjoner kan også defineres [61].

PostGIS utvider PostgreSQL med støtte for geografiske objekter og kan bli brukt som en romlig database for geografiske informasjonssystemer (GIS). Versjon 2.0 av PostGIS ble lansert våren 2012 med støtte for raster (bilder), rasteroperasjoner og 3D-geometri (polyhedral surface og TIN) [62]. *SQL-MM Spatial* modell for topologi og topologiske operasjoner er også støttet gjennom 9-snittsmodellen. I PostGIS 2.0 har 9-snittsmodellen noe støtte for 3D-objekter (punkt og linje) [59, kap. 11, 13]. Topologi er behandlet i kapittel 3.3.



Figur 3.21: OpenGIS Simple Features. Figur hentet fra [29, fig. 1].

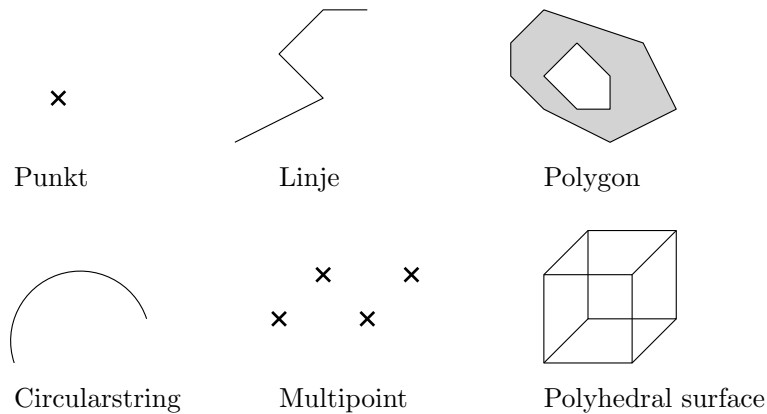
3.9.1 Geometristøtte (vektor)

PostGIS støtter OGC, Open Geospatial Consortium, *OpenGIS Simple Feature* standarden for geometri. Geometriklasser og forhold mellom klassene er vist i figur 3.21. *SQL-MM Spatial* utvider Simple Feature med interpolerte kurver og vi får følgende geometristøtte for vektordata i PostGIS 2.0 [59, kap. 4.1, 4.3]:

- Point
- Linestring
- Polygon
- Multipoint, multilinestring, multipolygon, multicurve, multisurface
- Geometrycollection
- Polyhedral surface
- TIN og triangel
- Circularstring
- Compoundcurve
- Curvepolygon

Noen av geometritypene i PostGIS er vist i figur 3.22. Multipunkt eller annen type *multi* er en samling av flere punkter, linjer eller polygoner på et objekt. En *circularstring* tilsvarer en *linestring* og består av 3 punkter; start, slutt og et vilkårlig

punkt på buen som legger grunnlaget for interpoleringen. *Compoundcurve* består av rette, lineære og kurvede segmenter. En triangulert overflate, TIN, består av flere triangler, definert av fire koordinater siden triangelet er lukket. En *polyhedral surface* er definert av polygonene som utgjøre flatene som avgrensner volumobjektet [59, kap. 4.1]. Et eksempel på en TIN bestående av to triangler og en *polyhedral surface* er gitt under.



Figur 3.22: Geometrityper i PostGIS: punkt, linje, polygon med hull, circularstring, multipunkt og polyhedral surface.

Eksempel på TIN og polyhedral surface:

```
TIN( ((0 0 0, 0 0 1, 0 1 0, 0 0 0)), ((0 0 0, 0 1 0, 1 1 0, 0 0 0)) )

POLYHEDRALSURFACE( ((0 0 0, 0 0 1, 0 1 1, 0 1 0, 0 0 0)),
  ((0 0 0, 0 1 0, 1 1 0, 1 0 0, 0 0 0)),
  ((0 0 0, 1 0 0, 1 0 1, 0 0 1, 0 0 0)),
  ((1 1 0, 1 1 1, 1 0 1, 1 0 0, 1 1 0)),
  ((0 1 0, 0 1 1, 1 1 1, 1 1 0, 0 1 0)),
  ((0 0 1, 1 0 1, 1 1 1, 0 1 1, 0 0 1)) )
```

PostGIS støtter en tabell med flere geometrikolonner hvor hver av disse kan ha samme eller forskjellig geometritype. Når geometrikolonnen opprettes kan geometritypen defineres eksplisitt til å være punkt, eller bare *geometry*. I førstnevnte kolonne kan det kun lagres punkt, men i sistnevnte kan alle geometrityper lagres. *Geometrycollection* kan også velges som geometritype og kan bestå av flere typer geometri.

3.9.2 Koordinatsystemer

PostGIS inneholder informasjon til over 3000 ulike koordinatsystemer fra koordinatsystembiblioteket *proj4* [59, kap. 4.3]. Hvert enkelt koordinatsystem har en SRID som er et tall som identifiserer koordinatsystemet. Se kapittel 3.2 om SRID. Prosjektet i Dronning Eufemias gate bruker EUREF89/UTM Sone 32 og har SRID (EPSG) 25832.

Et enkelt koordinat i PostGIS kan være av typen *geometry* eller *geography* som tar utgangspunkt i henholdsvis et kartplan eller en kule (lengde- og breddegrad).

Førstnevnte type kan være alle typer geodetiske koordinater med unntak av lengde- og breddegrad, som går inn under *geography*. Det er mulig å transformere mellom ulike koordinatsystemer med funksjonen `ST_Transform` i PostGIS [59, kap. 4.3, 8.5]. Det er også støtte for koordinater som ikke er knyttet til Jorda. I eksemplene på *TIN* og *polyhedral surface* over vises slike lokale koordinater. Se kapittel 3.2 for en gjennomgang av de ulike koordinatsystemene. Koordinatene kan være 2-dimensjonale (X, Y eller lengde- og breddegrad), 3-dimensjonale (X, Y, Z) eller 4-dimensjonale (X, Y, Z, M) [59, kap. 4.2]. *M* brukes for en måleverdi og kan for eksempel være et tidspunkt eller posisjonskvalitet i form av standardavvik på målingen.

3.9.3 Romlige spørringer

GiST, “Generalized Search Tree”, er en indekseringsmetode som kan benyttes på geografisk data og som benyttes i PostGIS for å øke hastigheten på spørringer. Data som indekseres med GiST deles opp i “things to one side, things which overlap, things which are inside” [59, kap. 4.6]. I prinsippet er dette en minste avgrensende boks rundt geometrien (eng: minimum bounding box). En tenkt spørring i spørrespråket SQL som er vist under benytter seg av GiST for å finne alle innmålinger som er 15m fra vannledningen med *id 1* og sortert på avstand:

```
select * from innmåling i, va_ledning v
where ST_DWithin (i.geom, v.geom, 15) and v.id = 1
order by ST_Distance(i.geom, v.geom) asc;
```

GiST benyttes også i spørringer for å undersøke om to geometrier krysser, overlapper, møter eller har noe felles. Den minste avgrensningsboksen til objektet brukes først for å sjekke om objektene kan ha noe til felles før det gjøres en nærmere sjekk på selve geometrien hvor disse overlapper eller krysser. Dette er en del av 9-snittsmodellen og er behandlet i kapittel 3.3 om topologi. Andre romlige funksjoner kan være for å beregne areal, lengde, omkrets, lage buffer og kutte overlappende geometri mot hverandre, slå sammen eller splitte geometri. En samling romlige funksjoner er [59]:

`ST_Crosses`, `ST_Overlaps`, `ST_Touches`, `ST_Intersects`, `ST_Intersection`, `ST_Buffer`, `ST_Area`, `ST_Length`, `ST_Perimeter`, `ST_Transform`

3.9.4 Formatstøtte

PostGIS har god innebygget støtte for ulike formater som den kan lese og skrive til og fra. Innebygget støtte for formater PostGIS leser er:

- WKT, EWKT
- WKB, EWKB
- KML
- GML

- GeoJSON

Formater PostGIS kan skrive til inkluderer de nevnte formatene og i tillegg X3D og SVG. WKT og WKB (eng: Well-Known Text/Binary) er en del av *OpenGIS Simple Feature* spesifikasjonen [29]. EWKT og EWKB er PostGIS sin utvidelse med SRID og støtte for 4-dimensjonale koordinater. I eksemplene under er en linje bestående av 3 punkter i WGS84 lengde- og breddegrad (SRID 4326) vist i de enkelte geometritypene:

WKT:

```
LINestring(10.776 59.665,10.777 59.666,10.779 59.672)
```

EKWT:

```
SRID=4326;LINestring(10.776 59.665,10.777 59.666,10.779 59.672)
```

EWKB:

```
\x0102000020e6100000030000005a643bdf4f8d254085eb51b81ed54d40e7fba  
9f1d28d25406891ed7c3fd54d40022b8716d98e2540bc74931804d64d40
```

GML:

```
<gml:LineString srsName="EPSG:4326">  
<gml:coordinates>10.776,59.6649  
10.7769,59.6659 10.779,59.6719  
</gml:coordinates>  
</gml:LineString>
```

KML:

```
<LineString>  
<coordinates>10.776,59.6649  
10.7769,59.6659 10.779,59.6719  
</coordinates>  
</LineString>
```

X3D:

```
<LineSet vertexCount='3'>  
<Coordinate point='10.776 59.6649  
10.7769 59.6659 10.779 59.6719' />  
</LineSet>
```

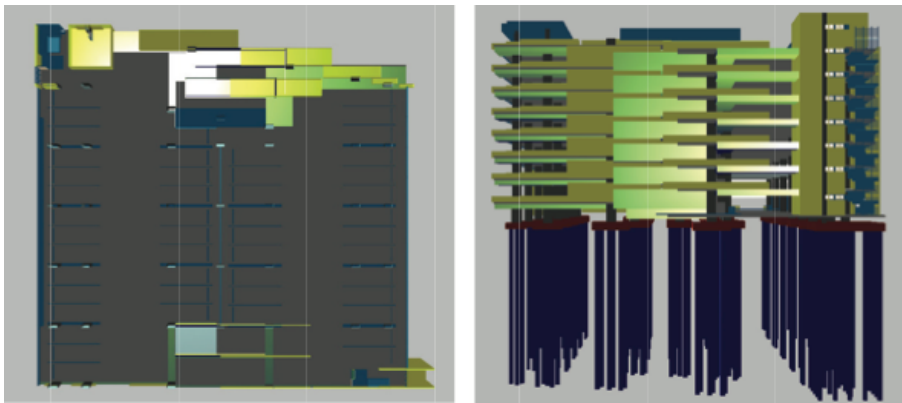
GeoJSON:

```
{"type": "LineString", "coordinates": [[10.776, 59.6649],  
[10.7769, 59.6659], [10.779, 59.6719]]}
```

Det er per i dag kun eksperimentell støtte for visualisering av PostGIS sitt interne format for volumobjekter gjennom *QGIS Globe plugin* eller *PostGIS SFCGAL* [63]. For visualisering av 3D-geometri kan *Blender* brukes for å vise geometrien som er eksportert til et X3D-dokument. 2D-data kan vises i de fleste GIS-programmer. Inneholder dataene Z-koordinat vises kun X- og Y-koordinatene i GIS-programvaren.

3.10 Tidligere arbeid

Idéen med å benytte seg av en romlig database for å håndtere BIM-data er testet ut og beskrevet i artikkelen “Integration of Java-based BIM with spatial database” [64]. Et parkeringshus med 8 etasjer ble modellert og lagret i PostGIS. *Java 3D API* ble brukt for å generere 3D-modellen som var lagret i databasen og er vist i figur 3.23. En romlig database er også benyttet i “Berlin 3D City model” hvor digitale terrengmodeller, matrikkeldata for eiendommer, flyfoto og 3D-modeller av de fleste bygninger i Berlin er laget i CityGML og importert i databasen [65][66]. Databasen kan eksportere 3D-modellene til KML/Collada for visualisering i *Google Earth* og all programvare utviklet for import, eksport og skript til databasen er åpen kildekode [67].



Figur 3.23: 3D-modell laget fra databasen. Sett i plan til venstre og fra side i høyre. Delfigur B og D fra [64, figur 4].

Kapittel 4

Studie av Dronning Eufemias gate

Det er flere store pågående infrastrukturprosjekter som jobber etter Håndbok 138, deriblant Dronning Eufemias gate. Dette prosjektet er valgt ut for å studeres i detalj hvordan modeller etter kravene i Håndbok 138 anvendes i et infrastrukturprosjekt.

Skanska ble tildelt utbyggingen av dette prosjektet av Statens Vegvesen i august 2011 og prosjektet inkluderer i tillegg til selve Dronning Eufemias gate, også Kong Håkon 5.s gate, deler av Langkaigata og ombygging av deler av Nylandsveien bru. Prosjektet har en kostnad på ca 585 mill. NOK eks. mva. Prosjektet hadde oppstart høsten 2011 og forventes å være ferdig sommeren 2015 [68][69]. Byggherren på prosjektet er Statens Vegvesen Region Øst (SVRØ) og hovedentreprenør er Skanska.

Dette prosjektet startet opp før den endelige utgaven av Håndbok 138 ble publisert og det er noe som gjøres forskjellig fra hva som er spesifisert i håndboka. Når det gjelder dataleveranser av modeller er det spesifisert originalformat + LandXML som et åpent format i EUREF89/NTM-prosjeksjonen (se kapittel 2.3.1). På dette prosjektet benyttes kun originalformatet (DWG) og EUREF89/UTM-prosjeksjonen i fagmodellene som er relevante for denne oppgaven (VA).

I henhold til oppgavens metode beskrevet i kapittel 1.2 vil jeg undersøke dataflyt fra modeller til stikningsdata og “som utført” dokumentasjon. Overordnede spørsmål er som følger:

- Hvordan benyttes modeller i anbuds- og planleggingsfasen i et prosjekt? (kapittel 4.1). Dette kapitlet behandler generell bruk av modeller i planlegging og er ikke knyttet til et spesifikt prosjekt.
- Hvordan benyttes modeller i produksjon og for til å lage stikningsdata? Hvordan håndteres revideringer? (kapittel 4.2)
- Hvordan brukes modeller og stikningsdata til å gjennomføre geometrisk kontroll og å lage dokumentasjon? (kapittel 4.3)
- Hvordan benytter byggherren seg av innmålinger til å lage “som utført” modell til bruk i forvaltning og drift? (kapittel 4.4)

Det er gjennomført intervjuer med de stikningsansvarlige fra entreprenør, og jeg har deltatt på et stikningsmøte med representanter fra byggherren, Statens Vegvesen

og konsulentfirmaet på prosjektet, Aas-Jacobsen. Intervjuet med de stikningsansvarlige er gjengitt i tillegg A. Mye erfaringer med bruk av modellen i produksjon og dokumentasjon er gjort gjennom feltarbeid som en del av sommerjobb på prosjektet i 2012.

4.1 Planlegging

Modeller brukes i prosjektering og planlegging av store og komplekse infrastrukturprosjekter. Det er vanlig at fagansvarlige for den enkelte fagmodell og byggherren møtes jevnlig (typisk hver 14. dag) for å gå igjennom siste oppdaterte modell og diskutere konflikter/mangler før nye endringer gjøres i modellen. Modellene gir en bedre oversikt enn vanlige 2D papirtegninger og bidrar til bedre framdriftsplaner for prosjektene. I kalkulasjonsprosessen (i forkant av anbudsinnlevering) fungerer modellene som et hjelpemiddel for å se hvordan de store grepene som for eksempel masseforflytning må gjøres. Dette gjør at prising på prosjekter i anbudene blir mer presis. Modellene brukes også på planleggings- og stikningsmøter underveis i et prosjekt. Modellene kan vises på storskjerm og det er lettere å diskutere endringer i modell og utfordringer dette kan få byggeprosessen videre [70][71].

Det er benyttet 3D-modeller i arbeidet med å utvikle Bjørvika i Oslo helt siden 1998. Denne modellen er bruk i planprosessen for å vise fram de ulike forslagene på bygg og plassering, sol- og skyggestudier og til formidling i media, til naboer og i reguleringsprosessen. Det nye veisystemet i Bjørvika inkludert Operatunnelen og Dronning Eufemias gate er modellert i 3D [72].

4.2 Produksjon

Produksjonsfasen i et prosjekt er fra når selve byggingen starter og fram til ferdigstillelse og overlevering av prosjektet.

Autodesk Buzzsaw er samarbeidsløsningen på prosjektet og er et webhotell for all kommunikasjon mellom byggherre, konsulent og entreprenør. Modeller, revideringer, arbeidstegninger og møtedokumenter leveres av byggherre og konsulent her. Entreprenøren leverer innmålinger, geometrisk kontrollrapport, målebrev og annen dokumentasjon gjennom den samme løsningen. Versjonshistorikk på alle filer (på filnivå) trekkes fram som et pluss i denne løsningen og gir en oversiktelig og god løsning for dataflyt i dag.

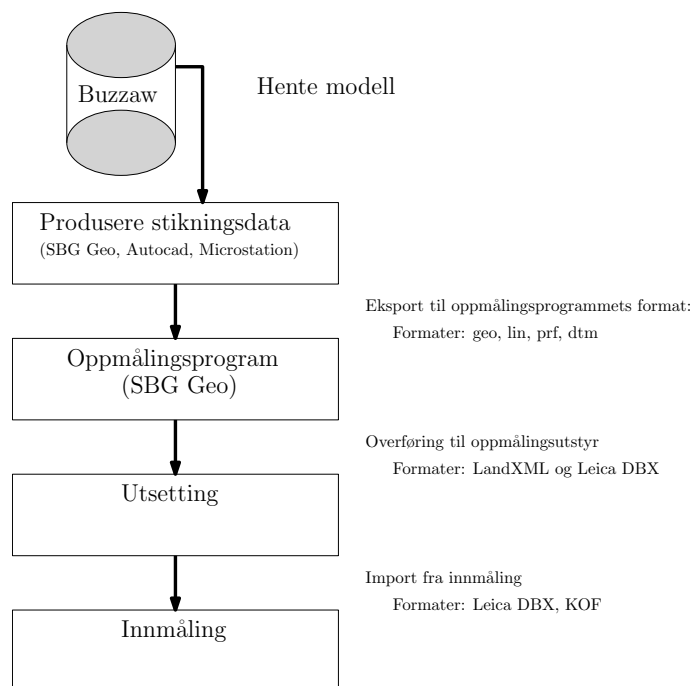
Data som leveres fra entreprenør til bruk i produksjon er 3D-modeller på AutoCAD DWG-format og arbeidstegninger på DWF-format for utskrift på papir. Arbeidstegningene genereres fra 3D-modellen av konsulenten, men oppdateres ikke automatisk hvis det gjøres endringer i 3D-modellen. Modellen inneholder all nødvendig geometri for å lage stikningsdata i dag, enten som egne lag eller ved å eksportere nødvendig geometri fra modellen. For å se alt som skal bygges må modellen og arbeidstegningene brukes sammen da ikke alle detaljer finnes på 3D-modellen og noe er bare på arbeidstegningene. For eksempel er boltegrupper til lysmast i 3D-modellen for belsysning tatt med, mens jordledning til samme mast er å finne under arbeidstegningen.

4.2.1 Stikningsdata

Produksjon av stikningsdata fra en modell foregår i dag med to metoder:

1. Eksport av geometri for stikning direkte fra eget lag (stikningsgeometri)
2. Bearbeide overflategeometri for å lage stikningsdata

En del modeller, som for eksempel fagmodellen for VA, er bygget opp med egne lag for stikningsgeometri. Oppmålingsprogrammet *SBG Geo* som Skanska benytter seg av kan åpne .dwg-filer og hvert lag skilles ut som en separat fil i programmet. Volumobjekter blir ikke med og en sitter igjen med enkel geometri som linjer og punkt. For VA-modellen sitter en igjen med stikningslinjer og punkt for rør og kummer. Stikningslinjene deles i *.lin-* og *.prf-filer* for henholdsvis horisontal og vertikal geometri før det blir konvertert til oppmålingsutstyrets format (Leica DBX). Dette er et eksempel på første metode. Modeller med egne lag for stikningsgeometri framheves som bedre av de stikningsansvarlige da dette er mindre arbeidskrevende enn å plukke punkter og linjer fra en overflate og bearbeide disse.



Figur 4.1: Dataflyt under produksjon. Fra modell til produksjon av stikningsdata og overføring til oppmålingsutstyr.

Den andre metoden for produksjon av stikningsdata krever en del erfaring og riktig programvare. Her inngår overflater (terrengmodeller) og krumme linjer. Linjene kan også komme fra eget lag som i den første metoden. Triangulerte terrengmodeller brukes for å grave dypt nok for konstruksjoner og under støping for at betongkonstruksjonen ikke blir for høy. For å lage en triangulert flate for betongkonstruksjoner må solids (volumobjektet) splittes opp i de enkelte linjer som avgrensner volumobjektet og linjene danner så utgangspunkt som knekklinjer i terrengmodellen. Det lages

terrengmodeller for overkant og underkant av betongkonstruksjonen slik at linjene som tilhører enten overkant eller underkant må plukkes ut. Hvis det er utsparringer i betongkonstruksjonen, dvs to punkter med samme koordinat i grunnriss, men med ulik Z-verdi (høyde) vil trianguleringen feile. Alle disse punktene må da flyttes manuelt noen mm i grunnriss og det er en veldig tidkrevende prosess. For konstruksjoner med en krum linje som avgrenser overflaten må linjen deles opp i små segmenter for at triangelmodellen skal følge den krummede overflaten godt. Punkter og linjer fra modellen blir triangulert, se også figur 3.3.A om triangulering. Det er ønske om to nye lag som en del av modellgrunnlaget, over- og underkant triangelmodell for betongkonstruksjoner.

Oppmålings og maskinstyringsutstyret har begrenset geometristøtte. Punkter, rette linjer og triangulerte terrengmodeller er godt støttet. Det er noe støtte for krumme linjer med en gitt radius, men dette brukes i liten grad. Det betyr at krumme linjer fra modellen må behandles slik det gjøres i dag og ikke kan overføres direkte til oppmålingsutstyret. Oppfatningen er at oppmålingsutstyret henger litt etter på geometristøtten sammenlignet med modellene i dag.

Stikningsansvarlige på prosjektet opplever i dag modellene som tilstrekkelige for å produsere stikningsdata ut i fra geometri i fagmodellene, men at det i mange tilfeller ligger mye arbeid bak. Egenskapene i lagnavnet i modellen hjelper på lang vei til å finne fram til riktig objekter i modellen som det skal lages stikningsdata på.

Ferdig stikningsdata lagres i en katalogstruktur (mapper og filer) på prosjektets egen server (ikke Buzzsaw) etter delprosjekt og modelltype.

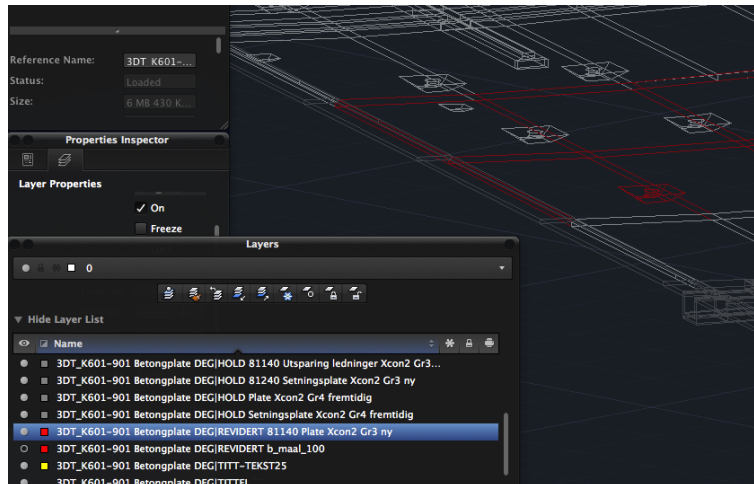
4.2.2 Revidering av modell

Revideringer i modellen farges i rødt og legges i et eget lag i modellen. Ofte er enkeltgeometri (linjer) gruppert sammen til en solid (volumobjekt) og det er da ikke lett å finne ut hvilken linje som er endret, eller om det er hele gruppen. Det er ikke lagret noen historikk til gruppen (av geometri) over endringene som er gjort på den enkelte geometri. Det er en manuell prosess å gjøre om og huske på å gi ny farge til geometrien, slik at det er en mulighet til å glemme å gjøre dette. I praksis lages all stikningsdata på nytt for den aktuelle fagmodellen. Eksempel på revidert geometri som er farget rødt i eget lag er vist i figur 4.2 og markert i figuren.

4.3 Dokumentasjon

Dokumentasjonsfasen er innmålinger som er gjort underveis i byggefasen av prosjektet for å dokumentere at det som bygges er innenfor toleransegrensene, “som utført” dokumentasjon over hva som er bygget og målebrev. Dette danner grunnlag for oppgjøret i prosjektet.

Innmålinger gjøres av stikkeren ute på byggeplassen og det er opp til stikkeren om det som måles inn kodes etter prosjektets objektkodeliste i felt eller inne på kontoret i etterkant. Innmålinger som leveres til byggherren må være kodet med temakode. Punkt/linjenavnet forteller også hva som er målt inn i felt. Eksempel på punktnavn kan være *prof-asf8cm001*. Når innmålingen for dagen er gjort og



Figur 4.2: Revidering i modell som eget lag kodet i rødt. Skjermdump fra AutoCAD.

importert til oppmålingsprogrammet deles oppmålingene etter hva som er målt inn. En oppdeling kan være:

- innm-ok-deg-plate24-20120812-ER
- innm-ok-rør400mm-deg-20120812-ER
- eks-ok-rør250mm-deg-20120813-OL

Dette er henholdsvis innmåling overkant betongplate og overkant rør med dimensjon 400 mm som er gjort den 12.08.2012 av stikkeren ER. I dag dekker dette behovet for enkelt å finne igjen hvem (eks. ER), hva (eks. ok-plate) og når innmålingen er gjort. Navngiving av filen er ikke tilfeldig da initialene til stikkeren benyttes til slutt, datoformat av typen ååååmmdd og fritekst. *Innm* beskriver innmåling av noe som er bygget og *eks* står for eksisterende. Det kan være rør som blir avdekket under graving og for eksempel skal fjernes. Enkeltfilene lagres så i en katalogstruktur hos den enkelte stikkeren på den lokale serveren. Innmålingene benyttes videre i dokumentasjonsarbeidet.

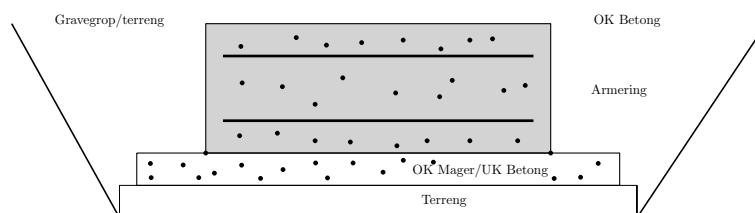
I etterkant blir temakode påført innmålingene av de stikningsansvarlige som gjennomfører geometrisk rapportering dersom stikkeren ikke allerede har kodet innmålingene. Dette er videre beskrevet i kapittel 4.3.1. Et utsnitt av temakode-lista på prosjektet for vann og avløp er vist i figur 4.3 og inneholder temakode, temanavn, datatype på innmålt geometri og eventuell forklaring.

Fra modellen over betongplata som er fundamentert på peler i Dronning Eufemias Gate lages det to terrengmodeller, underkant (UK) og overkant (OK) plate. Innmålinger av byggegrop og masser som legges på (terreng), samt mager kontrollers mot UK terrengmodell for å sjekke at det er gravet dypt nok for plata før støp. Mengden masse som er gravd ut beregnes. Mager er et 5cm tynt lag med betong som legges før støp for å få en jevn overflate å støpe på. Armeringsjern i plata måles inn for å sjekke at armeringen får tykk nok overdekking av betong. OK terrengmodell benyttes under støp og ferdig betongoverflate måles inn og sjekkes mot terrengmodellen. De enkelte lagene for en betongkonstruksjon er vist i figur 4.4.

Temakode	Gruppe	Temanavn	Datatype	Forklaringer
42000	Tekniske anlegg VA/Dren			
42010	Tekniske anlegg VA/Dren	Innmålt bunn innvendig kum	Volum	
42011	Tekniske anlegg VA/Dren	Sandfang, bunn innvendig	Punkt	
42012	Tekniske anlegg VA/Dren	Minisandfang, bunn innvendig	Punkt	
42013	Tekniske anlegg VA/Dren	Hjelpesluk, bunn innvendig	Punkt	
42020	Tekniske anlegg VA/Dren	Markeringskum	Volum	
42021	Tekniske anlegg VA/Dren	Markeringskum, topp utv ledn	Punkt	
42022	Tekniske anlegg VA/Dren	Markeringskum, topp lokk	Punkt	
42025	Tekniske anlegg VA/Dren	Markeringskum, Senterlinje	Linje	
42100	Tekniske anlegg VA/Dren	Sandfangskum	Volum	
42101	Tekniske anlegg VA/Dren	Sandfangskum, bunn utvendig	Punkt	Målt sentrisk kum
42102	Tekniske anlegg VA/Dren	Sandfangskum, topp lokk	Punkt	Målt sentrisk kum
42103	Tekniske anlegg VA/Dren	Sandfangskum, senter rist	Punkt	Senter slukrist/lokk.
42105	Tekniske anlegg VA/Dren	Sandfangskum, Senterlinje kum	Linje	

Figur 4.3: Utdrag fra temkodeliste for VA på prosjektet.

Innmålinger av rør består av en linje fra start til slutt rør. Avvik fra innmålingen til prosjektert linje måles fra enkeltpunktene som utgjør innmålt linje. Innmålinger av en kum og kumlokk består av nok punkter til å lage en sirkel (minimum 3) i oppmålingsprogrammet slik at senterpunkt kumlokk og kum bunn kan beregnes. Dette måles ofte også inn ute. For gravegroppen og terrenget rundt kummen og rør profileres dette og masseuttaket beregnes.



Figur 4.4: Forenklet oppbygging av betongkonstruksjon, mager og gravegrop.

4.3.1 Geometrisk kontrollrapport

Geometrisk kontrollrapport, ofte kalt grap, er en kontroll entreprenøren utfører for å “dokumentere at kontraktens geometriske kvalitetskrav for alle objekter under bygging er oppfylt” [73, kap. 4]. Differansen mellom innmålt og prosjektert objekt regnes ut og innmålinger som kommer utenfor objektets toleransekrav skal det lages avviksmelding på.

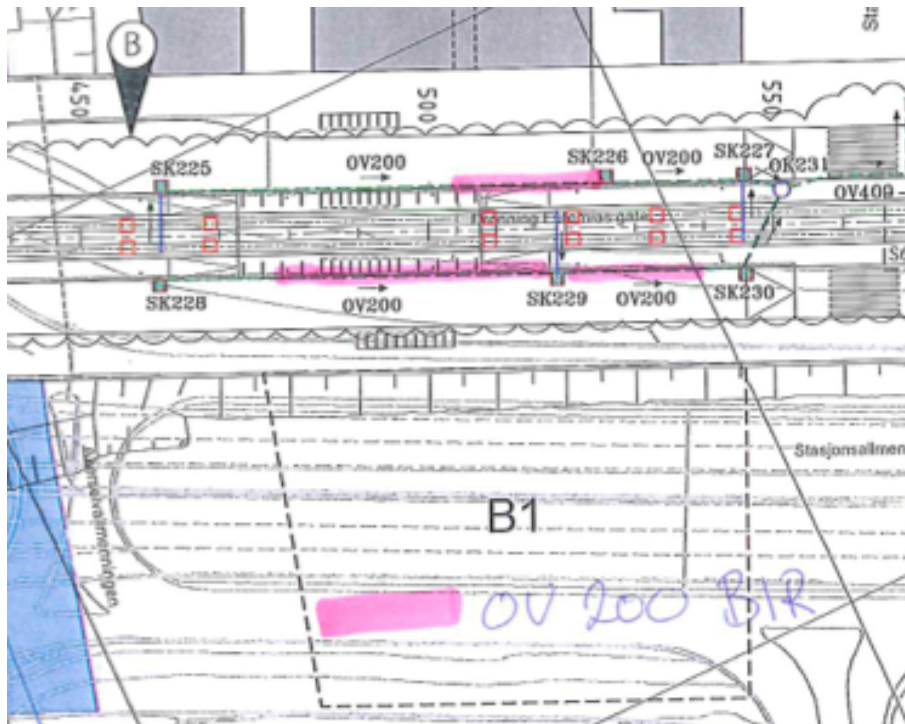
En utfordring i dag ved produksjon av dokumentasjon er at Statens Vegvesen kun ønsker innmålinger på for eksempel plate 24 og ikke noe mer rundt. Innmålinger som havner utenfor angitt område må manuelt filtreres bort før beregning av avvik mellom teoretisk og bygget gjennomføres.

Et annet eksempel er at det skal gjennomføres geometrisk kontroll på de enkelte rørstrekker som legges, hvis røret legges i flere omganger. Det kan være at 10 m av et rør legges og godkjennes før det legges 10 m til. Da er det kun innmålingen av det nye rørstrekket som er interessant, ikke en eventuell ny innmåling av hele røret. Dette setter krav til å huske hva som er målt inn og godkjent tidligere for å unngå

duplikater av innleverte data. I dag tegnes det med tusj på arbeidstegninger i papir over hva som er innlevert geometrisk rapport på. Det skal gjøres forsøk å bruke *.dwf-format*, samme format som arbeidstegningene, og legge inn kommentarer over filnavn til innleveringer og digital tusj for å merke hvor på arbeidstegningene innleveringen gjelder. En slik oversikt er vist i figur 4.5 for en overvannsledning med dimensjon 200 mm (OV 200) for bunn innvendig rør (BIR). Oversiktene får logisk navn som: 20121116_G604 42235 Dren OV200 oversikt GRAP_1.pdf. Innmålingen som er i denne oversikten leveres i kof-format med filnavnet:

20121116_3DT_G691 42235 Dren OV200 bunn innv GRAP xyz_1.kof.

Et regneark for de enkelte fagmodellene brukes for en oversikt over navn på innmålings- og oversiktsfilene, navn på avviksmelding og godkjenningsdato. En slik oversikt er vist i figur 4.6. En avviksmelding inneholder en forklaring på hva som er avviket, størrelsen på feilen i cm, mm etc og hvilke tiltak som må gjøres eller blitt gjort.



Figur 4.5: Oversikt over innmålinger i geometrisk rapport for OV 200 BIR på arbeidstegning. Skjermdump.

SBG Geo er programvaren som benyttes til å beregne differanser og dette programmet oppleves som godt nok av de stikningsansvarlige for å lage dokumentasjonen i dag. Et ønske for å gjøre arbeidet lettere og mer effektivt er bedre muligheter til å filtrere og velge punkter eller linjer innenfor et område som for eksempel bare innenfor omrisset til plate 24 eller som ikke er godkjent i geometrisk kontroll av byggherre enda. Muligheter for å beregne utgravet volum av en kabelkanal eller rør kun basert på innmålinger av selve røret eller kabelkanalen er et ønske fra de stikningsansvarlige. Det er spesifiserte grøftestørrelser rundt en kabelkanal eller rør i Statens Vegvesens Håndbok 18 og toleranser på rør, grøfter og kummer og som

Loggskjema geometrisk kontrollrapport				
Prosjekt:	E18 Bjørvika etappe 2	Fargekoder entreprenør	Fargekoder byggherre	
Entrepriise:	e06 01 Dronning Eufermiasgate	Fagmodell(kontroll komplett, dato)	Godkjent(ferdig behandlet (dato)	
Entreprenør:	Skanska Norge AS	Fagmodell (under behandling)	Avvist (dato)	
Fagmodell:	3DT_H692 Fjernvarme			
Geometrisk kvalitetskontroll av objekter etter logisk oppdeling				
fnavn / refe	Filnavn-grafisk oversikt	Filnavn-XYZ dokumentasjon	Filnavn-avviksrapport	Byggherre godkjent
FV400-560	H672_Fjernvarme oversikt GRAP.dwf	20130403_3DT_H692_43101 FV400-560 GRAP xyz.kof	Avvik 079 DEG Avvik på fjernvarme og kjørerør, høyde og plassering	
FV450-560	H672_Fjernvarme oversikt GRAP.dwf	20130403_3DT_H692_43104 FV450-560 GRAP xyz.kof	Avvik 079 DEG Avvik på fjernvarme og kjørerør, høyde og plassering	

Figur 4.6: Utsnitt av loggskjema for geometrisk rapport. Navn på innmålingsfiler, oversiktsfiler og avviksfiler. Skjermdump.

sammen med innmålingen kanskje kan disse spesifiseringene og innmålingen av røret brukes til automatisk beregning av masse som er gravet ut og fylt igjen i grøften til røret [74].

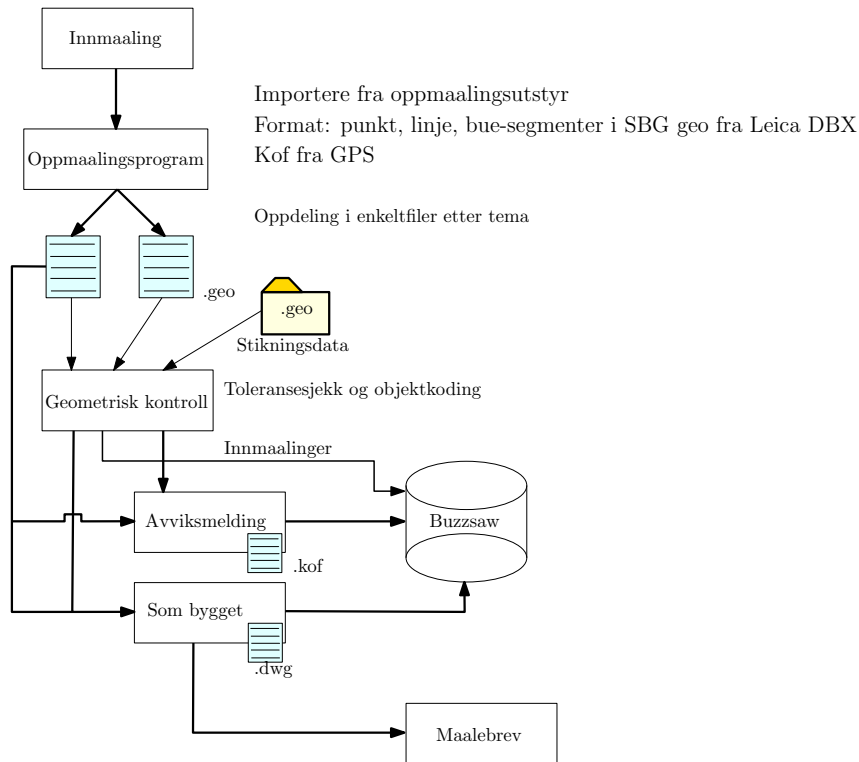
Innmålinger fra grap benyttes etter at de er godkjent av byggherre til å lage “som bygget” innmålinger og lagres i .dwg-formatet av entreprenør. “Som bygget” brukes av de stikningsansvarlige om denne dokumentasjonen framfor “som utført”. I dokumentet “Krav til digitale leveranser i gjennomføringsfasen” [73] brukes “som bygget”. Her settes alle innmålinger av ett rør sammen til en linje (jamfør med utfordringen med rør i forrige avsnitt) og kodes med riktig temakode. Filnavnet på innlevert data blir SB (“som bygget”) + objektkode og dato. “Som bygget” innmålinger leveres også til byggherre gjennom *Buzzsaw*. Entreprenøren bruker “som bygget” innmålinger som grunnlag for dokumentasjon til målebrev opp mot kontraktsposter for fakturering. Dette håndteres i dag i andre programmer.

4.4 Etterbruk & “som utført” modell

“Som utført” modell representerer det som er bygget i prosjektet, jamfør med kapittel 2.3.1. “Som bygget” innmålinger fra entreprenøren danner grunnlaget for denne modellen som lages av konsulenten i prosjektet. Er innmålingene innenfor toleransekravet gjøres ingen endringer og statusen på objektet oppdateres til “som utført”. Hvis innmålingene er utenfor angitt toleranse eller objektet måtte flyttes under bygging vil objektet bli modellert opp på nytt basert på innmålingene etter at avviket er godkjent av byggherre [73, kap. 5]. I praksis er “som utført” modell en separat .dwg-fil hvor prosjektet geometri kopieres inn og eventuelt modelleres opp på nytt.

“Som utført” modell danner grunnlag for eksport til FDV systemer som *Nasjonale Vegdatabank* (NVDB). Datakatalogen i NVDB bruker ikke volumobjekter slik at det kun er plassering i form av et punkt, linje eller flate (polygon) til objektet som er interessant [75]. Dette gjelder også for vann og avløp, se SOSI ledning 4.5 i kapittel 3.4.3. Eksport av nødvendig geometri til disse systemene er en ganske enkel, men omfattende oppgave.

All dokumentasjon lagres digitalt på *Buzzsaw* enten det er innmålinger, avviksmeldinger/oversikter, annen dokumentasjon og kommunikasjon mellom byggherren og entreprenøren. Oversikter lages i regneark med navn på de andre filene. Med andre ord er dokumentasjonsprosessen manuell. For å lage en oppdatert “som utført” modell må det først sjekkes i oversiktsfiler for geometrisk rapport og avviksmeldinger



Figur 4.7: Dataflyt fra innmåling og geometrisk kontroll til som-bygget dokumentasjon og målebrev.

før geometri som er innenfor toleranser eller godkjent av byggherre slik at lagnavnet kan oppdateres til “som utført”.

I AutoCAD er geometrien objektet og flere objekter kan grupperes sammen til et lag og egenskaper til det grupperte objektet kan lagres i lagnavnet. Se kapittel 3.5.2 for en beskrivelse av .dwg-formatet. På stikningsmøtet kom det fram at fordeler med å ha status på objekter som kan oppdateres underveis mens de blir bygget, revidert eller godkjent kan være nyttig. I dag blir “som utført” modell og prosjektert modell to separate modeller med mye av den samme geometrien.

Kapittel 5

Prototype på system & resultater

Prototypen på systemet bruker en geografisk database som har støtte for geometri for å lagre modell- og innmålingsdata og benyttter dette sammen til å gjennomføre geometrisk kontroll og lage “som utført” dokumentasjon. I henhold til oppgavens metode i kapittel 1.2 skal det utvikles følgende:

- Kravspesifikasjon for (kapittel 5.1):
 - Fagmodell vann og avløp
 - Innmålingsdata
 - Funksjonalitet til system
- Konseptuelle og logiske datamodeller (kapittel 5.2)
- Prototype på system (kapittel 5.3):
 - Program for å lese modell- og innmålingsdata
 - Program for å gjennomføre geometrisk kontroll

5.1 Kravspesifikasjon

Krav til geometri og egenskaper i fagmodellen er gitt av Håndbok 138 [1]. I prototypen brukes en begrenset fagmodell for vann og avløp avgrenset til kum og ledning. Fagmodellen skal i utgangspunktet dekke entreprenørens behov under planlegging-, produksjons- og dokumentasjonsfasen i et prosjekt. Ettersom modellene skal danne utgangspunkt for FDV-systemer i etterkant tas dette også med i betraktning. *SOSI ledning 4.5* i kapittel 3.4.3 danner utgangspunktet for modellering av fagmodellen.

For innmålingsdata tas det utgangspunkt i filer som inneholder eksisterende innmålingsdata, men også krav til dokumentasjon som er gitt i Håndbok 138 [1] og dokumentet “Krav til digitale leveranser i gjennomføringsfasen” [73] som beskriver krav til innmålinger, navngivning på filer og gjennomføring av geometrisk kontroll.

I kapittel 4 ble det av det stikningsansvarlige i Dronning Eufemias gate pekt på at det er utfordringer knyttet til revideringer av modell, unødvendig bruk av tid til å filtrere vekk innmålinger som er utenfor området (eksempel plate 24) som ikke skal

inngå i geometrisk kontroll, eller innmålinger som er godkjent av byggherre tidligere (eksempel rør som legges stykkevis). Det er også ønskelig å kunne beregne volum på en grøft ut i fra innmåling av røret som er lagt.

Fra kravspesifikasjonen lages det først en konseptuell modell og deretter en logisk modell. Disse er gjengitt i kapittel 5.2.1 og 5.2.2.

5.1.1 Kravspesifikasjon forenklet VA-modell

Fagmodellen for vann og avløp i Håndbok 138 har en høy detaljeringsgrad hvor blant annet kummer og rør er modellert i 3D med både indre og ytre avgrensning. Dette er vist i figur 2.1 for eksisterende fagmodeller. I denne oppgaven velges en lavere detaljeringsgrad hvor kun ytre avgrensning i bunn og topp kum i tillegg til kumløkk inngår. Stikningsdata for bunn innvendig kum og senter kumløkk samt senterlinje for kum inngår også.

For å modellere rør hadde det vært hensiktsmessig å bruke sylindere for å modellere runde rør, men dette utelukker rør i andre former (firkantede eller eggformet for eksempel). PostGIS støtter ikke sylindere, slik at senterlinjen til røret og egenskaper som indre og ytre diameter samt formen på røret lagres. Det er vist i artikkelen “Management and 3D visualization of pipeline networks using dbms and aec software” [76] at senterlinjen for rør og egenskaper om dimensjon er nok til å visualisere røret i sin fulle form og ikke bare senterlinjen. Kummer, ventiler og lignende kan vises med enkle 3D-symboler.

Egenskaper som ønskes dekket av fagmodellen er:

1. Senterlinje rør
2. Ytre avgrensning kum og kumløkk
3. Egenskaper som materiale, eier, volum, vedlikeholdsansvarlig, bruksområde, produsent, tverrsnittsform, tykkelse, indre og ytre diameter, type, prosesskode og temakode
4. Stikningsdata (linje) for rør (topp/bunn/senter etc)
5. Stikningsdata (punkt) for kum (overkant kumløkk, bunn utvendig kum)
6. Objektets kontraktpostnummer i kontrakten
7. Objektets prosesskode fra Håndbok 25 [77]

5.1.2 Kravspesifikasjon modell for innmåling

Innmålinger fra felt skal kunne brukes i geometrisk kontroll sammen med stikningsdata for å lage dokumentasjon og målebrev. For at innmålingene skal være enkelt å finne igjen må det kunne knyttes en del egenskaper til den. For innmåling av et rør er det ønskelig med flere egenskaper. Dette er egenskaper som kan legges til under innmåling på oppmålingsutstyret i dag og er til hjelp for å finne fram til riktig innmåling før geometrisk kontroll. Følgende egenskaper anses som nødvendig:

1. Generelle egenskaper
 - (a) Måletidspunkt
 - (b) Navn på objekt
 - (c) Objektkode/temakode
 - (d) Hvem har målt inn
 - (e) Geometri (punkt, linje etc)
 - (f) Godkjent i geometrisk kontroll
2. Rør (ledning)
 - (a) Høydereferanse (topp/bunn)
 - (b) Dimensjon
 - (c) Materiale (PVC, STJ etc)
 - (d) Farge rør
 - (e) Ledningstype (Vann, avløp etc)

5.1.3 Kravspesifikasjon for funksjonalitet

Data i fagmodellen og innmålingsdata skal kunne brukes til å gjennomføre en geometrisk kontroll. Kravene tar utgangspunkt i hvordan dette gjøres i Dronning Eufemias Gate og dokumentet “Krav til digitale leveranser i gjennomføringsfasen - vedlegg til konkurransegrunnlaget. E18 Bjørvika - Etappe 2 Dronning Eufemias Gate” [73] om geometrisk kontrollrapport og dokumentasjon. For planlegging er statistikk over mengder som antall, lengde og masser knyttet opp mot temakode og/eller prosesskode fra Håndbok 25 [77] interessant.

1. Funksjonalitet planlegging
 - (a) Antall (stk) ulike kummer.
 - (b) Lengde (meter) ulike dimensjoner og materiale
 - (c) Masseberegning
 - (d) Sortere på temakode og/eller prosesskode ved beregning av antall eller lengde
2. Funksjonalitet geometrisk kontroll og dokumentasjon:
 - (a) Beregne differanser mellom prosjektert geometri i fagmodell og innmålt geometri i 2D og 3D
 - (b) Linjer må splittes i sine enkeltpunkter og korteste avstand til objektet i fagmodellen beregnes fra enkeltpunktene
 - (c) Endre objektstatus i fagmodell til “som utført” hvis bygget innenfor toleranser

- (d) Lage avviksmelding på objektet hvis bygget utenfor toleranser (i denne prototypen satt til 10cm for alle objekter).
- (e) Skrive ut en rapport fra geometrisk kontroll med differanser og geometri som har et forhåndsdefinert navn:


```
åååå-mm-dd temakode tabellnavn i fagmodell GRAP xyz.csv
```
- (f) Legge inn byggetidspunkt på objektet i fagmodellen
- (g) Lage “som utført” dokumentasjon i .dwg-format.

5.2 Konseptuelle & logiske modeller

De konseptuelle modellene tar utgangspunkt i kravspesifikasjonene i kapittel 5.1.1 og 5.1.2 for henholdsvis fagmodellen og innmålingsdata. Fra den konseptuelle modellen i UML-diagram lages en logisk modell i programmet *pgModeler* som er et modelleringsprogram for PostgreSQL og PostGIS databaser. Ikke alle geometrityper (TIN, polyhedral surface etc) er støttet og SRID-kode for koordinatsystemer med mer enn fire siffer er ikke støttet. Dette kan manuelt endres i skriptet som *pgModeler* lager av modellen. Dette skriptet kan kjøres av databasen, slik at tabellene og forhold mellom tabeller opprettes i databasen.

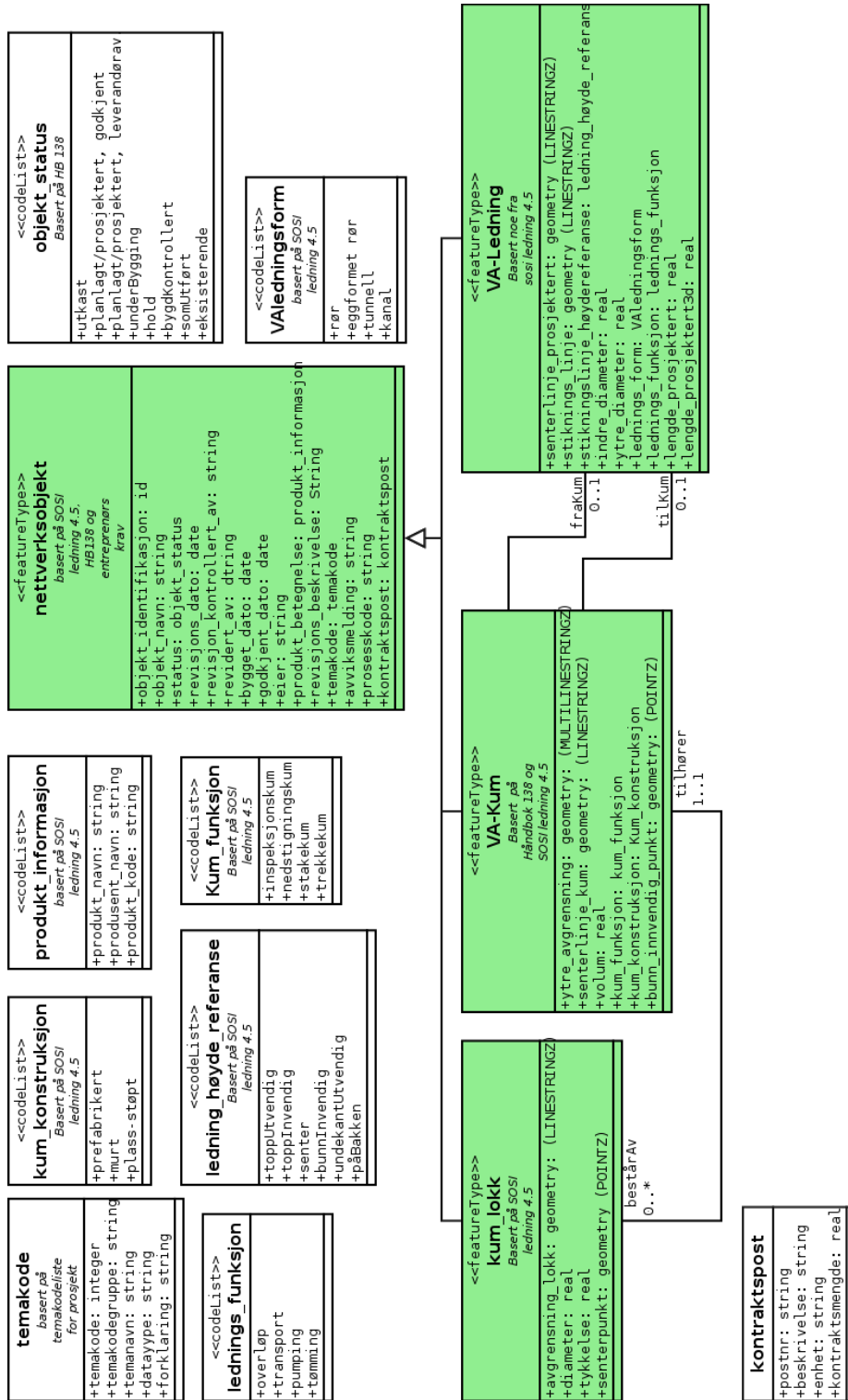
5.2.1 Forenklet fagmodell for VA

Den konseptuelle modellen er vist i figur 5.1 og beskriver en forenklet fagmodell for vann og avløp bygget opp av et nettverksobjekt. Til et nettverksobjekt kan det knyttes temakode, prosesskode fra Håndbok 25 [77] og kontraktspost sammen med egenskaper som revisjonsdato, byggedato, godkjent dato og avviksmelding. Disse egenskapene skal hjelpe de stikningsansvarlige under produksjon av stikningsdata, modellrevideringer og håndtering av dokumentasjon. Dette gir muligheter for å spore revideringer og status på det enkelte objekt i fagmodellen framfor på de enkelte lag i .dwg-filen som er dagens metode. Ved å knytte kontraktspostnummer og prosesskode åpnes det for lettere å kunne lage rapporter på mengene ved å filtrere data på denne egenskapen.

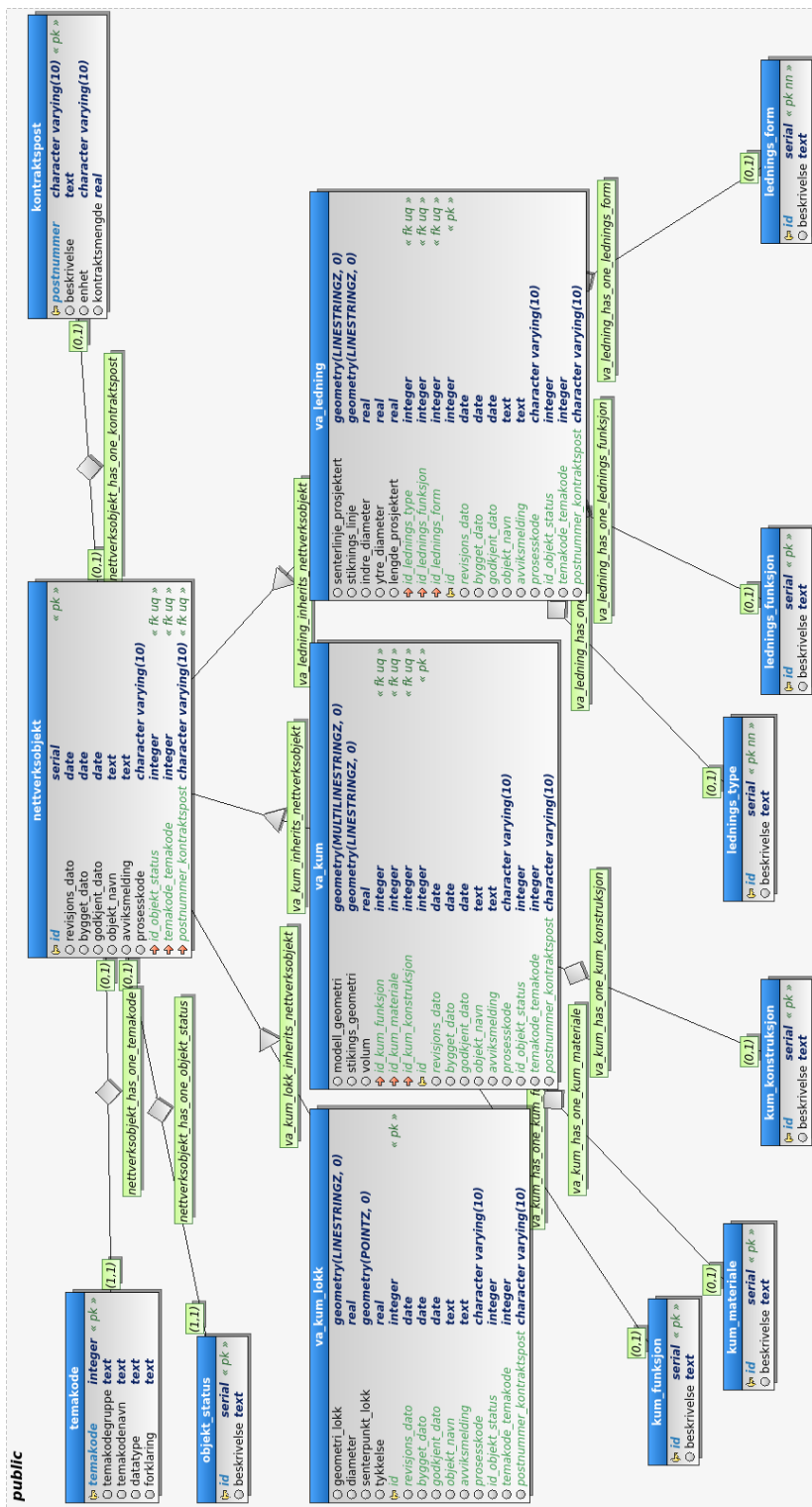
Et nettverksobjekt kan videre spesialiseres i en *va_kum*, *va_ledning* eller *va_kumlokk*. Det skilles ikke på om det aktuelle objektet tilhører et vann- eller avløpsnettverk i denne modellen slik som det gjøres i SOSI ledning 4.5. Temakoden på objektet angir hva slags type kum eller ledning og dermed nettverkstypen. Geometrien i fagmodellen består av to typer, modell- og stikningsgeometri. Dette tilsvarer praksisen i dag med egne lag med stikningsgeometri. *Kopling* fra SOSI ledning 4.5 er ikke tatt med i denne forenklede fagmodellen slik at en kum, hydrant, ventil eller lignende ikke er en spesialisering av *kopling*. I en mer omfattende modell for vann og avløp ville denne kunne vært med. Se kapittel 3.4.3 om en gjennomgang av SOSI ledning 4.5. Dette får betydning for topologiske forhold i modellen ved at en ledning kun kan gå mellom kummer.

Den logiske modellen er vist i figur 5.2. De enkelte tabellene er vist og egenskaper som er arvet fra *nettverksobjekt* er i grønn tekst. Topologiske forhold mellom

tabellene *va-kum* og *va-ledning* som i den konseptuelle modellen er utelatt i den logiske modellen. Det er valgt å ikke benytte seg av topologistøtten i PostGIS i denne implementeringen. En del egenskaper på *nettverksobjekt* er utelatt i den logiske modellen da det ikke var tenkt å bruke disse til å demonstrere geometrisk kontroll og lage dokumentasjon.



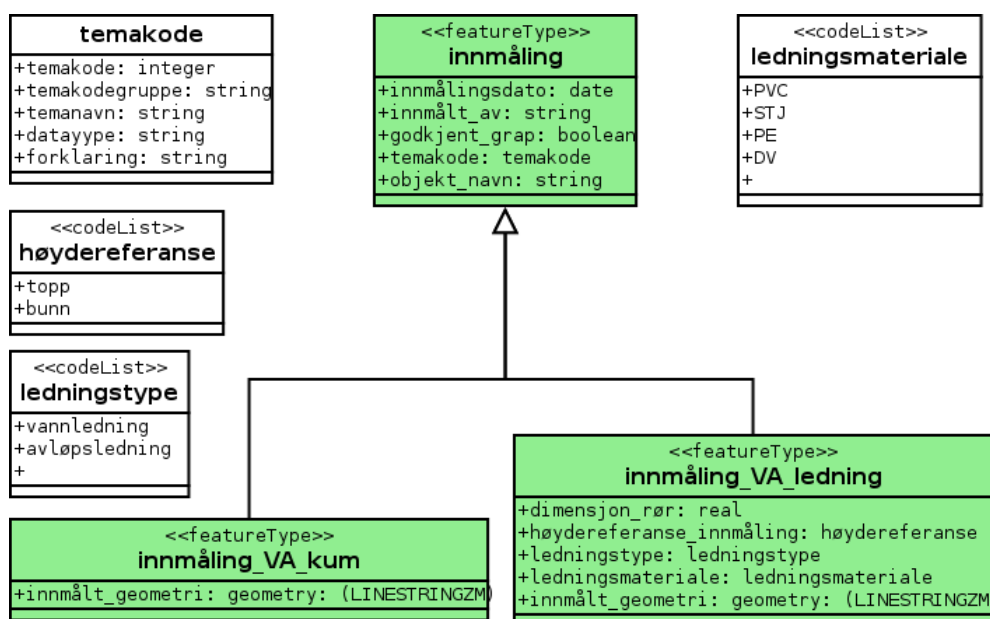
Figur 5.1: Konseptuell modell for forenklet fagmodell for vann og avløp.



Figur 5.2: Logisk modell for forenklet fagmodell for vann og avløp. Skjermdump fra pgModeler.

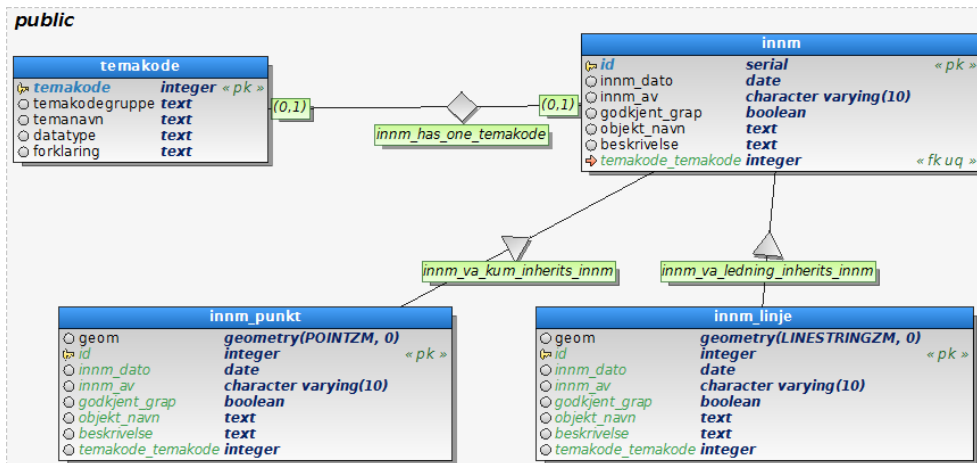
5.2.2 Modell for innmålingsdata

Modellen for innmålingsdata begrenses tilsvarende som for fagmodellen till å gjelde va-ledning, kum og kumløkk. Den konseptuelle modellen for innmåling er vist i figur 5.3 og er basert på kravspesifikasjonen i kapittel 5.1.2. For *innmåling-va-kum* er det senterpunkt til kumløkk og senterlinje for kummen som er interessant som dokumentasjon. Dette lages fra innmålte linjer som avgrenser kummen og kumløkket. Til *innmåling* knyttes egenskaper som måletidspunkt, hvem som har målt inn, temakode og om innmålingen er godkjent i grap. Dette har med filtreringsmuligheter på data som var ønsket av de stikningsansvarlige i kapittel 4.



Figur 5.3: Konseptuell modell for innmåling av vann og avløp.

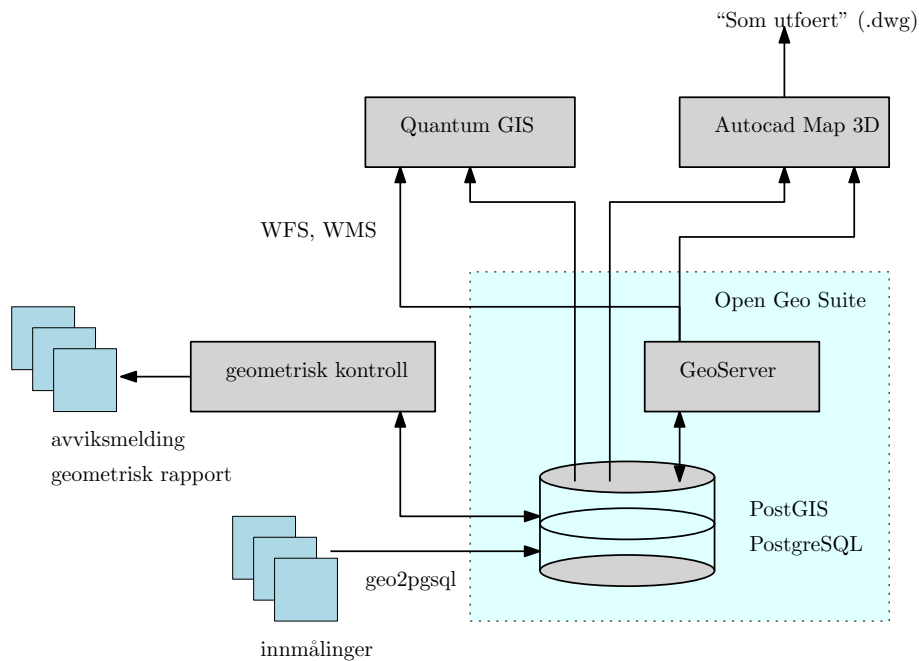
Det er valgt å lage en mer generell modell for innmålingsdata som tar hensyn til om geometrien er punkt eller linje. Egenskapene som er knyttet til *va-ledning* er ikke med i denne. Dette skyldes at tilgangen på innmålingsdata i .geo-formatet ikke har disse egenskapene og at den konseptuelle modellen ikke dekker innmålinger for punkter. Den logiske modellen for innmåling er vist i figur 5.4 og viser to klasser, *innm_linje* og *innm_punkt*, som begge er en spesialisering av *innm*. Egenskaper som er arvet fra tabellen *innm* er i grønt.



Figur 5.4: Generell logisk modell for innmåling. Skjermdump fra pgModeler.

5.3 Prototype på system

Prototypen omfatter flere programmer og komponenter for å kunne lagre modell- og innmålingsdata slik at det er mulig å gjennomføre geometrisk kontroll, visualisering og lage rapporter på mengder. Figur 5.5 viser de ulike delene i prototypen. Kjernen er databasehåndteringssystemet PostgreSQL med utvidelsen PostGIS. GeoServer kan brukes for å lage karttjenester som WMS og WFS, og som kan brukes i programmer som Quantum GIS eller AutoCAD Map 3D. WFS og WMS er nærmere beskrevet i kapittel 3.6. Disse programmene kan dessuten lese geometrien i PostGIS direkte.



Figur 5.5: Prototypens oppbygging av ulike komponenter.

Det er som en del av oppgaven utviklet et program i programmeringsspråket *Python* for å kunne lese innmålings- og modelldata inn i databasen (*csvgeo.py*) og et tilsvarende skript for å gjennomføre en geometrisk kontroll av innmålinger (*grap.py*). Dette er gjort fordi eksisterende programvare ikke kan lese *.geo*-formatet til databasen og for lettere å kunne gjennomføre en geometrisk kontroll. Programmene er gjengitt i tillegg B. For å lese inn modelldata fra *.dwg*-filer åpnes disse filene først i oppmålingsprogrammet *SBG Geo* hvor kun punkter og linjer som ikke er en del av et volumobjekt blir igjen. Det betyr at stikningslinjer for rør, punkter for bunn utvendig og topp lokk samt senterlinje for kummer står igjen. Dette blir i *SBG Geo* eksportert til *.geo*-formatet som er det samme formatet som for innmålinger. Dette kan så leses inn i databasen. Det leses inn ca 300 punkter og linjer til fagmodellen og ca 800 punkter og linjer fra innmålinger. Eksempel på hvordan innmålingsdata leses inn og utdrag fra SQL-dokument er gitt under:

```
python csvgeo.py -e 25832 -l public.innm_linje -p public.innm_punkt
-o ~/Desktop/ -m False -d 2012-08-14
-f ~/Desktop/innm_drensroer_25A_20120814_ER.geo
```

```
INSERT INTO public.innm_linje(id, innm_dato, objekt_navn,
temakode_temakode, geom)
VALUES (nextval('inm_id_seq'), '2012-08-14', '25ADREN2', 42203,
ST_GeomFromText('LINESTRING(598306.8224 6642440.3785 0.6031,
598307.6055 6642442.1829 0.6029)', 25832));
```

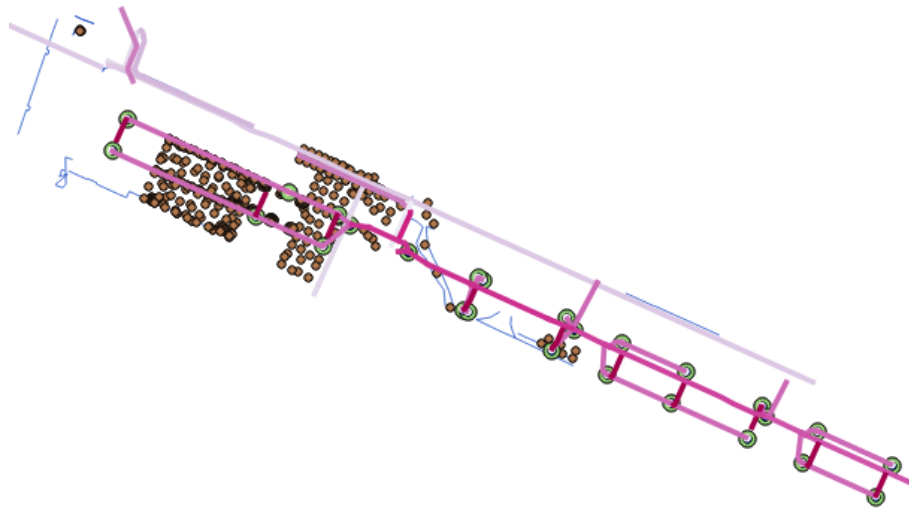
Det legges i liten grad inn egenskaper i tabellene som er beskrevet i de konseptuelle modellen med unntak av temakoder for vann og avløp og objektstatus. Alle nettverksobjektene gis i utgangspunkt statuskode 4 i denne oppgaven. Alle statuskodene med tilhørende beskrivelse er:

id	beskrivelse
1	som utført
2	under bygging
3	utkast
4	planlagt/prosjektert, godkjent
5	planlagt/prosjektert, leverandøravhengig
6	under bygging
7	hold
8	bygget kontrollert
9	eksisterende

5.4 Resultater

Med modell- og innmålingsdata lest inn i databasen skal geometrisk kontroll gjennomføres med programmet *grap.py*. Denne kontrollen gjennomføres på en vannledning og et kumlokk og skriptet lager en rapport med resultater fra kontrollen med egenskaper som innmålingsid, objektnavn, innmålt dato, geometri, differanser i 2D

og eventuelt i 3D, som skrives til en fil i csv-format. Geometrien blir konvertert til WKT. For innmålinger hvor gjennomsnittet av differansen er under en oppgitt toleranse, endres objektstatusen på objektet i fagmodellen til “som utført”, for objekter som er bygget utenfor toleransegrensen lages det en avviksmelding på. Endring av objektstatus og avviksmeldingen gjøres etter det som er avdekket i kapittel 4. Denne implementeringen tar kun hensyn til gjennomsnittet av alle målingene, ikke den enkelte innmålingen. En gjennomgang av hvordan denne kontrollen gjennomføres og resultater blir gitt. Det gis også eksempler på hvordan prototypen kan brukes i planlegging og lage “som utført” dokumentasjon i DWG. Et oversiktsbilde over geometri i fagmodell og innmålinger er vist i figur 5.6.



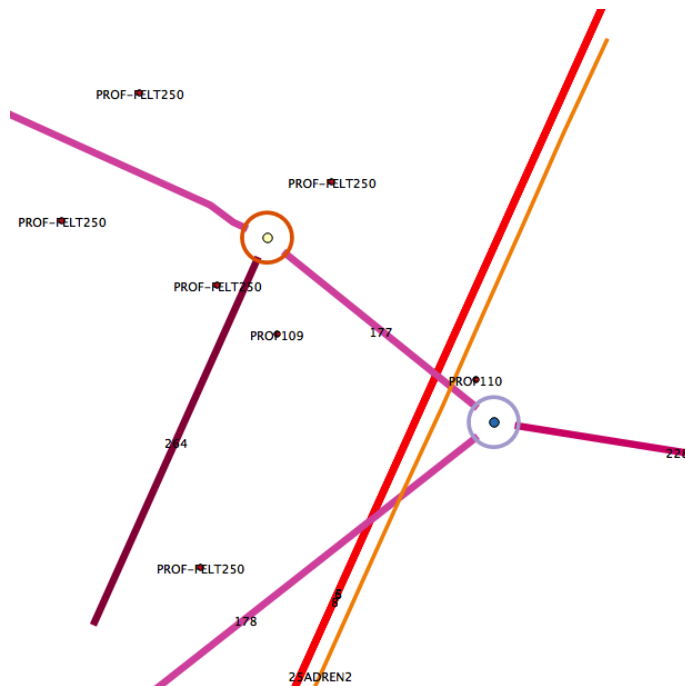
Figur 5.6: Oversiktsbilde over geometri fra fagmodell (rosa og grønn) og innmålingsdata (blå og brun) i prototype. Skjermdump fra QGIS.

5.4.1 Geometrisk kontroll av va-ledning

Figur 5.7 viser flere innmålinger bestående av små drenerør for planterøtter, profilering av bakken, modelldata for overvannsledninger farget i rosa, avgrensning til kummer og en vannledning vist i rødt. Denne vannledningen har temakode 42203 (VL 225) med *id* 5. En tenkt innmåling av deler av dette røret er vist i oransje farge. Spørringen som er vist under danner grunnlaget for å beregne avstand fra enkeltpunktene (`ST_DumpPoints`) i den innmålte linja til linja i fagmodellen.

Først filtreres innmålt geometri som er innenfor 5m fra vannledningen med *id* = 5 i fagmodellen: `ST_3DDWithin(i.geom, v.stiknings_linje, 5)`. Innmålingene må også ha samme temakode som objektet i fagmodellen (42203):

[...] and `i.temakode_temakode=42203`. Dette gjøres for å unngå at feil punkter, som for eksempel punkter fra profilering av grunnen vil bli tatt med under beregningen. Innmålinger kan også avgrensnes ved å angi en av de andre egenskapene i tabellen `innm_linje` som *innm-av* eller *innm-dato*. For de enkeltstående punktene fra innmålingen skal nærmeste punkt på linjen i fagmodellen finnes med `ST_3DClosestPoint`. Differansen på punktene beregnes i grunnriss og 3D.



Figur 5.7: Utsnitt fra fagmodell og innmålinger. Skjermdump fra QGIS.

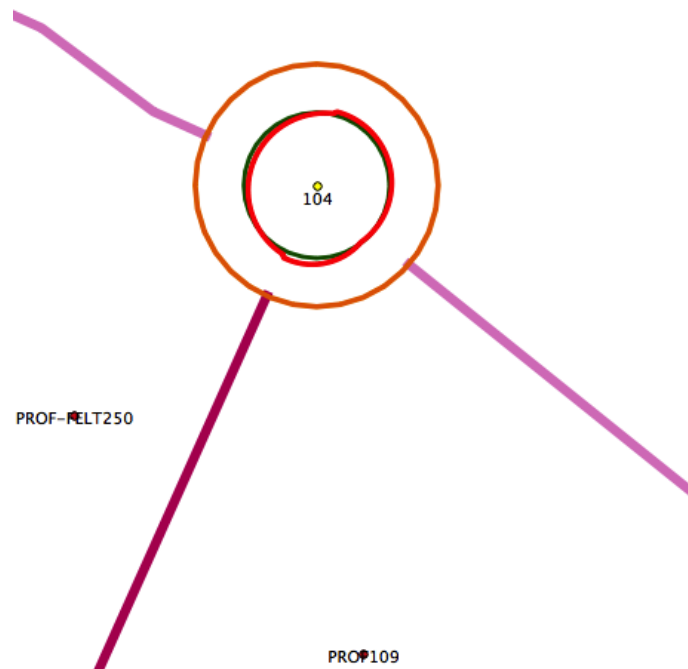
```
select ST_Distance(foo.rp, foo.sp) as diff2d,
ST_3DDistance(foo.rp, foo.sp) as diff3d from (
select ST_3DClosestPoint(v.stiknings_linje,
(ST_DumpPoints(i.geom)).geom) as rp,
(ST_DumpPoints(i.geom)).geom as sp
from innm_linje i, va_ledning v
where ST_3DDWithin(i.geom, v.stiknings_linje, 5)
and v.id=5 and i.temakode_temakode=42203
) as foo;
```

diff2d	diff3d
0.343	0.519
0.313	0.508
0.308	0.512

5.4.2 Geometrisk kontroll av kumlokk

Avgrensningen til en kum og et kumlokk er laget med å buffre rundt senterpunktet for kummen og lokket og plukke ut linja som avgrensner bufferet. Bufferoperasjonen fungerer kun i planet (2D), slik at kummen og kumlokket kun får en linje som viser ytre avgrensning uten høydeverdier.

For å finne senterpunktet benyttes funksjonen `ST_Centroid()` som også kun fungerer i 2D [59, kap. 8.8, 8.9]. Dette er en begrensning slik at geometrisk kontroll av kumlokket kun kan gjøres i 2D. Er senterpunktet på lokket også målt inn og ikke funnet med `ST_Centroid()` vil også denne kunne beregnes i 3D.



Figur 5.8: Utsnitt fra fagmodell for kummer og innmålinger. Skjermdump fra QGIS.

En tenkt innmåling av kumløkket til kum med *id 104* er vist i rødt i figur 5.8. Avgrensning til kumløkket og kummen er vist henholdsvis i grønn og oransje farge i samme figur. Senterpunkt til topp løkk fra fagmodellen er vist som et punkt. For kumløkket gjennomføres en geometrisk kontroll i 2D som beskrevet i kapittel 5.4.1 og en rapport med differanser for avgrensningen til kumløkket og senterpunktet lages. Et utdrag av denne rapporten er vist under med innmålingsid, objektnavn (tom i dette tilfellet), innmålingsdato, temakode, path (punktnummer i linja), differanse fra prosjektet (DIFF2D) og geometri i WKT-format. En utskrift av en avviksmelding med statistikk på avviket som skriptet lager er også vist. Avviksmeldingen er for den tenkte innmålingen av vannledningen i kapittel 5.4.1.

Utskrift fra rapport inkludert geometri i WKT-format:

```
ID;OBJEKT_NAVN;INNM_DATO;TEMAKODE;PATH;DIFF2D;GEOMETRY
798;;2013-04-04;42102;[1];0.017;POINT Z (598311.358 6642461.478 2.99)
798;;2013-04-04;42102;Null;0.015;POINT(598311.504 6642461.755)
```

Avviksmelding generert av program:

Avviksmelding

Objekt-id i fagmodell: 5

Objekt-id i innmaalinger: 251

Differanse i 2D (plan): 0.347

Differanse i 3D: 0.544

5.4.3 Planlegging

Et databasesystem som PostgreSQL har funksjoner for blant annet å beregne gjennomsnitt, summere, finne maks- og minimum, telle opp forekomster, gruppere og sortere data, matematiske funksjoner som kvadratrot, eksponenter og trigonometriske funksjoner [61]. Dette kan benyttes til å lage en rapport for å telle opp antall kummer eller finne lengden på rør som skal bestilles og legges på prosjektet. Spørringen for dette er vist under, og data fra fagmodellen kobles opp mot tabellen *temakode* for å få fram egenskapen *temakodenavn*. I spørringen for å beregne lengde på rør er stikningslinjen benyttet ettersom det ikke er lagt inn modelldata på senterlinje rør i fagmodellen.

Finn lengden på alle rør gruppert på temakode:

```
select temakode_temakode as temakode, temakodenavn,
sum(ST_Length(v.stiknings_linje))
from va_ledning v
join temakode t on v.temakode_temakode=t.temakode
group by v.temakode_temakode, t.temakode
order by t.temakode asc;
```

temakode	temakodenavn	sum
42202	Vannledning 180 PE	16.85
42203	Vannledning 225 PE	64.43

Finn antall kummer av en bestemt temakode:

```
select temakode_temakode as temakode, temakodenavn,
count(v.id) as antall
from va_kum_lokk v
join temakode t on v.temakode_temakode=t.temakode
group by v.temakode_temakode, t.temakode
order by t.temakode asc;
```

temakode	temakodenavn	antall
42102	Sandfangskum, topp lokk	47
42107	Minisandfang, topp lokk	3

5.4.4 “Som utført” dokumentasjon

AutoCAD Map 3D (versjon 2013) har mulighet for å koble seg til ulike databaser med geografisk informasjon, inkludert PostGIS. Det er mulig å filtrere data basert på verdier i tabellene, for eksempel kan bare data som har objektstatus “som bygget” vises eller data som har en bestemt temakode. Det er mulig å eksportere dette datasettet som en fil i dwg-format. Det opplyses om i dokumentasjonen til AutoCAD Map 3D at egenskaper til objektene, etiketter (eng: labels) og data fra andre tabeller som er koblet til datasettet ikke blir med. Lagnavnet blir navnet fra tabellen i originaldatasettet, men filnavnet kan bestemmes av brukeren.

Kapittel 6

Diskusjon & konklusjon

Resultater fra prototypen i et geografisk databasesystem for modell- og innmålingsdata prøver å løse utfordringene ved dagens bruk av modeller som er avdeket i kapittel 4. Diskusjonskapittelet starter med en kort oppsummering av dagens praksis ved Dronning Eufemias gate før konklusjon trekkes og muligheter videre arbeid presenteres.

6.1 Diskusjon

I studien av av Dronning Eufemias gate ble det avdekket at en sentral web-løsning (Buzzsaw) brukes for deling av modeller, referater, dokumentasjon og innmålinger mellom partene i prosjektet.

Modeller etter Håndbok 138 kan brukes som grunnlag for beregning av masser som brukes i kalkulasjonsprosessen i beregning av anbud. På stikningsmøter vises modellene på storskjerm for å visualisere hva som skal bygges og avdekke konflikter med geometri som kolliderer. I produksjon brukes modellene for å lage stikningsdata, enten fra egne lag eller ved å bryte ned geometri og bygge den opp igjen (med triangulering).

Fra innmålinger brukes stikningsdata og/eller fagmodellen til å beregne differanser mellom prosjektet og innmålt geometri. Det brukes i dag mye tid på prosjektet for å lete fram til riktige innmålinger som skal brukes i geometrisk kontroll fra ulike undermapper og filer med filnavn som skal beskrive innmålingens innhold.

Det brukes lister i regneark sammen med fargelegging av objektet på arbeidstegniner for å holde oversikt over hvilke objekter som det er levert innmålinger til, avviksmeldinger og godkjenningsdato. Dokumentasjonsarbeidet foregår for det meste manuelt.

6.1.1 Modelldata & formater

Formater som brukes for modelldata er IFC, DWG og LandXML. Sistnevnte format er spesifisert som åpent utvekslingsformat i Håndbok 138 for modelldata. DWG brukes som originalformat i Dronning Eufemias gate. Ut i fra definisjonen av BIM i kapittel 2.2 er det verdt å merke seg at hverken DWG eller LandXML kan betegnes

som BIM. Dette skyldes blant annet at det er statisk geometri (støtter ikke parametrisert oppførsel hvis omkringliggende objekter endres) og med lite egenskaper. I definisjonen av BIM legges også bygningskomponenter til en bygning til grunn. I kapittel 2.3 om Håndbok 138 utvides B-en i BIM til å gjelde alle typer konstruksjoner og anlegg (“BIM for infrastruktur”), men selv med en slik utvidet definisjon er ikke DWG og LandXML BIM. I implementeringen av denne fagmodellen er det ikke lagt opp til slik parametrisert oppførsel for objekter.

PostGIS kan hverken lese eller skrive til IFC-, DWG- eller LandXML-formatet direkte. Dette gjør at integrering med eksisterende formater er vanskelig. Formatene som PostGIS kan skrive til er basert på åpne standarder, slik at det er lett å lage programmer for å lese og skrive til disse formatene. GML er et alternativ for modelldata som PostGIS kan lese og skrive. CityGML som er basert på GML er det ikke innebygget støtte for. Programmet *FME* fra *Safe Software* kan lese og skrive til de fleste av disse formatene og kan integreres i en databaseløsning eller som et selvstendig program. Denne muligheten er ikke benyttet eller undersøkt i denne oppgaven.

6.1.2 Prototype på fagmodell vann og avløp

Den begrensede fagmodellen for vann og avløp i prototypen er modellert med enkel geometri som punkt og linje avledet av eksisterende modelldata i DWG. Nesten alle data er lagret med 3D-koordinater (X, Y, Z). Muligheten for å bruke volumobjekt (polyhedral surface) til for eksempel å representere en kum er ikke benyttet, men en linje som representerer ytre avgrensning til kummen i planet (X, Y) er brukt. Denne avgrensningen ble laget ved å ta linja som avgrenser bufferet rundt senterpunktet for kum (bunn invendig). Denne operasjonen for å lage ytre avgrensning ble gjort for å spare tid, men hadde disse linjene vært definert manuelt med høydeverdier, kunne differanser også i 3D vært gjennomført på geometrien for ytre avgrensning. Se kapittel 6.1.5 om geometrisk kontroll. Tilsvarende løsning ble også benyttet for kumløkket.

PostGIS har også støtte for topologi og dette er særlig viktig i IFC til BIM, men også for GIS-data. Dette åpner for at PostGIS kan håndtere modelldata som er bygget opp med topologi, men denne muligheten er ikke benyttet eller utforsket i denne oppgaven.

Det er lagt opp til at det enkelte nettverksobjekt i fagmodellen skal ha en egen objekstatus (hold, “som utført ” etc), godkjennings-, revisjons-, og byggedato. Dette vil sikre at endringer spores på objektnivå framfor praksisen i dag ved å gruppere store deler av geometrien i fagmodellen til et lag. Eventuelle avviksmeldinger eller referanser til en avviksmelding kan også knyttes til objektet. Sporing til når det enkelte objektet ble bygget og godkjent som bygget av byggherren er med å hjelpe både entreprenør og byggherre i dokumentasjonsarbeidet ved at det blir mindre papir og manuelt arbeid. Å knytte denne informasjonen til objektet er å se på som en berikelse av modellen med informasjon. Prosessen med å modellere opp en separat “som utført” modell blir redusert til spørring mot databasen som returnerer alle objekter med objektstatusen “som utført”. Det jobbes på en modell framfor flere (prosjektert og “som utført”).

Standarder for overføring av data som WFS og W3DS, kan brukes for å ha modelldata tilgjengelig i felt på pc-er eller mobile enheter som telefoner og nettbrett. Dette kan være til hjelp både for de som bygger og stikkeren som skal sette dette ut for bygging ved at modeller kan være tilgjengelig i felt. Det bør da være mulig å kunne plukke linjer og punkter direkte fra modellene (for stikkeren) og sette de ut på anlegget om vi ser et stykke framover i tid.

6.1.3 Prototype på modell for innmålingsdata

Innmålingsdata lagres i separate filer i mappestrukturer av den enkelte stikker på prosjektet som avdekket i kapittel 4.3 og beskrevet innledningsvis i kapittel 6.1. Ved å strukturere innmålingsdata og kode disse med objektnavn, temakode, hvem som har innmålt, innmålingsdato og eventuelt andre egenskaper gjør det jobben til databasehåndteringssystemet enklere med å finne fram til de riktige innmålingene som skal brukes i geometrisk kontroll.

PostGIS kan ikke lese .geo-filer som er formatet for innmålingsdata, men et egenutviklet program leser og konverterer innmålingsdata slik at det kan skrives til databasen. Programmet er lagt opp til å kjøre fra et terminalvindu og parametre som EPSG-kode, filnavn til innmålingsfil og navn på tabell for linjer og punkt i databasen må angis. Innmålingsdata konverteres til et SQL-dokument som må leses inn av databasen. Det er ikke den mest brukervennlige måten at alle parametrene må angis og at det lages en ny fil som må åpnes i et annet program for å kunne skrives til databasen.

WFS er en standard for overføring av vektor- og egenskapdata med mulighet for transaksjonsstøtte for å skrive eller endre data i databasen. Denne standarden kan benyttes sammen med oppmålingsutstyret når det har en Internett-oppkobling til å sette inn data i databasen direkte fra felt.

6.1.4 Planlegging

PostgreSQL har støtte for funksjoner som å telle opp og summere som vist i kapittel 5.4.3 om planlegging. Det er i dag ikke mulig å beregne volum i PostGIS, som begrenser muligheten noe når det kommer til å planlegge mengden masser som skal graves ut og/eller fylles på. Resultater fra spørringer som PostGIS lager kan inngå som inn-data i andre programmer (for eksempel regneark) for riktig formatering, presentering og videre beregning.

Ser man videre på planleggingsfasen utover entreprenørens behov for ved beregning av anbud er visualiseringsmuligheter av data en mulighet. 9-snittsmodellen kan brukes til å finne geometri som overlapper og dette kan brukes som en form for kollisjonskontroll. Det er ikke full støtte for dette i 3D og dette er heller ikke testet ut i denne oppgaven.

6.1.5 Geometrisk kontroll & dokumentasjon

Det er utviklet et program som gjennomfører geometrisk rapport på en vannledning og et kumløkk. For vannledninger er det vist at avstand mellom prosjekttert

og innmålt kan beregnes i 3D og 2D fra den innmålte linjen's enkeltpunkter. Denne vannledningen var utenfor toleransen slik at en avviksmelding ble laget. For kumlokket som kun var modellert i 2D kunne en tilsvarende avstandsberegning kun foretas i 2D. Rapporten som er laget sier ikke noe om punktet er innenfor eller utenfor toleransen og dermed ikke om det enkelte punkt er godkjent eller avvist. Oppdatering av status til "som utført" bruker gjennomsnittet av avviket for å sjekke om den skal oppdatere status. Dette er begrensning i den implementeringen som gjøres i denne oppgaven, men det er vist at modell- og innmålingsdata sammen kan brukes til å gjennomføre geometrisk kontroll med funksjonalitet som finnes i PostGIS.

Det er mulig å tenke seg at kun innmålingsdata lagres i en geografisk database, det vil si ikke noe fra en fagmodell, sammen med definerte områder i prosjektet. Disse områdene kan for eksempel være plate 24 eller 25, som brukes i Dronning Eufemias gate, og kan lagres som et polygon. Denne avgrensningen kan brukes til å finne innmålinger innenfor dette området med PostGIS. Sammen med andre filtreringsmuligheter er det fortsatt gode muligheter til å finne fram til innmålinger som skal brukes i geometrisk kontroll framfor å lete igjennom mapper og filer.

Det er avdekket at prosessen med å lage "som utført" dokumentasjon er manuell og krever mye sjekk i lister (regneark) over hva det er levert innmålinger på, hva som er godkjent og/eller levert avviksmeldinger på, før en "dom utført" modell kan lages. I denne prototypen ligger alt dette som egenskaper til det enkelte objekt i databasen og det gjør at brukerne på et prosjekt jobber mot en felles modell framfor enkeltfiler og lister.

AutoCAD Map 3D kan brukes for å eksportere data fra databasen til DWG, men ingen egenskaper utover tabellnavnet i databasen blir med. Dette er en begrensning i .dwg-formatet. DXF er en mulighet som et alternativt format, men AutoCAD Map 3D kan ikke eksportere til DXF, slik at det ikke blir med noen egenskaper hvis man velger å eksportere til DXF etter å ha lagret først i DWG. AutoCAD Map 3D kan blant annet eksportere til ESRI shapefile og GML som er vanlige formater i GIS.

6.2 Konklusjon

Det er vist at en prototype i et geografisk databasesystem (PostgreSQL + PostGIS) kan håndtere modelldata for en begrenset fagmodell for vann og avløp. Fagmodellen er bygget opp av enkel geometri som punkt og linje. Den er ikke bygget opp med topologi og det må sies å være en begrensning i denne prototypen. PostGIS har støtte for flere typer geometri, herunder TIN og volumobjekter (polyhedral surface), slik at det er muligheter for å bygge opp en fagmodell med mer avansert geometri.

Prototypen håndterer også innmålingsdata fra felt. Sammen med data fra fagmodellen kan også det egenutviklede programmet benytte seg av funksjoner i PostGIS til å finne fram til innmålinger av ett objekt i fagmodellen og gjennomføre geometrisk kontroll. Det er vist at resultater fra denne kontrollen kan brukes til "som utført" dokumentasjon ved at status på objekter i fagmodellen endres til "som utført" eller en avviksmelding skrives til fil på det aktuelle objektet.

Prototypen i databasen fungerer som et sentralt lagringssted for modell- og innmålingsdata hvor objektstatus og dato for godkjenning, byggedato, revisjonsdata og lignende er knyttet til det enkelte objekt. Det gjør at en spørring mot databa-

sen med filtrering på en eller flere av de nevnte egenskapene kan brukes til å få ut de nødvendige dataene, enten det er for bruk til stikning eller dokumentasjon. En slik sentral lagringsløsning åpner også for at en del av det manuelle arbeidet med dokumentasjon og lister kan minskes eller elimineres ved å kunne knytte denne informasjonen til objektene i fagmodellen. Ved å bruke standarder for overføring av data til og fra databasen kan det åpnes for nye muligheter i felt som ikke brukes i dag.

Alle komponenter i denne prototypen på systemet med unntak av AutoCAD og AutoCAD Map 3D som lager .dwg-filer er åpen kildekode og/eller gratis. Oppmålingsprogrammet SBG Geo er heller ikke gratis eller åpen kildekode.

DWG og LandXML brukes for modelldata i dag, men det er ulik geometristøtte (volumobjekter) som gjør at de ikke er helt kompatible. IFC har ikke i dag en modell som beskriver objekter som brukes i infrastrukturprosjekter, men dette kan forandre seg. GML eller et applikasjonsskjema tilsvarende CityGML laget med tanke på infrastrukturprosjekter er et alternativ for modelldata.

Selv med en prototype på en så begrenset fagmodell og de begrensningene som er avdekket underveis i PostGIS er det vist at et geografisk databasesystem kan anvendes til å håndtere modell- og innmålingsdata i et infrastrukturprosjekt. Håndtering av innmålingsdata er et særlig område som en slik database kan komme til anvendelse. En løsning med bruk av en geografisk database for modell- og innmålingsdata er veien å gå og støtter oppunder BIM i infrastrukturprosjekter. En slik løsning vil kunne være med å forenkle hverdagen og fungere som et godt verktøy for brukere i planleggings-, produksjons- og dokumentasjonsfasen.

6.3 Videre arbeid

Med et så godt potensiale for å håndtere modell- og innmålingsdata kan det være interessant å jobbe videre med å implementere andre fagmodeller som vei og konstruksjoner på en tilsvarende måte. De begrensningene som er foretatt i implementeringen i denne prototypen kan være interessant å undersøke videre. Det er geometrityper som ikke ble benyttet i denne prototypen, slik som *polyhedral surface* og *TIN* som kan være interessante å bruke til å modellere volumobjekter og terrengoverflater. For volumobjekter kunne kollisjonskontroll vært interessant å gjennomføre og se videre på. Det er interessant å se på fagmodeller bygget opp med topologi for å inkludere dette. Funksjonalitet i programmer som leser innmålingsdata og gjennomfører geometrisk kontroll kan utvides og bygges inn i programmer med grafisk brukergrensesnitt. For programmet som leser innmålingsdata kan utvidet funksjonalitet være å skrive data direkte til databasen uten å gå via .sql-dokumentet. Det er også mulig å bygge videre på dette programmet med å bruke informasjon om det enkelte punkt framfor gjennomsnittet av differansen for å avgjøre om “som utført” status skal gis. Det er også mulig å se på mulighetene som ligger i *FME* for å skrive og lese modelldata i .ifc-, .dwg-, eller landxml-formatene.

Bibliografi

- [1] Vegdirektoratet. Håndbok 138 modellgrunnlag – krav til grunnlagsdata og modeller. Statens vegvesen håndbok, Vegdirektoratet, Oktober 2012.
- [2] Andreas Haugbotn. Bim and automation in infrastructure sector. <http://tuta.oulu.fi/LCI/%20Haugbotn.pdf>, 2009.
- [3] Statsbygg. Statsbygg går for bim. <http://www.statsbygg.no/Aktuelt/Nyheter/Statsbygg-gar-for-BIM>, Mai 2007.
- [4] Inger Hokstad. Ba-nettverket - hva er det? *Posisjon*, 20(3):12–14, 2012.
- [5] Miljøverndepartementet/Regjeringen. Tverrsektorielt samhandlingsprosjekt for bygg, samferdsel, annen samfunns viktig infrastruktur og geodata. invitasjon til forprosjekt. http://www.regjeringen.no/pages/37857173/Vedlegg_1_Invitasjon_til_forprosjekt.pdf, April 2012.
- [6] Eivind Hurlen and Terje Glad. Virtuell virkelighet i ba-næringen. *Kart og Plan*, 71(4):241–244, 2011.
- [7] Vianova Systems AS. Bim for infrastructure. <http://www.vianovasystems.com/BIM>, April 2013.
- [8] BA-Nettverket. Om ba-nettverket. <http://banettverket.no/default.aspx>, April 2013.
- [9] Store norske leksikon. Modell. <http://snl.no/modell>, Januar 2013.
- [10] Store norske leksikon. Format - it. <http://snl.no/format/IT>, Januar 2013.
- [11] Store norske leksikon. Geografisk informasjonssystem. http://snl.no/geografisk_informasjonssystem, Januar 2013.
- [12] Michael F. Worboys. *GIS A Computing Perspective*. Taylor and Francis, 1995.
- [13] Chuck Eastman, Paul Teicholz, Rafael Sacks, and Liston Kathleen. *BIM Handbook*. Wiley, second edition, 2011.
- [14] buildingSmart Norge. Om buildingsmart. <http://www.buildingsmart.no/buildingsmart>, Januar 2013.

- [15] Thomas Liebich. Ifc for infrastructure - separation between bim and gis requirements. technical scope definition. proposed first steps. http://iug.buildingsmart.org/resources/product-room-workshop-16-october-2012/20121017_IFC_for_Infrastructure_TL.pdf/at_download/file, Oktober 2012.
- [16] Norge Digitalt. Norge digitalt hjemmeside. <http://www.norgedigitalt.no>, April 2013.
- [17] Philippe Rigaux, Michel Scholl, and Agnès Voisard. *Spatial Databases with application to GIS*. Morgan Kaufman, 2002.
- [18] Wikipedia. Srid. <http://en.wikipedia.org/wiki/SRID>, April 2013.
- [19] Wikipedia. Epsg. http://en.wikipedia.org/wiki/EPSG#European_Petroleum_Survey_Group, April 2013.
- [20] Günter Seeber. *Satellite Geodesy*. Walter de Gruyter, 2nd edition, 2003.
- [21] Thomas H. Meyer. *Introduction to Geometrical and Physical Geodesy. Foundations of Geomatics*. ESRI Press, 2010.
- [22] Wikipedia. Kilometrering. <http://no.wikipedia.org/wiki/Kilometrering>, April 2013.
- [23] Bernard Hofmann-Wellenhof, Herbert Lichtenegger, and Elmar Wasle. *GNSS - Global Navigation Satellite Systems. GPS, GLONASS, Galileo and more*. SpringerWienNewYork, 2008.
- [24] Kartverket. Referanserammer i norge. <http://www.kartverket.no/Kunnskap/Kart-og-kartlegging/Referanseramme/Referanserammer-for-Norge/>, April 2013.
- [25] Jan Terje Bjørke. Datamodeller, algoritmer og datastrukturer for geografisk informasjonsbehandling, 2005.
- [26] BuildingSMART. Industry foundation classes release 4 [ifc4]. <http://www.buildingsmart-tech.org/ifc/IFC4/final/html/index.htm>, 2013.
- [27] Open Geospatial Consortium Inc. Opengis®geography markup language (gml) encoding standard, version 3.2.1. Opengis® standard, OGC, ISO/TC211, 2007.
- [28] Vincent Picavet. State of the art of foss4g for topology and network analysis [foss4g 2010 - barcelona]. <http://2010.foss4g.org/presentations/3555.pdf>, 2010.
- [29] Open Geospatial Consortium Inc. Opengis® implementation standard for geographic information - simple feature access - part 1: Common architecture [version 1.2.1]. Opengis standard, OGC, May 2011.

- [30] S. Zlatanova, A.A. Rahman, and W. Shi. Topology for 3d spatial objects. In *International Symposium and Exhibition on Geoinformation*, pages 22–24, 2002.
- [31] A.A. Rahman. Developing three-dimensional (3d) topological model for 3d gis. *Geoinformation Science and Engineering*, 2005.
- [32] F Döner and C Bıyık. Modelling and mapping third dimension in a spatial database. *International Journal of Digital Earth*, 4(6):505–520, 2011.
- [33] Open Geospatial Consortium Inc. Open geospatial consortium homepage. <http://www.opengeospatial.org/>, 2013.
- [34] T.H. Kolbe. Representing and exchanging 3d city models with citygml. *3D geo-information sciences*, pages 15–31, 2009.
- [35] Open Geospatial Consortium Inc. Ogc city geography markup language (citygml) encoding standard [version 2.0.0]. Opengis® encoding standard, OGC, April 2012.
- [36] Kartverket. Sosi arbeidsgruppe 7b - ledning - sosi ledning 4.5. <http://kartverket.no/Standarder/SOSI/SOSI-arbeidsgruppe-7b/>, Mars 2013.
- [37] SOSI-Ag7b and SOSI-sekretariatet. Sosi - del 2, fagområde: ledningsnett v. 4.5. Sosi standard, Kartverket, Januar 2013. Utkast, redigeringsdato 2013-01-30.
- [38] buildingSmart. buildingsmart specifications. <http://www.buildingsmart-tech.org/specifications>, Mars 2013.
- [39] Wikipedia. Constructive solid geometry. http://en.wikipedia.org/wiki/Constructive_solid_geometry, Januar 2013.
- [40] International Alliance for Interoperability. Ifc2x edition 3 technical corrigendum 1. <http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/index.htm>, 2007.
- [41] IFD Developer Library, Lars Bjørkhaug, and Håvard Bell. Ifd in a nutshell. http://dev.ifd-library.org/index.php/Ifd:IFD_in_a_Nutshell, April 2013.
- [42] Wikipedia. Dwg. <http://en.wikipedia.org/wiki/.dwg>, April 2013.
- [43] underland.no. Landxml. http://underland.no/?page_id=612, Mars 2013.
- [44] LandXMLproject.org. Landxml project. <http://www.landxmlproject.org/>, April 2013.
- [45] Autodesk. Dxf reference. <http://www.autodesk.com/techpubs/autocad/acad14/dxf/>, April 2013.
- [46] GeoServer. Geoserver homepage. <http://geoserver.org/display/GEOS/Welcome>, April 2013.

- [47] W3DS. Web 3d service web. <http://w3ds.org/doku.php>, April 2013.
- [48] Open Geospatial Consortium Inc. Ogc® wcs 2.0 interface standard - core. Opengis® interface standard, OGC, Oktober 2010.
- [49] Open Geospatial Consortium Inc. Opengis® web map server implementation specification [version 1.3.0. Opengis standard, OGC, Mars 2006.
- [50] Open Geospatial Consortium Inc. Opengis web feature service 2.0 interface standard. Opengis® implementation specification, OGC, November 2011.
- [51] C. P. Lo and Albert K. W. Yeung. *Concepts and techniques of geographic information systems*. Pearson Prentice Hall, second edition, 2007.
- [52] Alan Dennis, Barbara Halye Wixom, and Tegarden David. *Systems Analysis and Design - An object-oriented approach with UML*. John Wiley & Sons, Inc, 2002.
- [53] Wikipedia. Uml. http://en.wikipedia.org/wiki/Unified_Modeling_Language, Mars 2013.
- [54] Wikipedia. Uml class diagram. http://en.wikipedia.org/wiki/Class_diagram, Mars 2013.
- [55] Wikipedia. Dbms. <http://en.wikipedia.org/wiki/DBMS>, Februar 2013.
- [56] Wikipedia. Ordbms. <http://en.wikipedia.org/wiki/ORDBMS>, Februar 2013.
- [57] S. Zlatanova, S. Pu, and WF Bronsvort. Freeform curves and surfaces in dbms-a step forward in spatial data integration. In *Proceedings of the ISPRS Commission IV Symposium on 'Geospatial Databases for Sustainable Development*, pages 27–30, 2006.
- [58] S. Pu and S. Zlatanova. Integration of gis and cad at dbms level. In *Proceedings of UDMS*, volume 6, pages 15–17, 2006.
- [59] Postgis Refractions Research. Postgis 2.0 manual. <http://postgis.net/docs/manual-2.0/>, Januar 2013.
- [60] Oracle. Advanced spatial data management for enterprise applications. Oracle white paper, Oracle, August 2010.
- [61] The PostgreSQL Global Development Group. About postgresql. <http://www.postgresql.org/>, Januar 2013.
- [62] PostGIS. Postgis 2.0. <http://postgis.refrations.net/>, Januar 2013.
- [63] Oslandia. 3d spatial operations and exact geometries for postgis. <http://www.oslandia.com/tech/?p=1235>, Mars 2013.
- [64] S Malaikrisanachalee and H Vathananukij. Integration of java-based bim with spatial database. *International Journal of Civil Engineering*, 9(1), 2010.

- [65] A. Stadler, C. Nagel, G. König, and T.H. Kolbe. Making interoperability persistent: A 3d geo database based on citygml. *3D Geo-Information Sciences*, pages 175–192, 2009.
- [66] Jürgen Döllner, Thomas H Kolbe, Falko Liecke, Takis Sgouros, and Karin Teichmann. The virtual 3d city model of berlin-managing, integrating, and communicating complex urban information. In *Proceedings of the 25th Urban Data Management Symposium UDMS*, pages 15–17, 2006.
- [67] Institute for Geodesy and Technische Universität Berlin Geoinformation Science. 3d city db. <http://www.3dcitydb.org/>, Mars 2013.
- [68] Skanska. Skanska bygger ny praktgate i oslo. <http://www.skanska.no/Pages/DisplayNews.aspx?id=3631&epslanguage=no&nid=bQTLfW0u>, August 2011.
- [69] Skanska. Dronning eufemias gate. <http://www.skanska.no/no/Prosjekter/Prosjekt/?pid=31&plang=nb-no>, 2011.
- [70] Byggeindustrien. Hæhre og skanska sikrer anbud med bim. <http://www.bygg.no/2013/04/104798.0>, April 2013.
- [71] Torbjørn Tveiten. Rv. 150 ring 3 ulven-sinsen - bruk av bim for infrastruktur. *Kart og Plan*, 72(3):206–208, 2012.
- [72] Torbjørn Tveiten. Bjørvika - bruk av 3d i planarbeidene. *Kart og Plan*, 71(4):235–240, 2011.
- [73] Statens Vegvesen. Krav til digitale leveranser i gjennomføringsfasen - vedlegg til konkurransegrunnlaget. e18 bjørvika - etappe 2 dronning eufemias gate. Buzzsaw prosjektstyringsverktøy Bjørvikaprosjektet, November 2011.
- [74] Vegdirektoratet. Håndbok 018 vegbygging (normaler). Statens vegvesen håndbok, Vegdirektoratet, Januar 2011.
- [75] Statens Vegvesen. Nvdb datakatalog. <http://kunder.runit.no/vegvesen/datakatalog/>, April 2013.
- [76] Y. Du, S. Zlatanova, and X. Liu. Management and 3d visualisation of pipeline networks using dbms and aec software. In *Proceedings of the ISPRS commission IV symposium on geospatial databases for sustainable development*, pages 27–30, 2006.
- [77] Vegdirektoratet. Håndbok 025 prosesskode 1. Statens vegvesen håndbok, Vegdirektoratet, Desember 2012.

Tillegg A

Intervju med entreprenør

Intervjuet er foretatt med stikningsansvarlige Bezim Zekaj og Lars Petter Lingjærde fra entreprenøren Skanska på anleggsriggeren i Bjørvika 24. januar 2013. Notatene fra intervjuet er renskrevet.

Buzzsaw og dataflyt mellom entreprenør og byggherre

- Hvordan benyttes Buzzsaw i prosjektet deres til å motta data og levere data mellom entreprenør og byggherre?
- Opplever du at Buzzsaw er en god nok løsning eller er det mangler ved funksjonalitet til produktet?
- Benyttes kun 3D-modeller? Tegninger som supplement/støtte eller som en del av datagrunnlag for entreprenør
- Innlevering av “as-built” dokumentasjon og innmålinger

Buzzsaw anvendes for all kommunikasjon mellom entreprenør og byggherre. Buzzsaw er et webhotell for møtedokumenter, modeller og arbeidstegninger. Innmålinger, geometrisk kontroll og målebrev leveres her. Reviderte modeller og tegninger lastes opp og det er sporingshistorikk på de enkelte filene (hvem, når, versjon). Når ny utgave av en fil lastes opp overskrives den forrige utgaven, men en kan gå tilbake i historien på tidligere utgaver av filen. Det er mulig å varsle andre brukere på prosjektet med e-post når ny modell/tegning lastes opp direkte i Buzzsaw.

Buzzsaw oppleves som en ryddig og oversiktlig løsning. Den virker tilfredsstillende for kommunikasjon mellom entreprenør og byggherre. De stikningsansvarlige opplever ingen spesielle mangler ved funksjonaliteten i dag.

3D-modeller benyttes til stikning og 2D-tegninger (papir) må genereres manuelt (av byggherre/konsulent) når en 3D-modell endres/revideres. 2D-tegningen brukes til arbeidstegninger. Ikke alt er i 3D-modellen av detaljer på objektene (eks. Belysning der boltegruppe er med, men ikke jordledning, som er på arbeidstegningen). All nødvendig geometri for å sette ut ting på byggeplassen hentes fra modellen som egne lag med geometri for stikkeren eller lages av de stikningsansvarlige.

Oppdatering av tegninger skjer ved å skrive ut tegningene og bytte ut i en perm hvor alle arbeidstegningene er satt inn samlet. Dato påføres med stempel. Revideringer på modeller kommer som et eget lag i modellen og geometrien er fargekodet (rødt). Dette gjelder også for nye data i en eksisterende modell som kommer inn på et senere tidspunkt.

Produksjon av stikningsdata til VA (kum og rør) i dag

- Beskriv og/eller illustrer arbeidsflyten kort for hvordan stikningsdata til rør og kummer lages i dag fra dwg-modell til stikningsdata.

Nødvendig geometri for kum er OK (overkant), UK (underkant), og for rør er en 3D linje OK/UK rør som kommer på eget lag. Denne .dwg-fila blir åpnet i geo direkte som en egen fil og det lages .lin- og .prf-filer av linja (henholdsvis horisontalgeometri og vertikalgeometri) i oppmålingsprogrammet SBG Geo. SBG Geo bryter ned geometrien. Eksport til Leica DBX, som er oppmålingsutstyrets format.

Kompabilitet mellom formater

- Hvordan opplever du formatet LandXML? Er det dekkende for dine behov eller er det mangler i dag?
- Har du behov for 3D volumobjekter i LandXML?
- Konvertering mellom formater bevarer nødvendig informasjon eller forsvinner denne underveis?
- Oppleverer du at oppmålingsutstyr/maskinstyringsutstyr ikke har støtte for nok geometri og at dette er en ulempe for stikkeren/graveren ute på byggeplassene når du produserer stikningsdata?

Det er lite kjennskap til LandXML-formatet, men det brukes i maskinstyring i dag (gravemaskin). 3D volumobjekter savnes dermed ikke i dag. Bruker originalformatet til modell til alt i dag. VIPS for vei er greit da det gir flere muligheter i programvaren (Novapoint) i dag.

For innmålingsdata fra felt følger eventuell koding av objektet i felt med. Navn på punkter og linjer som eksporteres fra modell (dwg) til stikningsdata i SBG Geo får generert tilfeldig navn, ikke noe med lagnavn (temakode etc). Lagnavnet består som navn på .geo-fila, og i egenskapsfeltet til punktet eller linja i .geo-fila.

Oppmålingsutstyr har begrenset geometristøtte: punkt, rette linjer og kan lage krumme linjer med angitt radius hvis start og stoppunkt angis. Denne muligheten er lite brukt i dag da radius må skrives inn manuelt og stikkeren må huske radiusen på den krumme linja. Kabelanlegg bruker dette, men for VA er det rette linjer.

Krumme kurver fra modell kan være vanskelig å bruke ute. Også “free-form” kurver (nurbs, bezier etc) i fagmodell kan kun approksimeres med rette linjesegmenter på oppmålingsutstyret. Oppfatningen er at utstyret henger litt etter med muligheter enn hva som brukes i modellen (av geometri). Det blir påpekt at kjennskap til hvordan krumme linjer kan approksimeres med rette segmenter under produksjon av stikningsdata gjøres av personer med riktig kompetanse.

Koding av data

- I hvilken grad kodes innmålinger i felt etter objektkodelista til Håndbok 138/prosjektet eller SOSI-koding?
- Bevarer de ulike formatene kodingen underveis?
- Leveres ferdig kodet data til byggherre etter objektkodelista?

Det er litt opp til den enkelte stikker om han eller hun koder data i felt eller etterpå. Noen gjør det ikke i det hele tatt og filnavnet forklarer hva som er målt inn i stedet. Ferdige innmålinger til geometrisk kontroll kodes etter prosjektets egen objektkodeliste. Konvertering fra oppmålingsutstyrs format til andre format bevarer koding underveis. Det er i dag forskjellig kode om det er topp eller bunn rør på innmåling som leveres, men modellen i utgangspunktet er kodet uten hensyn til dette. I lagnavnet kan da UK eller OK rør spesifiseres. Noen stikningslinjer til rør kan for eksempel være bunn rør, men innmålingen av ferdig lagt rør er foretatt på toppen av røret.

Produksjon av stiknings- og maskinstyringsdata generelt i dag

- Beskriv og/eller illustrer arbeidsflyten for hvordan terrengmodeller til konstruksjoner eller gravegrop lages?
- Er det noen spesielle utfordringer ved modellen når stikningsdata skal lages (datauttrekk av punkt, linjer/kurver, overflater, terrengmodeller)?
- Opplever du modellene som tilstrekkelige for å produsere stiknings- og maskinstyringsdata for prosjektet?
- Hva mangler modellene av egenskaper som du ønsker deg?
- Hvordan håndterer dere revisjoner på modell?

For å lage en triangulert flate (terrengmodell) for betongkonstruksjoner må solids splittes opp i linjer som danner utgangspunkt for knekklinjer. Hvis det er utsparinger i konstruksjonen, dvs. to punkter i planet, men ulik z-koordinat (et slags overheng) vil trianguleringen feile. Disse punktene må da flyttes manuelt noen mm i grunnriss. Dette er en tidkrevende oppgave og må i stor grad gjøres på nytt hver gang det kommer revideringer. For flater til bru er det gjerne en free-form linje som avgrenser overflaten og denne må deles opp i mange små, rette segmenter og det må fortettes med flere triangler under trianguleringen for å få en god nok approksimering mot overflaten. Bezim ønsker 2 nye lag i modell, triangulert OK og UK som en del av modellgrunnlaget. For å lage gravegrop der det ikke leveres må underkant betongkonstruksjon benyttes og forskyves nedover (endre Z til lavere verdi).

Punkter og rette linjer er enkelt å ta ut, kurver og særlig “free-forms” kurver er vanskelig å ta ut uten riktig verktøy og kompetanse/erfaring for å lage stikningsdata. For overflater er krumme flater vanskelig. En del gravegroper kan være bygget opp

av sær geometri (Smart Surface etc) som ikke er triangler og vanskelig uten erfaring å gjøre noe med. Det påpekes at mer triangler enn sær geometri i modellene er ønsket. Det er nok av egenskaper i lagnavnet i dag, i følge de stikningsansvarlige.

Modellen for betongkonstruksjoner er vanskelig (se over og om triangulering). Modellen for VA og belysning er veldig enkelt med hensyn til geometri og produksjon av stikningsdata. Generellt er modellene i dag tilstrekkelig for å lage stikningsdata med egne stikningsdatalag, men det krever mye erfaring og kompetanse hvis geometrien er mer avansert enn punkt og rett linje. Den vanlige stikkeren kan klare det enkle i dag, men ikke det avanserte.

Produksjon av “as-built” dokumentasjon generelt i dag

- Beskriv og/eller illustrer arbeidsflyten for hvordan innmålingsdata av konstruksjoner brukes til geometrisk kontroll og as-built dokumentasjon.
- Opplever du håndtering av innmålinger fra stikkeren er tilfredsstillende i dag?
- Hvordan ser du for deg at et program/system kan håndtere dette for deg?

Innmålingsdata fra for eksempel en betongkonstruksjon(3D-modell) eller armering (ikke 3D-modell, blir en punktsky og denne kontrolleres opp mot terrengmodellen ved å beregne avstand fra punkt til overflate (terrengmodell)) og se om avstanden er innenfor angitt toleranse for konstruksjon. For armering sjekkes det at overdekningen er innenfor spesifisert toleranse til overdekning. Avvik (utenfor toleranser) leveres det avviksrapport på.

I dag er slik at det er opp til stikkeren å kode innmålingen underveis i innmålingen eller i etterkant. Det er mest vanlig at innmålingsfilen for dagen splittes opp i enkeltfiler, for eksempel innm-ok-degplate24 eller innm-ok-rør400-deg, som inneholder kun innmålinger av den enkelte ting. Dette er metoden i dag. Innlevering av innmålinger til byggherre må ha temakode.

En utfordring i dag er at byggherren ønsker kun innmålinger på betongplate 24 og da må punkter utenfor modellen fjernes manuelt før beregning av avvik fra terrengflate gjøres. De ønsker ikke innmålinger fra det som er tidligere bygget og levert geometrisk kontroll på, for eksempel på rør: legges 10 m og det måles inn før det så legges 10 m til, vil de ikke ha måling på alle 20 m. Det er kun fra der det ble målt forrige gang til slutten av siste 10 m. I dag må det manuell filtrering til på dette og flere funksjoner for å velge ut områder eller punkter i nærheten som ikke er godkjent (eks røret over, antar første 10 m er godkjent).

Innmålinger lages som egen .dwg-fil som kodes med samme objektkode som levert data og leveres byggherre/konsulent. Filnavn er da SBXYZ + dato og objektkode. Innmålinger kan da sees sammen med originale modeller og gir en visuell inspeksjon av som bygget opp mot prosjektert modell.

Tillegg B

Program

B.1 geo2pgsql

Programmet består av to klasser, *Georeader* og *Geo2pgsql*. Førstnevnte leser geometri og egenskaper i innmålingsfilene. Sistnevnte klasse oversetter geometri fra innmålingsfila til WKT-format og sammen med egenskapene lages en SQL-setning som skrives til en fil. Skriptet kjøres fra en terminal og brukeren må gi skriptet argumenter som koordinatsystemkode, filnavn til innmålingsfil og tabellnavn for punkter og linjer i databasen. Skriptet gir mulighet for å lese 3- eller 4-dimensjonale koordinater.

```
#!/usr/bin/env python
```

```
'''
```

```
Read Svensk Byggnads Geodesi (SBG) .geo coordinate document for
converting to extended well-known text (EWKT) and sql
insert statements to a specific table.
```

```
KNOWN LIMITATIONS:
```

- no support for circle segments (curves)
- validation of input arguments

```
HOW TO USE:
```

```
python csvgeo.py [options]
```

```
Options:
```

- f Filename: path or filename to geo document. Required.
- e EPSG: EPSG coordinate code (default: 0). Required.
- l Location: schema.tablename for lines
(default: public.innm_linje).
 Not required.
- p Location: schema.tablename for points
(default: public.innm_punkt). Not required.
- o Output: path to output location (default: same location


```
as input file + .sql ending added). Not required.
- m      Mode: read M-coordinate (default: True). Not Required.
- d      Date: YYYY-MM-DD. Required.
```

Complete example:

```
python csvgeo.py -f path/to/file.geo -e 25832 -l public.innm_linje
-p public.innm_punkt -o path/to/output.sql -m False -d 2013-03-30
```

```
'''
```

```
__author__ = 'Eirik Oulie Rosbach'
__email__ = 'eirik.rosbach@student.umb.no'
__version__ = '0.2.31'
__copyright__ = "Copyright (c) 2013 Eirik Oulie Rosbach"
```

```
import os
import re
import csv
import argparse
```

```
class Georeader(object):
```

```
    '''
```

```
    Create a reader for SBG geo file format
    (SBG Object Text - Coordinate Document)
```

```
    '''
```

```
    def __init__(self, filename = None):
```

```
        '''
```

```
        Create a reader object.
```

```
        filename:          filename or path to file
        file_header:       header of file
        point_list:        container for lines
        line_list:         container for points
        attribute_list:    container for attributes
```

```
        '''
```

```
        self.filename = filename
        self.file_header = []
        self.point_list = []
        self.line_list = []
        self.attribute_list = []
```

```
        # Start reading after object is created
```

```

self.main()

def main(self):
    '''
    Read line by line from csv SBG Geo coordinate document and
    parse PointList and LineList.

    '''

    with open (self.filename, 'rb') as csvfile:
        self.csvreader = csv.reader(csvfile, delimiter = ',')

        for row in self.csvreader:
            if re.match('FileHeader', row[0]):
                if not re.search('SBG Object Text v2.01', \
                    row[0])
                and not re.search('Coordinate Document', row[1]):
                    break
                else:
                    self.read_file_header()
            elif re.match('PointList', row[0]):
                self.point_list = self.read_point_list()
            elif re.match('LineList', row[0]):
                self.line_list = self.read_line_list()
            elif re.match('AttributeList', row[0]):
                self.attribute_list = self.read_attribute_list()

def read_file_header(self):
    '''Read FileHeader and FileInfo.'''

    row = ['']

    while not re.match('end', row[0]):
        row = self.csvreader.next()
        if re.search('Application', row[0]):
            self.file_header.append(['Application', row[1]])
        elif re.search('SerialNo', row[0]):
            self.file_header.append(['SerialNo', row[1]])
        elif re.search('Author', row[0]):
            self.file_header.append(['Author', row[1]])
        elif re.search('Company', row[0]):
            self.file_header.append(['Company', row[1]])
        elif re.search('Description', row[0]):
            self.file_header.append(['Description', row[1]])
        elif re.search('Coordinate System', \
            row[0]):

```

```

        self.file_header.append(['Coordinate System', \
            row[1]])

def read_point_list(self):
    '''Read a PointList containing points.'''

    point_list = []

    row = self.csvreader.next()
    if re.search('LineList', row[0]): # Empty PointList
        point_list = None
        self.line_list = self.read_line_list()
    else:
        while not re.search('end', row[0]):
            row = self.csvreader.next() #
            if re.search('Point', row[0]):
                if len(row[6]) == 0: # st.dev may be empty
                    row[6] = 0
# Format: (pointname),Y (north), X (east), Z,
#object code, C/R point, st. dev
                point_list.append([row[0].strip("\t"), \
                    float(row[2]), float(row[1]), \
                    float(row[3]), row[4], row[5], row[6]])

    return point_list

def read_line_list(self):
    '''
    Read LineList containing lines built up from a PointList.
    '''

    row2 = ['']
    line_list = []

    row = self.csvreader.next()
    if re.search('AttributeList', row[0]): # Empty LineList
        line_list = None
        self.read_attribute_list()
        return line_list
    else:
        while not re.match('end', row[0]):
            row = self.csvreader.next()
            if re.search('Line', row[0]):
                closed_line = self.check_line_closed(row)
                line_name = row[0][7 : -1] # i.e. \tLine "1"
                line_attribute = row[2]

```

```

        while True:
            row2 = self.csvreader.next()
            if re.search('PointList', row2[0]):
                points = self.read_point_list()
# For closed lines append first point to end
                if closed_line is True:
                    points.append(points[0])
                    line_list.append((line_name, \
closed_line, \
line_attribute, \
points))
                else:
                    line_list.append((line_name, \
closed_line, line_attribute, points))
            # End of line
            elif re.search('end', row2[0]):
                break

    return line_list

def check_line_closed(self, line2check = ''):
    '''Check if a line is closed or not,
    i.e. (same start and end point'''

    # line name, 1 = closed; otherwise empty
    if line2check[1] == '1':
        return True
    else:
        return False

def read_attribute_list(self):
    '''Read attribute_list'''

    return NotImplemented

class Geo2pgsql(object):
    '''
    Read an SBG coordinate document and return geometry,
    attributes and theme code in WKT format as sql insert
    statements. Write sql statements to sql file. This script
    only handles attributes designated for a predefined
    table.

    '''

    def __init__(self, filename = None, epsg = 0, mode = True, \

```

```

        output = None , locationl = 'public.innm_linje', \
locationnp = 'public.innm_punkt', date = None):
    '''
    Create an object

    filename:          filename
    epsg:              epsg coordinate system code
    mode:              read m-coordinate value
    location_lines:    schema.tablename in database for lines
    location_points:   schema.tablename in database for points
    points2write:      container for sql statements for points
    lines2write:       container for sql statements for lines
    output:            path to output sql file
    date:              date foramttd in YYYY-MM-DD

    '''

    self.filename = filename
    self.epsg = epsg
    self.mode = mode
    self.location_lines = locationl
    self.location_points = locationnp
    self.points2write = []
    self.lines2write = []
    self.output = output
    self.date = date

    self.main()

def main(self):
    '''Create geo file reader and parse points and lines
    to pgsqformat.'''

    # Split path and filename of file to process
    dirname, filename = os.path.split( \
    os.path.abspath(self.filename))
    abspath_output = os.path.abspath(self.output)

    # Replace .geo file extension with .sql
    filename = filename[0: -3] + 'sql'

    # Check if input and output directory are the same
    if abspath_output == dirname:
        self.output = os.path.join(dirname, filename)
    else:
        self.output = os.path.join(abspath_output, filename)

```

```

print 'Processing: {0}'.format(self.filename)

# Create a georeader object
georeader = Georeader(filename = self.filename)

# Extract points, lines and author from header
self.points = georeader.point_list
self.lines = georeader.line_list
self.header = georeader.file_header
author = self.header[2][1]

# Convert points and lines if not empty
if not georeader.point_list is None:
    self.point2pgsql()
if not georeader.line_list is None:
    self.line2pgsql()

# Write output to .sql file
self.write2pgsql()

try:
    nbr_points = len(self.points)
except TypeError:
    nbr_points = 0

try:
    nbr_lines = len(self.lines)
except TypeError:
    nbr_lines = 0

print 'Completed! {2} lines and {1} points treated.\
\nOutput: {0}'.format(self.output, nbr_points, nbr_lines)

def point2pgsql(self):
    '''Convert pointlist from geo to sql string.'''

    for i in xrange(len(self.points)):
        point_name = self.points[i][0]
        point_code = self.points[i][4]
        if len(point_code) == 0:
            point_code = '0000000000' # empty linecode attribute
        geom = self.point2wkt(self.points[i])
        sql = """INSERT INTO {0}(id, innm_dato, objekt_navn, \
        temakode_temakode, geom) VALUES \
        (nextval('innm_id_seq'), '{1}', '{2}', {3}, {4});"""\.

```

```

        format(self.location_points, self.date, \
              point_name[7: -1], point_code[0: 6], geom)

        self.points2write.append(sql)

def point2wkt(self, point):
    '''Convert SBG geo point to wkt equivalent string'''

    temp = ''

    if self.mode is not True:
        temp += "POINT({0} {1} {2})" .\
              format(point[1], point[2], point[3])
    else:
        temp += "POINT({0} {1} {2} {3})" .\
              format(point[1], point[2], point[3], point[6])

    return "ST_GeomFromText({0}, {1})".format(temp, self.epsg)

def line2pgsql(self):
    '''Convert linelist geo to to sql string.'''

    for i in xrange(len(self.lines)): # For a single line
        current_line = self.lines[i]
        line_name = current_line[0]
        line_code = current_line[2]
        if len(line_code) == 0:
            line_code = '0000000000' # empty linecode attribute
        points = current_line[3]
        temp = ''

        for j in xrange(len(points)):
            if self.mode is not True:
                points_in_line = '{0} {1} {2}' .\
                      format(points[j][1], points[j][2], points[j][3])

            else:
                points_in_line = '{0} {1} {2} {3}' .\
                      format(points[j][1], points[j][2], \
                              points[j][3], points[j][6])

            temp += (points_in_line + ', ')

        # Remove extra whitespace and ', ' in the end
        geom = "ST_GeomFromText('LINESTRING({0})', {1})".format(\
              temp[0: -3], self.epsg)

```

```

        sql = """INSERT INTO {0}(id, innm_dato, objekt_navn, \
        temakode_temakode, geom) VALUES \
        (nextval('innm_id_seq'), '{1}', '{2}', {3}, {4});""".\
        format(self.location_lines, self.date, \
        line_name, line_code[0: 6], geom)

        self.lines2write.append(sql)

def write2pgsql(self):
    '''Write sql insert statements to file.'''

    with open(self.output, 'w') as outfile:

        # Write points and lines if not empty
        if not len(self.points2write) == 0:
            for j in xrange(len(self.points2write)):
                outfile.write(self.points2write[j])
                outfile.write('\n')

        if not len(self.lines2write) == 0:
            for k in xrange(len(self.lines2write)):
                outfile.write(self.lines2write[k])
                outfile.write('\n')

def main():
    '''Command line argument parser.'''

    parser = argparse.ArgumentParser(\
        description = 'Convert SBG geo to postgres')
    parser.add_argument('-f', '--filename', dest = 'filename', \
        help = 'Path to filename.', required = True)
    parser.add_argument('-e', '--epsg', dest = 'epsg', \
        help = 'EPSG CODE.', required = True)
    parser.add_argument('-l', '--ll', dest = 'locationl', \
        help = 'schema.tablename', required = False)
    parser.add_argument('-p', '--lp', dest = 'locationp', \
        help = 'schema.tablename', required = False)
    parser.add_argument('-o', '--output', dest = 'output', \
        help = 'Output filename.', required = False)
    parser.add_argument('-m', '--mode', dest = 'mode', \
        help = 'Coordinate M True/False.', required = False)
    parser.add_argument('-d', '--date', dest = 'date', \
        help = 'YYYY-MM--DD', required = True)

    args = parser.parse_args()

```



```

Geo2pgsql(filename = args.filename, \
           output = args.output, \
           locationl = args.locationl, \
           locationp = args.locationp, \
           epsg = args.epsg, \
           mode = args.mode, \
           date= args.date)

if __name__ == '__main__':

    main()

```

B.2 Geometrisk kontrollrapport

Skriptet etablerer en kobling til en PostGIS database og spør brukeren om id til objekt, navn på tabell, søkeradius i meter, temakode og hvem som har målt inne. Skriptet finner geometri og egenskaper for en va-ledning og et va-kumlokk og splitter nødvendige linjer og beregner avstand fra prosjektert mot innmålt. For objekteter som er innenfor en oppgitt tolereanse endres objektet i fagmodellen til som bygget. Når objektet er bygget oppdateres også i databasen. Resultatet fra spørringene skrives til fil.

```

#!/usr/bin/env python

'''
Connect to PostGIS database to query and return data to perform
geometric control of surveyed objects against projected objects.
Write result from query to a report. Update status of projected
objects in database.

'''

__author__ = 'Eirik Oulie Rosbach'
__email__ = 'eirik.rosbach@student.umb.no'
__version__ = '0.2.0'

import csv
import psycopg2 as ps
from datetime import date

class Grap(object):
    '''
    Grap - short for 'geometrisk rapport'

    '''

```

```

def __init__(self, host, dbname, user, password, port):
    '''
    Create an object.

    port:            port number
    password:        user password in database
    user:            username in database
    dbname:          databse name
    host:            host (local or remote)
    tolerance:       tolerance to built objects in meters

    '''

    self.port = port
    self.password = password
    self.user = user
    self.database = dbname
    self.host = host
    self.tolerance = 0.10

    self.main()

def main(self):
    '''
    Run program to grab data and perform geometric controll.
    '''

    # Create a connection and cursor object
    self.dbconnection = ps.connect(host = self.host, \
                                   dbname = self.database, user = self.user, \
                                   password = self.password, port = self.port)

    self.cursor = self.dbconnection.cursor()

    # Grab data from va_ledning and surveyed lines
    data_ledning, object_id_ledning, table_ledning = \
        self.grap_va_pipe()
    self.write_report_pipe(data_ledning, \
                           table_ledning, object_id_ledning)

    # Grab data from va_kum_lokk and surveyed lines
    centroid_kum, data_kum, id_kum, table_kum = \
        self.grap_va_manhole()
    self.write_report_manhole(data_kum, centroid_kum, \
                              id_kum, table_kum)

```

```

# Close connection to database
self.dbconnection.close()
print 'Connection closed!'

def get_user_tokens(self):
    '''Get search input from user.'''

    print '*' * 80
    object_id = eval(raw_input('ID paa objekt i fagmodell: '))
    temakode = eval(raw_input(
        ('Temakode paa objekt i innmaalinger: '))
    table = raw_input('Navn paa tabell i fagmodell: ')
    surveyed_by = \
    raw_input('Navn eller forkortelse paa hvem som har maalt: ')
    dist = eval(raw_input(
        'Soekedistanse i meter rundt angitt geoemtri i fagmodell: '))

    return object_id, temakode, table, surveyed_by, dist

def grap_va_pipe(self):
    '''Perform grap on a specified va_ledning'''

    object_id, temakode, table, surveyed_by, dist = \
        self.get_user_tokens()

    sql = """select  foo.id, foo.objekt_navn, foo.innm_dato,
foo.temakode, foo.path,
ST_Distance(foo.rp, foo.geom) as diff2d,
ST_3DDistance(foo.rp, foo.geom) as diff3d,
ST_AsText(foo.geom)
from (select ST_3DClosestPoint(v.stiknings_linje,
(ST_DumpPoints(i.geom)).geom) as rp,
(ST_DumpPoints(i.geom)).*, i.id, i.objekt_navn,
i.innm_dato, i.temakode_temakode as temakode,
i.godkjent_grap, i.avviksmelding_referanse,
i.beskrivelse
from innm_linje i, {0} v
where
ST_3DDWithin(i.geom, v.stiknings_linje, {2})
and v.id={3}
and i.temakode_temakode={1}
and i.innm_av='{4}') as foo;""".\
        format(table, temakode, dist, object_id, surveyed_by)

    self.cursor.execute(sql)

```

```

print self.cursor.query
print self.cursor.statusmessage
data = self.cursor.fetchall()

# Update built date on object
self.update_built_date_objects(table, \
    object_id, str(date.today()))

return data, object_id, table

def write_report_pipe(self, data2write, table, object_id):
    '''Write result from query'''

    temakode = str(data2write[0][3]) # fetch temakode
    date_report = str(date.today()) # fetch todays date
    filename = '{0} {1} va_ledning GRAP xyz.csv'.\
        format(date_report, temakode)

    # Containers for differences in 2d and 3d
    data2d = []
    data3d = []

    with open(filename, 'wb') as csvfile:
        csvwriter = csv.writer(csvfile, delimiter=';')
        csvwriter.writerow(['ID', 'OBJEKT_NAVN', \
            'INNM_DATO', 'TEMAKODE', 'PATH', \
            'DIFF2D', 'DIFF3D', 'GEOMETRY'])

        for i in xrange(len(data2write)):
            row = list(data2write[i])
            row[5] = round(float(row[5]), 3)
            row[6] = round(float(row[6]), 3)
            data2d.append(row[5])
            data3d.append(row[6])

            csvwriter.writerow(row)

    # Compute average differences to check if need
    # to update status
    avg2d = sum(data2d) / float(len(data2d))
    avg3d = sum(data3d) / float(len(data3d))

    print 'Gj.snitt avstand i 2D: {0} og 3D: {1}'.\
        format(avg2d, avg3d)

    if avg2d < self.tolerance and avg3d < self.tolerance:

```

```

        self.update_object_status_model(table, object_id)
    else:
        object_id_s = row[0]
        diff2d = avg2d
        diff3d = avg3d
        self.create_discrepancy_message(object_id, \
            object_id_s, diff2d, diff3d)

def grap_va_manhole(self):
    '''Perform grap on a specified va_kum'''

    object_id, temakode, table, surveyed_by, dist = \
        self.get_user_tokens()

    # Center point top manhole
    sql = """select i.id, i.objekt_navn, i.innm_dato,
i.temakode_temakode as temakode,
ST_Distance(ST_Centroid(i.geom), v.senterpunkt_lokk)
as diff2d,
ST_AsText(ST_Centroid(i.geom)) from innm_linje i, {0} v
where
ST_DWithin(i.geom, v.senterpunkt_lokk, {2})
and v.id={3}
and i.temakode_temakode={1}
and i.innm_av='{4}';""".\
        format(table, temakode, dist, object_id, surveyed_by)

    self.cursor.execute(sql)
    centroid = self.cursor.fetchall()

    # Plan view of manhole
    sql2 = """select foo.id, foo.objekt_navn, foo.innm_dato, \
foo.temakode, foo.path,
ST_Distance(foo.rp, foo.geom) as diff2d,
ST_AsText(foo.geom) from
(select ST_ClosestPoint(v.modell_geometri,
(ST_DumpPoints(i.geom)).geom) as rp,
(ST_DumpPoints(i.geom)).*, i.id, i.objekt_navn, i.innm_dato,
i.temakode_temakode as temakode,i.godkjent_grap,
i.avviksmelding_referanse, i.beskrivelse from innm_linje i,
{0} v where ST_DWithin(i.geom, v.senterpunkt_lokk, {2})
and v.id={3} and i.temakode_temakode={1}
and i.innm_av='{4}') as foo;""".\
        format(table, temakode, dist, object_id, surveyed_by)

    self.cursor.execute(sql2)

```

```

print self.cursor.query
print self.cursor.statusmessage
data = self.cursor.fetchall()

# Update built date on object
self.update_buildt_date_objects(\
    table, object_id, str(date.today()))

return centroid, data, object_id, table

def write_report_manhole(self, data2write, \
    centroid2write, object_id, table):
    '''Write result from query'''

    temakode = str(data2write[0][3]) # Fetch temakode
    date_report = str(date.today()) # Fetch todays date
    filename = '{0} {1} va_kum GRAP xyz.csv'.\
        format(date_report, temakode)

    with open(filename, 'wb') as csvfile:
        csvwriter = csv.writer(csvfile, delimiter=';')

        # For manhole
        csvwriter.writerow(['ID', 'OBJEKT_NAVN', \
            'INNM_DATO', 'TEMAKODE', \
            'PATH', 'DIFF2D', 'GEOMETRY'])

        for i in xrange(len(data2write)):
            row = list(data2write[i])
            row[5] = round(float(row[5]), 3)

            csvwriter.writerow(row)

        # For centroid to manhole
        row = list(centroid2write[0])
        row[4] = round(float(row[4]), 3)
        row.insert(4, 'Null')
        csvwriter.writerow(row)

    print 'Differanse senter kum og innmaalt: {0}'.format(row[5])

    # Check if need to updte object status
    # (built within tolerance)
    if row[5] < self.tolerance:
        self.update_object_status_model(table, object_id)
    else:

```

```

        object_id_s = row[0]
        diff2d = avg2d
        diff3d = 0
        self.create_discrepancy_message(\
            object_id, object_id_s, diff2d, diff3d)

def update_object_status_model(self, table, object_id):
    '''Update status to 'as built' for objects
    built within tolerance.'''

    sql = """update {0} set id_objekt_status=1
    where id ={1};""".format(table, object_id)

    print 'Oppdaterer status til som bygget (kode 1)'

    # Execute and commit changes
    self.cursor.execute(sql)
    print self.cursor.query
    print self.cursor.statusmessage
    self.dbconnection.commit()

def update_buildt_date_objects(self, table, object_id, \
    date_built):
    '''Update built date on projected object.'''

    sql = """update {0} set bygget_dato='{1}'
    where id={2};""".format(table, date_built, object_id)

    # Execute and commit changes
    self.cursor.execute(sql)
    print self.cursor.query
    print self.cursor.statusmessage
    self.dbconnection.commit()

def create_discrepancy_message(self, \
    object_id_f, object_id_s, diff2d, diff3d):
    '''Create report for objects built outside tolerance.'''

    print 'Lager avviksmelding paa objekt {0}'.\
        format(object_id_f)

    with open('Avviksmelding.txt', 'w') as outfile:

        outfile.write('Avviksmelding')
        outfile.write('\n')
        outfile.write('*' * 50)

```

```
outfile.write('\n')
outfile.write('Objekt-id i fagmodell: {0}'.\
  format(object_id_f))
outfile.write('\n')
outfile.write('Objekt-id i innmaalinger: {0}'.\
  format(object_id_s))
outfile.write('\n')
outfile.write('Differanse i 2D (plan): {0}'. \
  format(round(float(diff2d), 3)))
outfile.write('\n')
outfile.write('Differanse i 3D: {0}'. \
  format(round(float(diff3d), 3)))
outfile.write('\n')
outfile.write('*' * 50)
```