Datamodellering av geografisk informasjon basert på UML som skjemaspråk

Steinar Høseggen, Geomatikk AS

Steinar Høseggen: Data modeling of geographic information based on UML as schema language

KART OG PLAN, Vol 66, pp. 218–224. P.O.Box 5003, N-1432 Ås, ISSN 0047-3278

This article provides an overview of the conceptual data modeling process and use of UML Static Diagrams to describe data models precisely. Furthermore the article describes how to make application models of geographical information using UML in conformity with the ISO 19100 series of standards, supported with an example taken from the tourism information area.

Key words: Conceptual data models, UML, geographical information, ISO 19000 standard

Steinar Høseggen, Geomatikk AS. E-mail: steinar.hoseggen@geomatikk.no

Innledning

UML – Unified Modeling Language – ble brukt som gjennomgående verktøy for beskrivelse av konsepter og modeller i ISO/TC 211 – Geographic information/ Geomatics. Standarden definerer viktige modeller for geometri, topologi, tid og metadata, som applikasjonsmodeller må benytte hvis de skal være konforme med ulike standarder i ISO 19100-serien.

UML er et omfattende språk som kan beskrive samme modell på flere måter. For å sikre at applikasjonene kan implementeres, foreligger ISO 19109 Rules for application schema og ISO 19103 Conceptual schema language, som spesifiserer regler for hvordan applikasjonsmodellen skal lages. Statens kartverk har videreutviklet reglene for å sikre at norske modeller blir likt oppbygd.

Jeg håper ved denne artikkelen å vise at de regler som er innført ikke er så langt unna vanlig praksis for informasjonsmodellering, og velger å introdusere problemstillingen generelt og vise hvilken betydning datamodellering har for å utvikle gode informasjonssystemer.

Konseptuell datamodellering

De viktigste komponentene som spesifiserer et informasjonssystem, er beskrivelse av systemets funksjoner og beskrivelse av de data som systemet skal håndtere. Datamodeller benyttes til beskrive datastrukturene som skal inngå. Det kan være datastrukturene i databasen eller datastrukturene for data som importeres til eller eksporteres fra informasjonssystemet.

For å komme fram til en god datamodell, kreves aktiv innsats fra fagpersonell som kjenner de oppgaver og den informasjon som informasjonssystemet skal håndtere.

Datamodellering er en top-down analyse som ender opp i en konseptuell beskrivelse av datastrukturene (konseptuell modell) som senere kan overtas av en datateknisk implementasjon. En konseptuell modell bør helst beskrives implementasjonsuavhengig. Følgelig er det ikke nødvendig at alle som deltar i datamodelleringsarbeidet, er datateknisk kyndig. Snarere er det snakk om en prosess hvor det deltar personell som kjenner det aktuelle fagområdet og personell som kan bruke det aktuelle modelleringsverktøyet.

Framgangsmåten ved datamodellering er generell i den forstand at den også er gyldig for geografisk informasjon. Man starter ved å avgrense hvilken type informasjon som skal inngå i informasjonssystemet. Noe pompøst sier man at datamodellering er å avbilde en avgrenset del av den virkelige verden (Universe of discourse) til en formell beskrivelse (datamodell).

Neste prosess å identifisere de ulike objekttyper innenfor denne avgrensningen. Eksempel: I et informasjonssystem for vann og avløp, er ledninger, koplingspunkter og ledningsnett typiske objekttyper.

Neste trinn i analysen er å identifisere og beskrive sammenhenger (relasjoner eller assosiasjoner) mellom de ulike objekttypene. Eksempler på assosiasjoner: I et VA-system kan en ledning være koplet til et koplingspunkt i hver ende. Et koplingspunkt kan også ligge på en ledning. En ledning kan ligge i en ledning. En samling ledninger og koplingspunkter inngår i et ledningsnett.

De forskjellige objekttypene har vanligvis forskjellige sett med egenskapsdata (attributter). Attributtene beskrives ved navn og datatype. Enkle datatyper (basis datatyper) kan være heltall, tekst eller dato. Eksempel: Ledningens navn kan være definert med basis datatype «tekst». Komplekse attributter består av flere attributter. Eksempel: Attributten «Adresse» kan bestå av attributtene Gatenavn, Gatenummer, Postnummer og Poststed, hvor hver enkelt er definert med egen datatype. Attributter kan også være definert med egne datatyper som omfatter et sett med lovlige verdier.

Siste trinn modelleringsprosessen er å definere ulike restriksjoner og multiplisitet som skal gjelde de ulike elementene i datamodellen. Restriksjoner og multiplisitet defineres både på assosiasjoner og attributter. Eksempel: En ledning kan bare være tilkoplet 0,1 eller 2 koplingspunkter i enden, mens et koplingspunkt kan stå i enden av 0,1 eller mange ledninger. En attributt kan defineres til alltid å ha verdi, eller kan forekomme med flere verdier. Det er en restriksjon når en attributt bare kan ha et gitt sett med lovlige verdier (verdidomene). Eksempel: Attributten «ledningstype» kan bare ha verdidomenet «Vannledning», «Overvannsledning» eller «Avløpsledning».

Hvorfor konseptuell datamodellering?

Hensikten er å gi en presis beskrivelse av de data som skal inngå i et informasjonssystem. Viktigst av alt i prosessen, er å få en korrekt avbildning av datastrukturene slik fagpersonene og brukerne mener informasjonen er. Dette er helt avgjørende for at implementasjonen av systemet skal bli som forventet. Ofte er brukere og fagfolk på det aktuelle fagområde lite datateknisk skolert, og vice versa har systemfolk lite kunnskap om fagområdet.

Følgelig er datamodellering en kommunikasjonsprosess hvor de ulike aktørene i arbeidet samles om en felles presis beskrivelse (datamodellen) som ivaretar en felles forståelse av datastrukturene som inngår i interesseområdet.

Det er mulig å beskrive datamodellene så presist at de automatisk kan tolkes av datamaskin og overføres til implementering i et datasystem. F.eks. finnes det verktøy for datamodellering som automatisk genererer SQL-tabelldefinisjoner (Structured Query Language)eller XSD- modell for XML (XML Schema Definition, Extensible Markup Language). Dette er selvsagt også en viktig side av datamodellering.

Metoder og verktøy for konseptuell datamodellering

Gjennom de siste 20 år er det lansert flere teknikker for å beskrive datamodeller, men de kan grovt deles i to grupper

– Grafisk representasjon. Eksempler: ERM (Entity-relationship model), NIAM (Natural language Information Analysis Modell) og UML(Unified Modeling Language)
– Tekstbasert representasjon. Eksempler: Express (datamodelleringsspråket i STEP, Standard for the Exchange of Product model data) og XSD (XML)

De tekstbaserte verktøyene har en syntaks som minner om dataprogrammer. De gir en presis beskrivelse av datastrukturen som lett lar seg videreføre til implementasjon, men ulempen er at beskrivelsen fort blir uoversiktlig. Det er stor risk for at personell som representerer fagområdet faller av lasset i prosessen. Det skaper usikkerhet om hvor god modellen til slutt er blitt. XML er i dag mye brukt for datautveksling, og ikke sjeldent modellert direkte som XSD-modell. Pga. kompleksiteten finnes eksempler på at løsningen ikke ble bra fordi datamodellen ble for uoversiktlig.

De grafiske verktøyene er gode til bruk i kommunikasjonsprosessen mellom fagfolk og systemfolk. Tidligere var ulempen at den grafiske presentasjonen ikke lot seg overføre til implementasjon. Dette er til en viss grad også tilfelle i dag, men flere verktøy for modellering av f.eks. UML, fungerer slik at det bygges opp en database i tillegg til den grafiske framstillingen. Fra databasen kan det automatisk lages implementasjonsavhengige datadefinisjoner, f.eks. definisjoner for SQL-tabeller og XML-filer. Det er også mulig å overføre UML-modeller for etablering av GML-strukturer [2]. Eksempler på UML-verktøy er

- Rational Rose
- MS Visio
- Argo UML
- Sparx Systems Enterprise Architect

UML

UML – Unified Modeling Language – er en implementasjonsuavhengig metode og teknikk for beskrivelse av datastrukturene i et informasjonssystem, ofte også kalt informasjonsmodell eller applikasjonsskjema. Etter at modellen er beskrevet i UML spesifiseres systemløsninger, samt modellering for en systemavhengig implementasjon.

Et applikasjonsskjema (informasjonsmodell) beskrevet i UML dekker to formål:

- Gi en korrekt menneskelig forståelse av objekter, egenskaper, relasjoner og eventuelt operasjoner innenfor sitt fagområde/ interesseområde.
- Være leselig av en datamaskin, for å kunne anvende automatiske rutiner i henhold til implementasjon, dataforvaltning og utveksling.

Det vil være for langt å gi fullstendig innføring i UML i denne artikkelen. Til dette anbefales generell UML-litteratur, som det finnes mye av. Figur 1, (som er hentet fra [7]) viser et eksempel med de mest sentrale konseptene og beskrivelsesteknikkene i UML.

Hovedelementene i UML samsvarer med hovedkonseptene for datamodellering:

- KLASSE, grafisk representert som 3-delt rektangel, samsvarer med *objekttype*, *komplekse attributter* og *verdidomener*
- ASSOSIASJON, grafisk representert som linje mellom KLASSER, samsvarer med *sammenhenger* mellom *objekttyper*
- ATTRIBUTTER, tekst i midterste del av KLASSE, samsvarer med *attributter*

UML-modellering av geografisk informasjon

I standardiseringsarbeidet som har endt opp med ISO 19100-serien av standarder for geografisk informasjon, ble UML valgt som modelleringsverktøy. Alle grunnleggende konsepter og datatyper som er vanlig i interesseområdet geografisk informasjon – geometri og topologi (spatial), tid (temporal og metadata – er definert i form av UML klassediagram.

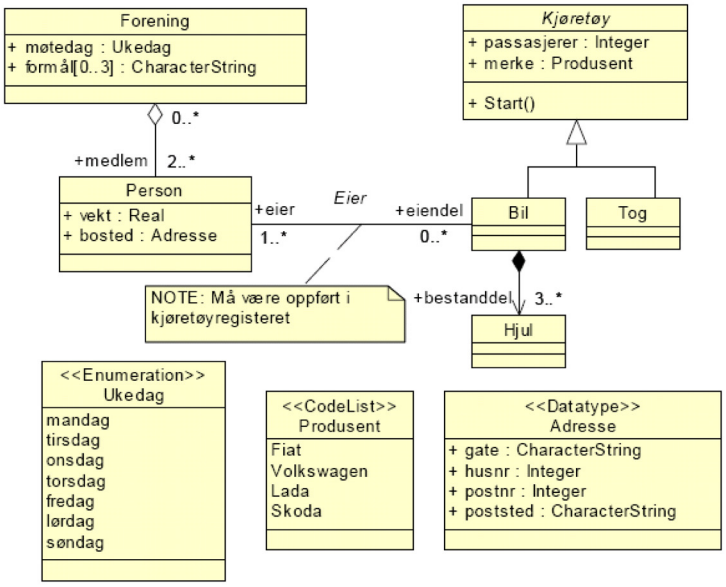
UML er et omfattende språk, og en kan beskrive det samme på flere måter. Imidlertid er det ønskelig å modellere mest mulig ensartet, først og fremst for å sikre at modellene lar seg implementere og sikre interoperabilitet.

ISO-standarden ISO 19109:2005, Geographic information – Rules for application schema, inneholder regler for hvordan et applikasjonsskjema i UML skal etableres, og strammer noe inn på frihetsgradene i UML. Den tar utgangspunkt i en generell objektmodell for geografiske objekter (General Feature Model) og bygger regler ut fra den. I tillegg er det laget regler for hvordan applikasjonsskjemaet skal benytte de grunnleggende konseptene i [3], [4] og [5].

Statens kartverk har gått et skritt lenger og utarbeidet mer detaljerte retningslinjer for bruk av UML i applikasjonsmodeller. Disse reglene er brukt ved remodellering av alle datamodeller i SOSI-standarden, uten at reglene rokker på den generelle framgangsmåte å modellere på, slik som beskrevet i det første kapitlet. Hovedreglene er også i samsvar med Rules for application schema [1] og Conceptual schema language [6]:

1. Lag oversikt over modellen med UML's pakkemekanismer
2. Avgrens virkeligheten, identifiser objekttyper som beskrives som UML-klasser

Eksempel på noen av de viktigste hovedelementene i et UML klassediagram.



Figur 2. Introduksjon til UML

Objekttyper:
Interesseområdet kan deles inn i objekttyper som har samme egenskaper, assosiasjoner og oppførsel. Forening, Person, Bil, Tog, Hjul.

Egenskaper:
En forening kan ha minst en, og inntil tre (av egenskapen) formål Foreningen har en fast møtedag, som er tatt ifra en lukket liste over ukedager.
En Person kan ha en angitt egenskapsverdi for sin vekt, av typen desimaltall.
Et kjøretøy kan ta et antall passasjerer, av typen heltall.
Et kjøretøy kan ha et fabrikat, med verdi tatt ifra en åpen liste.

Åpen aggregering:
En Forening består av medlemmer som er Personer.
Foreninger må ha minst to medlemmer. (2..*).
En Person kan være medlem av flere Foreninger.

Vanlig assosiasjon:
Personer Eier Biler. Assosiasjonsnavn er i verbs form. Null eller flere biler (0..*) kan ha rollen som en eiendel til en Person. En Bil må ha minst en Person (1..*) som eier. Roller skal være i substantivsform.

Komposisjon:
En Bil har (som eide komponenter) minimum tre Hjul. Pilen angir at instanser av Hjul ikke kan finne ut hvilken Bil de sitter på.

Subtyping:
Bil er en subtype av Kjøretøy. (Arver egenskaper, assosiasjoner og operasjoner).
Kjøretøy er supertype til Bil og Tog. Abstrakte supertyper instansieres ikke. (Navnet er i kursiv).

Oppførsel:
Et Kjøretøy kan utføre en operasjon Start().

Lukket liste – Enumeration:
Ukedag er en lukket liste hvor ingen nye verdier kan legges inn.

Åpen liste – CodeList:
Produsent er ei liste over kjente produsenter, åpen for nye produsenter.

Datatype:
Adresse er en egendefinert datatype som beskriver verdidomenet til bostedet for en Person.

Note:
Noten koblet til eierskapet mellom Person og Bil er en beskrankning på assosiasjonen.

Figur 1 Hovedelementene i UML klassediagram (fra [7])

3. Organiser objekttypene slik at generalisering/spesialisering framkommer
4. Egenskaper beskrives som attributter
5. Assosiasjoner mellom objekttyper beskrives
6. Kvalitet og metadata legges inn som attributter
7. Geometri-attributter refererer til klasser i [3] (eller en profil av denne)
8. Erstatt underforståtte «geometriske» sammenhenger med eksplisitt beskrivelse
9. Definer verdidomener (lovlige verdier for attributter)

Det henvises til [7] for ytterligere detaljer. Dokumentet er et godt beskrevet regelverk som det kan være vel verdt å støtte seg til ved modellering av applikasjonsskjemaer hvor geografisk informasjon inngår.

Eksempel på UML-modellering

Eksemplet er hentet fra fagområdet *Reiseliv og Turisme* som i de siste årene har hatt en sterk vekst i forvaltning og distribusjon av informasjon om reisemål og servicetilbud for turister. Internett er en sentral distribusjonskanal i denne sammenheng.

Eksemplet er hentet fra et Forskningsrådsprosjektet – MOVE [8]. Undersøkelser i prosjektet viser at oppbygging av databaser og strukturering av turistinformasjon er forskjellig hos de ulike aktører som forvalter slik informasjon. Forskjellene er ikke store når det gjelder hva som er kjerneinformasjon, men forskjellene øker med detaljeringsgraden og bredden av informasjon. Dette er naturlig, ettersom de ulike aktørene har forskjellige forretningsinteresser. Undersøkelsen viser også at aktørene i liten grad har tatt i bruk eksisterende standarder for sentrale dataelementer som koordinater og tid. Likevel synes det å være økende interesse for å utveksle turistinformasjon mellom aktører.

En forutsetning for effektiv og kvalitetsmessig god utveksling av turistinformasjon er å innføre en felles datamodell – som grunnlag for en standard for utveksling av data. MOVE-prosjektet utviklet en datamodell som grunnlag for en databasebeskrivelse, som også kan være aktuell som utvekslingsformat mellom databaser.

Som nevnt tidligere, er første trinn i modelleringsarbeidet å avgrense den del av verden og tilhørende informasjon som skal forvaltes i informasjonssystemet. I denne sammenheng er det valgt å begrense detaljeringsgraden, og sett med «reiselivsaktørers øyne» avgrenses verden til en objekttype – *Interessepunkt (Object of Interest)*.

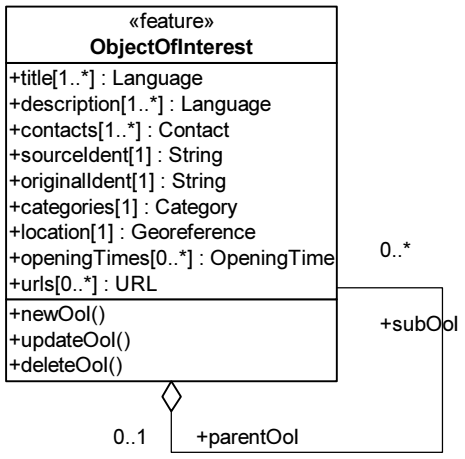
Denne objekttypen inneholder kjerneinformasjonen om et objekt som er av interesse for en turist (f.eks attraksjon eller service):

– Hva det er (beskrivelse)
– Hvor det ligger geografisk
– Når det er tilgjengelig
– Hvordan komme i kontakt
– Lenker til detaljinformasjon

Kjerneinformasjonen er et minimumssett av turistinformasjon, men vil likevel dekke alle typer stedfestet informasjon som etterspørres av turistene:

– Attraksjon og kjente steder
– Arrangementer, begivenheter og aktiviteter
– Overnatting og tjenester
– Trafikk og transport

Den konseptuelle modellen av Interessepunkt er vist som UML-modell på figur 2.

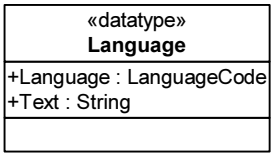


Figur 2 UML-modell for Interessepunkt

Objekttypen *Interessepunkt* er representert i en UML-klasse *ObjectOfInterest*.

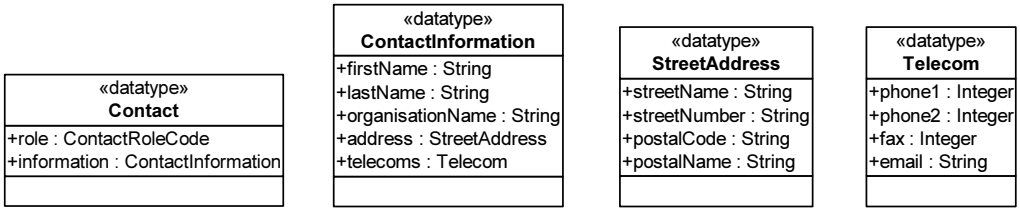
Modellen viser også at det eksisterer sammenhenger (assosiasjoner) mellom *Interessepunkter*, riktignok med den egenskap at et *Interessepunkt* kan «tilhøre» et annet *Interessepunkt*, og eventuelt arve egenskaper fra sin tilhørighet. Dette er vist ved assosiasjonen *parentOoI*. Eksempel: Et hotell med restaurant og golfbane er 3 ulike Interessepunkter, men hører sammen og kan ha noe felles informasjon, f.eks. tittel og adresse. Assosiasjonen viser også at et *Interessepunkt* bare kan tilhøre ingen eller ett annet *Interessepunkt* ved angitt multiplisitet (0..1), mens ett *Interessepunkt* kan ha mange tilhørende *Interessepunkt* (0..*).

Egenskapsdata (attributtene) for *Interessepunkt* som er angitt i UML-klassen, viser at flere er av typen komplekse attributter. F.eks. *Title* er definert med datatypen *Language* (se figur 3), som ivaretar muligheten for å uttrykke tittelen på flere språk. Følgelig må *Title* kunne forekomme med flere instanser som vist ved multiplisitet (1..*).



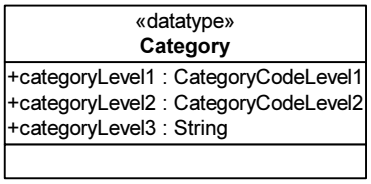
Figur 3 Datatype Language

Komplekse attributter kan ha flere nivåer av kompleksitet, eksempelvis attributten *Contacts*. Se figur 4.



Figur 4 Datatype Contact

Attributten Categories er viktig i modellen ettersom den representerer klassifikasjonen av de ulike *Interessepunktene*. Klassifikasjonen kan angis i 3 nivåer. Se figur 5.

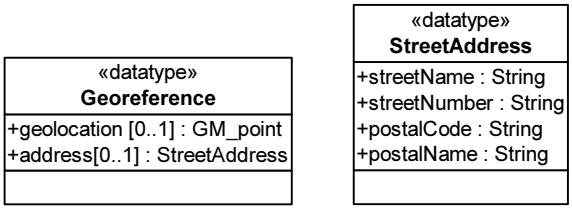


Figur 5 Datatype Category

Modellen forteller at et Interessepunkt beskrives kun av ett sett med kategorier. Noen produsenter av turistinformasjon knytter ofte flere kategorier til et interessepunkt. F.eks. kan et hotell både ha kategorien *overnatting/hotell* og *servering/restaurant*. Denne modellen krever at ved slike tilfeller, må et nytt interessepunkt etableres (gjerne med tilhørighet til hverandre). Dette er valgt fordi det vurderes som enklere og mer logisk (?) ettersom noe av informasjonen for øvrig kan være forskjellig, f.eks. kan hotellets åpningstider være forskjellig fra restaurantens åpningstider.

Imidlertid viser dette eksemplet at det ikke finnes en fasit på en datamodell. Modellene vil være forskjellige alt etter hvilke øyne som ser og hvilke oppgaver som skal løses.

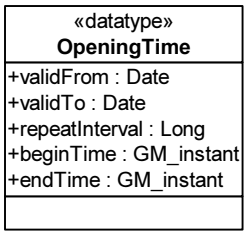
Modellen forholder seg også til andre standarder, f.eks. ISO 19107:2003, Geographic information – Spatial schema. *Interessepunktets* geografiske posisjon angis enten med koordinater eller offisiell adresse. Se figur 6.



Figur 6 Datatype Georeference

Koordinater angis i henhold ISO-standard, ref [2] og [3], som *GM_Point* som datatype.

ISO 19108:2003, Geographic information – Temporal schema følges i datamodellen for spesifikasjon av åpningstider. *Interessepunktets* åpningstider kan angis med gyldighetstidsrom og med mulighet for flere ulike spesifikke åpningstider innenfor tidsrommet for gyldighet. F.eks. kan gyldigheten være mai-august, med ulike åpningstider for mandag-fredag, lørdag og søndag.



Figur 7 Datatype OpeningTime

Gyldighetstidsrommet angis med *validFrom* og *validTo*, og *repeatInterval* angir perioden mellom hver åpningstid (f.eks. daglig, ukentlig,..), og *beginTime* og *endTime* spesifiserer åpningstiden med datatype hentet fra ISO 19108:2003. Se figur 7.

Oppsummering

UML er velegnet som modelleringsverktøy av applikasjonsskjema for geografiske data av flere grunner:

- UML fungerer godt for modellbeskrivelse i kommunikasjonen mellom fagpersonell og systempersonell
- Konsepter og modeller fra ISO 19100-serien foreligger i UML og kan integreres i applikasjonsmodellen
- Datamodeller i UML kan automatisk overføres til implementasjon av databaser
- Datamodeller i UML kan automatisk overføres til implementasjon av XML (ikke nødvendig å modellere direkte i XSD)
- Datamodeller i UML kan automatisk overføres til implementasjon av GML

Regelverket som er etablert i [1], [6] og [7] sikrer at modellene kan implementeres og at modeller fra ulike fagområder kan realiseres på samme plattform. Regelverket sporer modelleringsarbeidet inn på en enkel og grunnleggende bruk av UML, men ikke i en slik grad at det skaper store hindringer i å utrykke modellen.

Referanser

[1]. ISO 19109:2005,Geographic Information–Rules for Application Schema
[2]. ISO 19139:2005,Geographic Information–Geography Markup Language (GML)
[3]. ISO 19107:2003,Geographic Information–Spatial Schema
[4]. ISO 19108:2003,Geographic Information–Temporal Schema
[5]. ISO 19115:2003,Geographic Information–Metadata Schema
[6]. ISO 19103, Geographic information — Conceptual schema language
[7]. Retningslinjer for modellering i UML. Statens kartverk – SOSI-sekretariatet
[8]. MOVE. NFR-prosjekt 2004-2006. Telenor R&D. www.moveweb.no