

Metodologi-dokumentasjon

Dato	bidragsytte
25.03.2025	Virginia Antonijevic, Elling Aronsen Oftedal og Ida Hope Barth

Introduksjon

Denne metodologidokumentasjonen beskriver prosessen og metodene som ble brukt for å utvikle en automatisk generaliseringsløype for veidata fra Nasjonal Vegdatabank (NVDB) til N100K, et kartdatasett tilpasset en målestokk på 1:100 000. Målet med prosjektet var å oppdatere eller skifte ut eksisterende modell med en modell som skulle gi et bedre kartografisk resultat enn forrige versjon lagd i 2018.

I den første versjonen av N100K ble N50K benyttet som kilde for å forenkle harmoniseringen med de øvrige målestokkene. Dette valget ble gjort med tanke på at N50K skulle være den primære kilden der det var mulig. Imidlertid oppstod det utfordringer med å skille mellom viktige og mindre viktige veier, noe som er en grunnleggende forutsetning for automatisk generalisering. Resultatene fra N50K oppfylte ikke de ønskede kravene, og det ble derfor nødvendig å se etter andre kilder for å oppnå bedre resultater.

I forbindelse med oppdateringen av modellen for 2024/2025 ble NVDB, sammen med N50K og Elveg, vurdert som potensielle kilder for generaliseringen av veiene til N100K. NVDB er et nasjonalt system i Norge for lagring, forvaltning og tilgjengeliggjøring av vegrelaterte data. Det inneholder svært detaljert informasjon om det norske veinettet og er et viktig verktøy for vei- og transportforvaltning i Norge. Ved å bruke NVDB som kilde, kunne vi dra nytte av dens detaljerte data og hierarkiske struktur for å utvikle en robust og nøyaktig generaliseringsmodell.

Denne dokumentasjonen vil gå i detalj på de metodene og teknikkene som ble brukt, inkludert valg av kilder, hierarkisk strukturering, og oversettelse fra NVDB-språk til N100K. Videre vil vi diskutere de ulike verktøyene og parameterne som er vurdert og enten avvist eller tatt med i den ferdige modellen, samt tiltak for å sikre stabilitet og nøyaktighet.

Kilde

I arbeidet med å utvikle en automatisk generaliseringsløype for veidata fra NVDB til N100K, vurderte vi flere potensielle kilder for generaliseringen. De viktigste kildene vi vurderte var N50K, NVDB og Elveg. Hver av disse kildene har sine fordeler og ulemper, og vi gjennomførte en grundig evaluering for å finne den mest egnede kilden for vårt formål.

N50K ble opprinnelig brukt som kilde i den første versjonen av N100K. Dette valget ble gjort for å forenkle harmoniseringen med de øvrige målestokkene. Imidlertid oppstod det utfordringer med å skille mellom viktige og mindre viktige veier, noe som er en grunnleggende forutsetning for automatisk generalisering. Resultatene fra N50K oppfylte ikke de ønskede kravene, og det ble derfor nødvendig å se etter andre kilder.

NVDB og Vegtrase

Etter en grundig vurdering av NVDB og Elveg, valgte vi å bruke vegtrase fra NVDB som utgangspunkt for generaliseringen. Vegtrase representerer en konstruert senterlinje som den fysiske veien ikke nødvendigvis følger, spesielt ved fysisk adskilte kjørebaneer. Dette valget ble gjort fordi vegtrase allerede er generalisert, noe som gjør generaliseringen lettere for oss. Det er viktig å merke seg at vegtrase eksisterer i NVDB og ikke i Elveg, og vi måtte derfor ta et uttrekk fra NVDB.

Vi anser at vegtrase passer godt som utgangspunkt for generaliseringen, da det gir oss en nøyaktig og brukervennlig representasjon av veinettet, samtidig som det forenkler prosessen med å skille mellom viktige og mindre viktige veier.

Hierarki

I arbeidet med å utvikle en automatisk generaliseringsløype for veidata fra NVDB til N100K, var det viktig å etablere et hierarki for veiene. Hierarkiet hjelper oss med å skille mellom viktige og mindre viktige veier, og sikrer at kartdataene gir en nøyaktig og brukervennlig representasjon av veinettet.

Vurdering av Hierarkier

Vi vurderte flere hierarkier for å finne den beste måten å skille mellom viktige og mindre viktige veier. De hierarkiene vi vurderte var vegkategori, funksjonellveiklasse og funksjonsklasse. Vegkategori, som egentlig beskriver hvem som eier veien, ble brukt i den første versjonen da vi hadde N50K som utgangspunkt. Imidlertid ga vegkategori spesielt i Oslo et dårlig skille mellom viktige og mindre viktige veier, og alt for mange veier ble beholdt under generaliseringen ved bruk av vegkategori.

Funksjonellveiklasse

Funksjonellveiklasse, som er basert på ruting, gir en verdi mellom 1 og 9 for hva som er viktige veier og mindre viktige veier. Spesielt i Oslo gir dette en god differensiering i veiviktighet. Viktigere kommunale veier skilles fra mindre viktige kommunale veier. Da Oslo er både en kommune og et fylke, og de aller fleste veiene er registrert som kommunale, er dette spesielt viktig.

Funksjonsklasse

Funksjonsklasse er et annet hierarki som også eksisterer i NVDB. Dette hierarkiet er primært brukt for å differensiere de viktigste veiene, som Europa- og riksveier. De aller fleste av disse veiene blir uansett tatt med i N100K, og vi ser derfor ikke nytten av å bruke dette hierarkiet. Det kan imidlertid være nyttig å se på en kombinasjon mellom funksjonellveiklasse og funksjonsklasse i fremtiden for å gi en enda bedre differensiering, spesielt hvis vi skulle bruke denne modellen til å generalisere videre til N250.

Viktigheten av Hierarki for ESRI-verktøy

Det er flere verktøy fra ESRI som krever hierarki for å gjøre en utvelgelse. Derfor er det viktig at hierarkiet er differensiert, selv om det kan være noen ulemper. Et godt differensiert hierarki sikrer at verktøyene kan utføre nøyaktige analyser og utvelgelser, noe som er avgjørende for å oppnå et

godt kartografisk resultat.

Det er imidlertid slik at verktøyene fungerer best med maksimalt 5 hierarkiverdier (Se dokumentasjon ESRI) og vi måtte gjøre en vurdering for å optimalisere grupperingen av verdiene. Siden vi ønsker å beholde de viktigste veiene i N100K og at de i utgangspunktet ikke ligger så nærme hverandre, har vi samlet alle de viktigste veiene i samme hierarki. Vi har testet ulike varianter av gruppering av funksjonellveiklasse og det gir litt ulikt resultat om man er i urbane strøk eller ikke. Etter mye testing har vi falt på at det som gir best resultat i Oslo er det vi bruker, siden hierarkiet har mer å si for områder der veinettet er tett og mange veier må vike for å tilpasses målestokken. Derfor er det viktig å skille mellom funksjonellveiklasse 5 og 6, som skiller ringveiene fra de resterende kommunale veiene i Oslo.

Funksjon ell- veiklasse	Beskrivelse	Hierarki verdi
0	De viktigste vegene i et vegnettverk, f.eks. motorveger.	1
1	De nest viktigste vegene i et vegnettverk, f.eks. større riksveger.	1
2	De tredje viktigste vegene i et vegnettverk, f.eks. mindre riksveger.	1
3	De fjerde viktigste vegene i et vegnettverk, f.eks. større fylkesveger.	1
4	De femte viktigste vegene i et vegnettverk, f.eks. mindre fylkesveger.	2
5	De sjette viktigste vegene i et vegnettverk, f.eks. de minste fylkesvegene og viktige kommunale veger.	2
6	De sjuende viktigste vegene i et vegnettverk, f.eks. kommunale veger.	3
7	De åttende viktigste vegene i et vegnettverk, f.eks. private og skogsbilveger.	4
8	De niende viktigste vegene i et vegnettverk, f.eks. veger det ikke er anbefalt å kjøre på, men mulig å kjøre på.	5
9	De minst viktige vegene i et vegnettverk, f.eks. veger det ikke er anbefalt å kjøre på, men mulig å kjøre på.	5

Det har også vist seg at vi må bruke "Thin Road network" i flere omganger for å få stabilitet, men også for å kunne bruke hierarkiet på en mer effektiv måte. Det er mange veier med ulik hierarki og reduksjonen fra 10 - 5 klasser, gjør at stier ikke blir differensiert og blir noe tett for 1:100 000. Derfor etter flere omganger med "Thin road network" med samme hierarki, blir det til slutt kjørt en "thin road network" kun for stiene, der alle kjørbare veier er låst. Dette fører til at vi også får generalisert stinettverket. Vi har prøvd å generalisere stinettverket først og så låst stiene for generalisering av veier i etterkant, men dette førte til at noen privatveier ble borte men stiene ble beholdt. Når vi generaliserer veinettet blir stiene aldri låst, men ved grundig generalisering av stiene låser vi veiene slik at stiene aldri trumfer en kjørbare vei.

Det kan diskuteres om man bør skille kjørbare vei-generaliseringen med stigeneraliseringen, men da mister vi nettverket og overgang fra vei til sti.

Sti-type	Hierarkiverdi
Traktorveg	1

Sti-type	Hierarkiverdi
Barmerksløyper	1
Merket sti - DNT	2
Merket sti - Andre	2
Umerket sti	3
Gang og Sykkelveg	4

N50K Stier

Hovedkilden er NVDB. Det er imidlertid slik at stier idag ikke er fullstendig i NVDB. Det gjøres en jobb for å få datasettet oppdatert og planen er at stinettet, samt barmarksløyper og traktorvei skal ligge og ajourføres i NVDB i framtiden. I dag derimot bruker vi N50K som kilde for sti, traktorvei og barmarksløyper og legger det sammen med kjørbareveier fra NVDB før vi starter generaliseringsrutinene.

Tilpassninger

For å få det datasettet vi ønsker må vi hente og sette sammen flere datasett, objekter og verdier. Dette gjøres i en FME-Løype.

NVDB

I førsteomgang har vi fått en tilpasset leveranse fra grunndata med utdrag av NVDB, men vil etter vært koble oss på API for å få fortløpende oppdateringer. Dette utdraget av NVDB inneholder følgende geometrier:

- FunksjonellVegklasse
- Motorveg
- vegsenterlinje med følgende egenskaper:

Typeveg = enkelBilveg, rundkjøringer og kanaliserteVeg
 Detaljnivå = vegtrase

Som en del av krevspesifikasjonen så kreves subtypekode som en egenskap i resultatet. NVDB har ikke subtypekode og dette kodes om i FME.

N50K

Vi henter ut alle ikke kjørbare veier fra N50K, siden N50K er den mest oppdaterte kilden til dette formålet.

- vegsenterlinje med følgende egenskaper:

OBJTYPE = Sti, Traktorveg, Barmarksløyper og GangogSykkelveg

Deretter kombineres funksjonellvegklasse og motorvegtype. Alle Europa- og Riksveier som ikke overlapper med Motorveg fra NVDB får MOTORVEGTYPE = ikke motorveg slik som kravspesifikasjonen krever.

Smidig tilnærming eller stabil leveranse

Opprinnelig var målet vårt å levere en versjon så snart vi hadde oppfylt de primære kravene for N100K vei. Dette fordi N100K vei ikke har blitt oppdatert på to år, og den forrige versjonen ikke fungerer med nyere versjoner av ArcGIS Pro. Det har imidlertid vist seg at alle mottakerne våre foretrekker en stabil leveranse, der utvelgelsen av veier ikke endrer seg vesentlig fra første til andre versjon. Etter å ha klargjort dette leveransekravet, har vi gjort en grundig testing for å finne en løsning som gir konsistente resultater.

ESRI-Verktøy

Basert på dokumentasjonen for ESRI-verktøyene har vi en klar forståelse av hva vi forventer av resultatene. Imidlertid fungerer de kartografiske ESRI-algoritmene som svarte bokser, og vi opplever at de ikke gir konsistente resultater. Dette er bekymringsfullt og utfordrende. Vi har vært i kontakt med ESRI sentralt, og de har forklart at algoritmene kan feile i de kartografiske verktøyene fra ESRI, uten å gi feilmelding. De har anbefalt oss å bruke små partisjoner, for å unngå denne problematikken. På bakgrunn av dette har vi prøvd å optimalisere modellen for å oppnå så konsistente resultater som mulig. I kravspesifikasjonen har vi angitt at ved to kjøringar med samme kildegrunnlag må minst 95 % av resultatene være geometrisk like.

Optimalisering

For å optimalisere stabiliteten til modellen har vi gjennomført grundige undersøkelser av følgende parametere:

- Partisjonsstørrelse
- Faste partisjoner eller skalerbare partisjoner
- Bufferavstand rundt partisjoner
- Flere kjøringar av ikke-deterministiske verktøy fra ESRI

Fremgangsmåte

Etter kontakt med ESRI-sentralt ble det klart at vi må gå for små partisjoner med færre objekter enn det som faktisk er oppgitt i dokumentasjonen til verktøyet. Etter litt testing er det klart at det er flere parametere som spiller en rolle for hvor stabil modellen kan bli. Vi har prøvd å avgrense de ulike parameterne for så å teste hver for seg. Det er imidlertid slik at alle må være til stede i en

hvis grad for å få ett konsistent resultat

- Det aller viktigste er å kjøre det "thin road network" verktøyet flere ganger. Hvorfor dette er nødvendig er vanskelig å svare på siden vi ikke har fullstendig oversikt over hva verktøyet faktisk gjør, siden dette ikke er dokumentert fra ESRI sin side. Det som imidlertid er dokumentert er at verktøyet "Thin road network" bør kjøres opp til flere ganger for å nærme seg ett ønsket deterministisk modell. Vi har prøvd å kjøre "thin road network" opp til 7 ganger. Kjører vi imidlertid "thin road network" kun en gang, vil uansett partisjon eller størrelse på buffer ikke gi oss konsistent resultat.
- Hvor mange veier eller vertices som kan prosseseres samtidig er ikke dokumentert, men med litt testing ser det ut til å være mellom ca. 6000 til 10 000 veier. Dette gjelder da både de som skal bli med videre og de veiene som er tatt med som kontekst for å gjøre generalisreen så lik som mulig. Det er her buffer avstanden kommer inn. Hadde vi ikke tatt med en buffer ville det alltid bli forskjellig generalisering i utkanten av partisjonen fordi de ville blitt behandlet som endelinjer og ikke en del av nettverket. Spørsmålet da er hvor stor bufferen må være for at veiene i partisjonen alltid vil bli behandlet likt. For å teste dette ble flere partisjoner kjørt flere ganger, da med mange omganger av "thin road network" som vi vet må til for å få ett konsist resultat. Ved en partisjon + buffer på ca. 8000 linjer, der buffern er 3000 m, ble resultatet helt likt ved flere kjøringar etter hverandre. En test vi kjørte med mindre buffer på 1500 m ble resultatet noe ulikt, men fortsatt under 5% endring.

Ettertanke

Etter optimaliseringen ble det en diskursjon om vi trenger faste partisjoner eller ikke. Modellen er satt opp slik at partisjonene blir beregnet ut i fra antall veier i kilden og er derfor i utgangspunktet ikke fast. Er det ingen endring vil heller ikke partisjonen endre seg, men når det er endringer vil partisjonene tilpasse seg endringene. Etter mange tester ser vi at en fast partisjon vil gi ett mer konsistent resultat, selv om det er vanskelig å si i hvor stor grad. Grunnlaget for diskursjonen er allikevel at så lenge kilden endrer seg så endrer også premisset for selve generaliseringen. Velger vi å ha faste partisjoner vil vi ikke ha kontroll om maksimal antall veier i partisjonene. Ut i fra denne diskursjonen ble det bestemt at vi prøver å optimalisere antall linjer/veier og buffer avstand i tillegg til antall omganger "thin road network". En test for ulike partisjoner vil bli tatt etter at alle de andre forholdene er optimalisert.

Oppbygging av modell

Målet for N100K veier er å vise veinettet og de viktige gjennomgående veiene. Det er fokus på at de viktigste veiene som motorveger, europaveier, riksveier skal med, men at kommunale og private er med for å gi inntrykk av tetthet i veinettverket.

I sammenligning med N50K må kryss og på- og avkjøringsramper forenkles, rundkjøringer enten forstørres eller fjernes, og den generelle tettheten av veier må reduseres.

Forenklet forklaring på modellen

Selve scriptet dokumenterer i detalj oppbyggingen av modellen, men for ikke

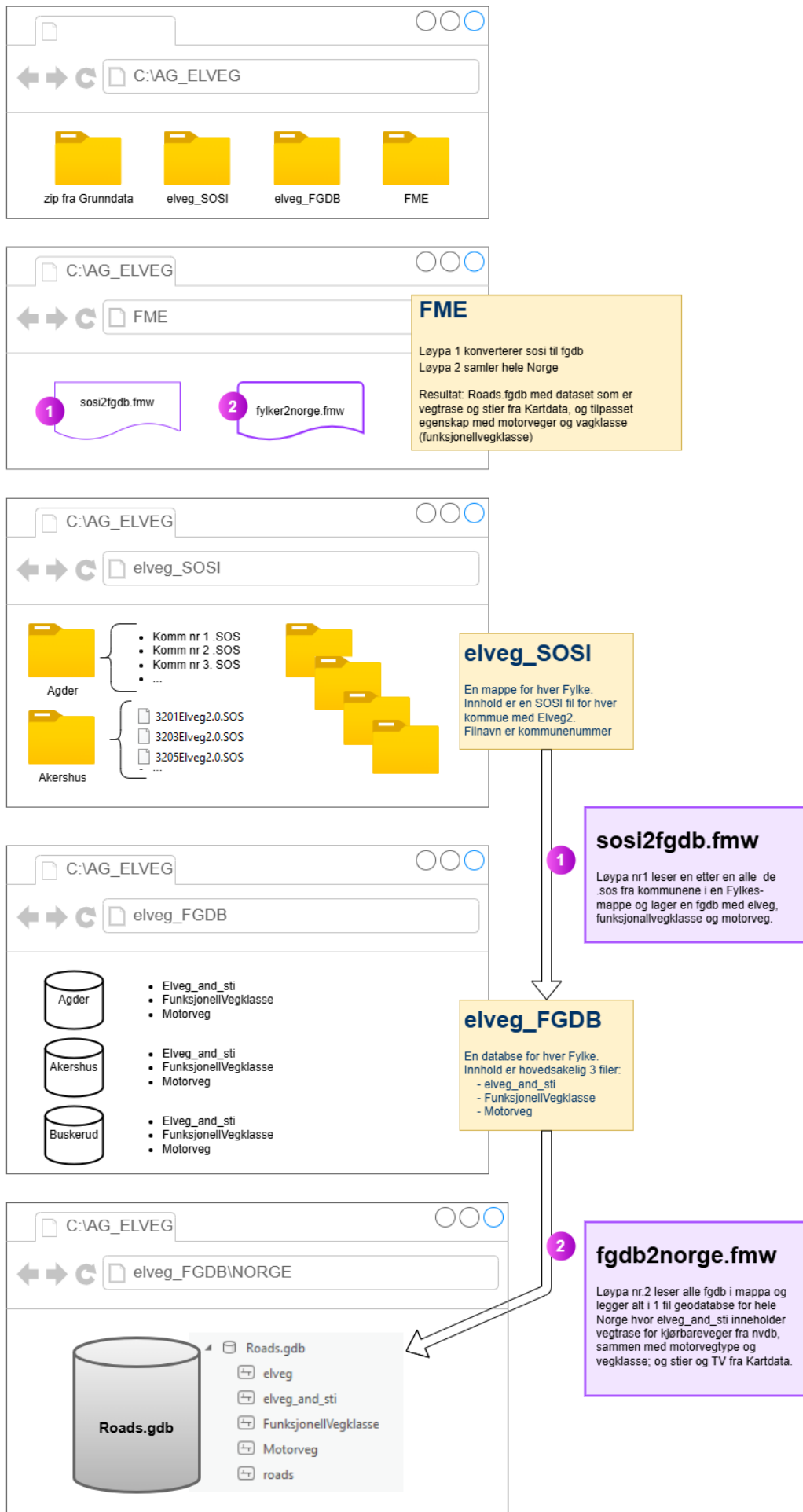
programmeringsdyktige kommer en forklaring på modellen under

Data tilpasning

Vi benytter to kilder som hentes og tilpasses for videre bruk i automatisk generalisering. Disse kildene inkluderer veinettet fra NVDB, som for øyeblikket leveres som en spesialleveranse fra grunndata, og N50K Sti og traktorveg. Tilpasningen av disse dataene utføres ikke ved dokumentert kode i GitHub, men vil bli lagt til der i ettertid.

I dag brukes FME-løype (ansvarlig for modell, Virginia Antonijeciv)

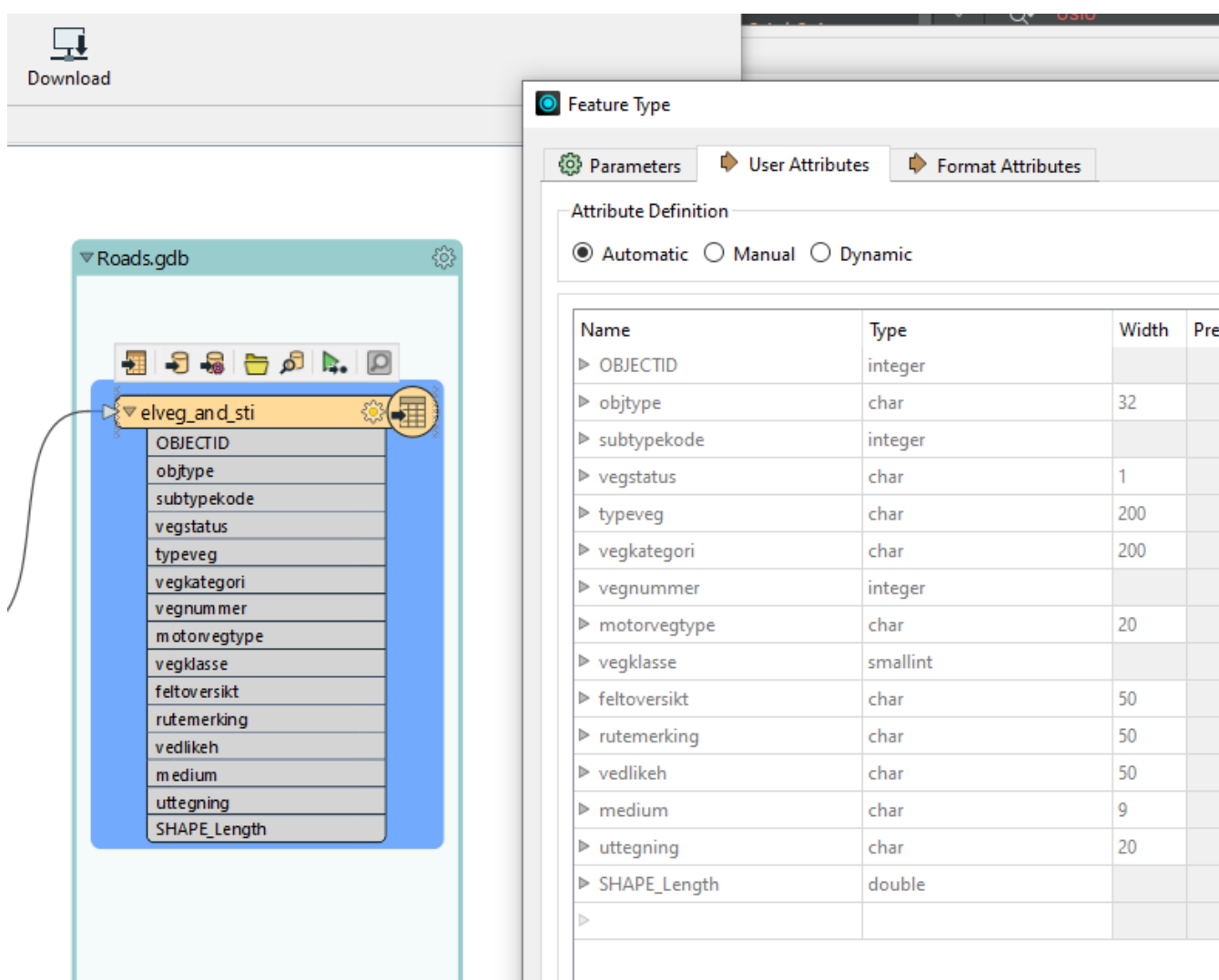
fra SOSI to fgdb



Beskrivelse av data tilpasningen

Første del innebærer en prosess hvor .zip-filer med kommunevis sosi-filer eksporteres fra Grunndata. Disse filene inneholder elvegdata på alle nivåer, inkludert vegtrase. Deretter pakkes filene ut til fylkesvise mapper, med en sos-fil for hver kommune. Videre konverteres sosi-filer til fgdb-format og kombineres med objekter fra Kartdata, og skaper en fgdb for hvert fylke med funksjonell vegklasse, motorveg og "elveg og sti". Fra elveg inkluderes kun kjørbare veger, ikke vegtyper som fortau og trapp, mens Kartdata bidrar med alt unntatt bilveger, som sti, traktorveg, barmarksløype og gang- og sykkelveg. Deretter settes hele landet sammen og kombinerer funksjonell vegklasse og motorvegtype, og skriver til fgdb som er input til AG. Til slutt utføres en geometrisjekk hvor veglinjen dissolveres på tvers av egenskapene og lager kryss etter medium på nytt. Noen egenskaper blir tilpasset for videre bruk.

Under er det et bilde av egenskapene til FGDBen "elveg_and_sti" som er utgangspunktet for den automatiske generaliseringsmodellen.



The screenshot shows the QGIS interface with the 'Feature Type' dialog open for the 'elveg_and_sti' layer. The 'Attribute Definition' tab is active, showing a table of attributes. The 'Automatic' radio button is selected under 'Attribute Definition'.

Name	Type	Width	Pre
▶ OBJECTID	integer		
▶ objtype	char	32	
▶ subtypekode	integer		
▶ vegstatus	char	1	
▶ typeveg	char	200	
▶ vegkategori	char	200	
▶ vegnummer	integer		
▶ motorvegtype	char	20	
▶ vegklasse	smallint		
▶ feltoversikt	char	50	
▶ rutemerking	char	50	
▶ vedlikeh	char	50	
▶ medium	char	9	
▶ uttegning	char	20	
▶ SHAPE_Length	double		
▶			

Generalisering

Valg av rekkefølge og bruk av ulike generaliseringsverktøy er basert på ESRI-anbefalinger for veg-generalisering og ved gjentatte tester. Dette er den overordent modellen for Vei-generalisering for N100K.

Verktøy	Hvorfor
Multipart to singelpart	For å få den rette oppdelingen av vegsegmenter.
Hierarki	For differansiering mellom viktighet av vei, må vi definere ett hierarki. For kjørbare veier blir hierarkiet bygd fra Funksjonellveiklasse, beskrevet tidligere i dokumentet. Siden Funksjonellveiklasse kun er for veier fra NVDB, brukes veikategori for å sette opp hierarkiet for alt untatt bilveger.
Remove small lines	Fjerner alle linjer under 100 m for å fjerne unødvendig smårusk
Colapse Road detail	Fjerner de fleste rundkjøringer og annet smårusk
Simplify	Forenkler linjene ved å fjerne unødvendige punkter i linjen.
Thin Road network: Vei	<p>Beholder veimønster men fjerner detaljene.</p> <p>Dette verktøyet ga i første omgang ulikt resultat for hver gang det kjørte og vi startet arbeidet med å stabilisere resultatet (beskrevet tidligere i dokumentet). Det ble derfor lagd en iterasjon som gjør flere operasjoner i flere omganger, helt til det ikke lenger skjer endring fra omgang til omgang. Under er rekkefølgen beskrevet:</p> <p>En omgang =</p> <ol style="list-style-type: none"> 1) Dissolve 2) Select & make permanent MEDIUM = U 3) Select & make permanent MEDIUM = T 4) Feature to Line (+ delete fields) 5) Append 6) Thin!
Thin Road network: Sti	Beholder stimønsteret men fjerner detaljene. Til slutt kjøres en tynning av stiene.
Smooth Line	For å gjøre vegsegmentene mer naturlige og mindre hakkete
Resolve Road Conflict	For å flytte vei vekk fra andre linjer med betydning som f.eks vannkontur eller jernbanelinje.
Dissolve	For å forenkle vegnettet før leveranse