

Digitaliseringsdirektoratet
Norwegian Digitalisation Agency

Veileder for beskrivelse av informasjonsmodeller

Innholdsfortegnelse

Innledning	2
Hensikt og avgrensning	2
Målgruppe	2
Struktur	2
Navnerom som er brukt i veilederen	3
Kort om ModellDCAT-AP-NO	4
Generelt	5
Om bruk av egenskap «identifikator» (dct:identifier)	5
Om bruk av egenskap «utgiver» (dct:publisher) og «produsent» (dct:creator)	5
Om bruk av egenskap «dekningsområde» (dct:spatial)	6
Om bruk av egenskap «lisens» (dct:license)	6
Informasjonsmodell	7
Hva er en informasjonsmodell?	7
Hvordan beskriver du en informasjonsmodell?	7
Eksempel på en informasjonsmodell (i RDF Turtle)	8
Hva gjør du når du skal gjøre din informasjonsmodell tilgjengelig?	8
Modellinnhold	9
Modellkatalog	10
Hva bruker du en modellkatalog til?	10
Hvordan beskriver du en modellkatalog?	10
Eksempel på en modellkatalog (i RDF Turtle)	10
Hva gjør du når du skal publisere modellkatalogen din i Felles datakatalog?	10
Informasjonsmodell og relasjoner til datasett, datatjenester og begreper	11
Referanser	13
Hjelpemidler	14
Akronymer	15
Informasjonsmodellering enkelt forklart	16
Modellering – et enkelt utgangspunkt	16
Relasjoner	16
Metode	17



Innmelding av feil og mangler:

Dersom du finner feil eller mangler i dokumentet, ber vi om at dette meldes inn på [Github Issues](#). Dersom du ikke allerede har bruker på Github kan du opprette bruker gratis.

Status: under utarbeidelse

Versjon: forberedelse til 1.0

Publisert: under utarbeidelse

Oppdatert: 2021-06-24

Gjeldende versjon: <https://data.norge.no/guide/veileder-modelldcat-ap-no/>

Redaktørens utkast: <https://informasjonsforvaltning.github.io/veileder-modelldcat-ap-no/>

Innledning

Hensikt og avgrensning

Denne veilederen skal gi en hjelp til virksomhetene i offentlig forvaltning til å beskrive informasjonsmodellene sine i henhold til [Spesifikasjonen for beskrivelse av informasjonsmodeller \(ModellDCAT-AP-NO\)](#) og tilgjengeliggjøre dem i [Felles datakatalog](#). Den gjelder altså først og fremst virksomheter som allerede har informasjonsmodeller som skal høstes, og ikke de som skal starte opp informasjonsmodellering.

Ulike virksomheter bruker ofte ulike modelleringsspråk og -verktøyer. Selv om noen etater forholder seg til de samme modelleringsspråkene og -verktøyene, kan bruken av disse og hvilke versjoner man anvender variere. Hensikten med ModellDCAT-AP-NO er å kunne beskrive virksomhetenes ulike informasjonsmodeller på en ensartet måte. Gjennom en lik tilnærming, er det enklere å finne, forstå, sammenligne og gjenbruke informasjonsmodeller i offentlig sektor.

Det er ikke et mål for denne veilederen å beskrive alle felter i ModellDCAT-AP-NO, heller ikke teknisk og normativt. For normative beskrivelser av alle feltene i spesifikasjonen for informasjonsmodeller, se [ModellDCAT-AP-NO](#) og [valideringsreglene \(shacl\)](#).

Målgruppe

Målgruppen for denne veilederen er deg som:

- skal bruke ModellDCAT-AP-NO for å beskrive eksisterende informasjonsmodeller i virksomheten din
- skal utvikle/tilpasse verktøystøtte i virksomheten din for beskrivelse av informasjonsmodeller og/eller publisering av disse i henhold til ModellDCAT-AP-NO.
- ønsker å få bedre forståelse av ModellDCAT-AP-NO
- ønsker å tilgjengeliggjøre informasjonsmodeller i virksomheten din i Felles datakatalog.
- ønsker bedre å finne, forstå, sammenligne og gjenbruke modeller beskrevet i henhold til ModellDCAT-AP-NO
- ønsker å bruke erfaring med og kunnskap om ModellDCAT-AP-NO inn i diskusjoner om relevante, internasjonale standarder

For å få best mulig innsikt i ModellDCAT-AP-NO og utbytte av veilederen, er det en fordel om du har litt kjennskap til Resource Description Framework (RDF).

Struktur

Veilederen består av følgende deler:

- **Kort om ModellDCAT-AP-NO** – gir en kort innføring i spesifikasjonen for beskrivelse av informasjonsmodeller.
- **Generelt** – generell del som beskriver egenskaper og vokabularer som brukes på tvers av

klasser.

- **Informasjonsmodell** – omhandler hvordan man beskriver informasjonsmodeller.
- **Modellinnhold** – tar for seg hvordan man beskriver modellelementer og egenskaper.
- **Modellkatalog** – omhandler hvordan man beskriver en modellkatalog
- **Informasjonsmodell og relasjoner til begreper, datasett og datatjenester** – beskriver hvordan man knytter informasjonsmodeller sammen med øvrige katalogressurser.

I tillegg er det laget et vedlegg, **Modellering – et enkelt utgangspunkt** som gir en kort innføring i hva informasjonsmodeller er og anbefalt metode for utforming av disse.

Navnerom som er brukt i veilederen

Prefiks	Navnerom	Forklaring/navn
adms	http://www.w3.org/ns/adms#	Asset Description Metadata Schema
dcat	http://www.w3.org/ns/dcat#	Data Catalog Vocabulary
dct	http://purl.org/dc/terms/	DCMI Metadata Terms
foaf	http://xmlns.com/foaf/0.1/	FOAF Vocabulary
modelldcatno	https://data.norge.no/vocabulary/modelldcatno#	Spesifikasjon for beskrivelse av informasjonsmodeller (ModellDCAT-AP-NO)
owl	http://www.w3.org/2002/07/owl#	OWL Web Ontology Language
xsd	http://www.w3.org/2001/XMLSchema#	XML Schema Part 2: Datatypes Second Edition

Kort om ModellDCAT-AP-NO

ModellDCAT-AP-NO er basert på [Standard for beskrivelse av datasett, datatjenester og datakataloger \(DCAT-AP-NO\)](#) og inngår i [Rammeverk for informasjonsforvaltning](#).

Figuren nedenfor viser en forenklet modell av ModellDCAT-AP-NO beskrevet i UML. Egenskapene er utelatt, og kun de mest sentrale klassene og relasjonene er tatt med. Modellene viser kun de norske klasse- og relasjonsnavnene. For fullstendig oversikt, se spesifikasjonen for ModellDCAT-AP-NO.

<@@@ sett inn figuren her>.

Spesifikasjonen består av to hoveddeler, en katalogdel som er basert på DCAT-AP-NO 2.0 (lysebrune klasser) og en utvidet del, modelldel (grå klasser), som beskriver en informasjonsmodell og dens innhold. Som datasett og datatjenester, er informasjonsmodell beskrevet som en subklasse til klassen [Katalogisert ressurs](#) (`dcat:Resource`).

I ModellDCAT-AP-NO er det definert noen overordnede klasser. Disse er:

- [Katalogisert ressurs](#) (`dcat:Resource`)
- [Modellelement](#) (`modelldcatno:ModelElement`)
- [Egenskap](#) (`modelldcatno:Property`)

Disse skal i en konkret bruk erstattes med en av de spesifikke subclassene.

Siden ModellDCAT-AP-NO er basert på DCAT-AP-NO 2.0, er spesifikasjonen beskrevet i RDF. Det betyr at modellen (målmodellen) som du skal transformere til, også må være i RDF.

Generelt

Om bruk av egenskap «identifikator» (`dct:identifier`)

Egenskapen `dct:identifier` brukes til å oppgi identifikatoren til subjektet (første leddet) i en RDF-trippel. Subjektet i en RDF-trippel er per definisjon en identifikator (URI). I en konkret realisering vil instanser av klassene (inkl. deres subklasser) Modellkatalog (`dcat:Catalog`), Informasjonsmodell (`modelldcatno:InformationModel`), Modellelement (`modelldcatno:ModelElement`), Egenskap (`modelldcatno:Property`), Kodeliste (`modelldcatno:CodeList`) og Kodeelement (`modelldcatno:CodeElement`) derfor få en «innebygd» identifikator. Det er med andre ord strengt tatt ikke nødvendig å ha en identifikator (`dct:identifier`) i tillegg til den «innebygde» identifikatoren.

Egenskapen `dct:identifier` er derfor satt som anbefalt og ikke obligatorisk i de nevnte klasser og deres subklasser, med unntak av for Modellkatalog (`dcat:Catalog`). `dct:identifier` er satt som obligatorisk for Modellkatalog (`dcat:Catalog`) for at ModellDCAT-AP-NO skal være kompatibel med DCAT-AP-NO, ettersom `dct:identifier` er obligatorisk for `dcat:Catalog` der.

Egenskapen `dct:identifier` trenger ikke å inneholde den samme identifikatoren som den innebygde URIen i en RDF-trippel, men når det er den samme identifikatoren, anbefales det at hele den innebygde URIen (subjektet i en trippel) kopieres til `dct:identifier` i en Modellkatalog (`dcat:Catalog`).

Eksempel i RDF Turtle:

```
@prefix dct: <http://purl.org/dc/terms/> .
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://examples.com/infomoc/exmodcat1> a dcat:Catalog ;
    dct:identifier "https://examples.com/infomod/exmodcat1"^^xsd:anyURI .
```

Identifikatoren bør utformes i henhold til [Standard for pekere til offentlige ressurser på nett](#).

Om bruk av egenskap «utgiver» (`dct:publisher`) og «produsent» (`dct:creator`)

Det anbefales å bruke følgende mønster for URI til utgiver (`dct:publisher`) og produsent (`dct:creator`), der det siste leddet er organisasjonsnummeret:

```
<enInfoModell> dct:publisher <https://organization-
catalogue.fellesdatakatalog.digdir.no/organizations/974760673> .
```

Når det er behov for å oppgi at «eieren» for en informasjonsmodell som ikke er den samme som utgiveren (`dct:publisher`), anbefales det å bruke produsent (`dct:creator`).

Om bruk av egenskap «dekningsområde» (**dct:spatial**)

Til dekningsområde (**dct:spatial**), anbefales det å bruke EU sine kontrollerte vokabularer [Continent](#), [Country](#), og [Place](#). Eksemplet under (i RDF Turtle) viser bruk av Country-vokabularet for å oppgi Norge (NOR):

```
<enInfoModell> dct:spatial  
<http://publications.europa.eu/resource/authority/country/NOR> . # Norge
```

Det anbefales å bruke Kartverket sine kontrollerte vokabularer for fylke og kommune. Eksemplene under viser Oslo som fylke hhv. kommune.

```
<mod1> dct:spatial <https://data.geonorge.no/administrativeEnheter/fylke/id/173159> .  
# Oslo som fylke  
  
<mod2> dct:spatial <https://data.geonorge.no/administrativeEnheter/kommune/id/173018>  
. # Oslo som kommune
```

Du kan finne fram til URIene [her](#).

Om bruk av egenskap «lisens» (**dct:license**)

For lisens (**dct:license**) anbefales det å bruke EU sitt kontrollerte vokabular for lisens, [Licence](#), hvis lisensen du bruker finnes på listen. I eksemplet under er det [CC BY v. 4.0](#) som er brukt.

```
<enInfoModell> dct:license  
<http://publications.europa.eu/resource/authority/licence/CC_BY_4_0> .
```


Informasjonsmodell

Hva er en informasjonsmodell?

Med informasjonsmodell (`modelldcatno:InformationModel`) mener vi en formell beskrivelse av informasjonen en virksomhet trenger å motta eller selv produsere for å utføre sitt daglige virke. Vi legger til grunn en vid tolkning av begrepet informasjonsmodell. Den omfatter modeller på ulike abstraksjonsnivåer (som konseptuelle, logiske og fysiske modeller) og modeller som beskriver ulike aspekter ved informasjon (som felles- og anvendelsesmodeller).

For ModelldCAT-AP-NO er det utarbeidet et [vokabular for klassifisering av modeller](#). Dette er:

- **Konseptuell modell** er en form for kvalitativ modell som beskriver de viktigste konseptene innenfor et fagdomene og sammenhengen mellom disse.
- **Logisk modell** beskriver hvilke typer informasjon som inngår i en avgrenset sammenheng og hvordan de er logisk relatert, uavhengig av teknologi.
- **Fysisk modell** er en logisk modell som er utarbeidet for å beskrive datautveksling eller lagring av data for en bestemt løsning.
- **Fellesmodell** er en informasjonsmodell til felles bruk på tvers av virksomheter, forretningsområder og/eller applikasjonssegmenter.
- **Anvendelsesmodell** er en modell som er rettet mot et spesifikt anvendelsesområde i en avgrenset kontekst, og er sammensatt av elementer i en fellesmodell.

Hvordan beskriver du en informasjonsmodell?

Kun tittel (`dct:title`) og utgiver (`dct:publisher`, se [Om bruk av egenskap «utgiver»](#) (`dct:publisher`) og «produsent» (`dct:creator`)) er obligatorisk for en informasjonsmodell. Det er med andre ord ikke påkrevd å ta med modellelementer i modellbeskrivelsen din, selv om dette er anbefalt. Hvis modellen din kun er tilgjengelig som f.eks. en bildefil, anbefales det å gjøre den tilgjengelig på en nettside og peke til denne ved bruk av `dct:hasFormat`. Det er også mulig å referere til en hjemmeside hvor modellen er nærmere beskrevet ved bruk av egenskapen `foaf:homepage`.

Hva som betraktes som større eller mindre endringer, er en faglig vurdering som denne veilederen ikke dekker. Når du ut fra den faglige vurderingen mener at endringene i informasjonsmodellen din er så store at det kan ha betydning for hvordan informasjonsmodellen skal forstås/brukes, bør du opprette en ny instans av `modelldcatno:InformationModel` for den nye versjonen av informasjonsmodellen din, slik at den nye versjonen også får en egen identifikator. Dette gjør at både den nye og den gamle versjonen av informasjonsmodellen din kan refereres til. Du bør også bruke egenskapen «Informasjonsmodell: erstatter» (`dct:replaces`) fra den nye versjonen til den gamle versjonen av informasjonsmodellen (ev. den motsatte egenskapen, «Informasjonsmodell: erstattet av» (`dct:isReplacedBy`) fra den gamle til den nye versjonen av informasjonsmodellen). For mindre endringer kan du bruke egenskapene «Informasjonsmodell: versjon» (`owl:versionInfo`) og «Informasjonsmodell: versjonsnote» (`adms:versionNotes`) til å dokumentere endringene, uten at det blir opprettet en ny instans og dermed en ny identifikator.

Eksempel på en informasjonsmodell (i RDF Turtle)

Se [eksempel 1 i RDF Turtle](#), som er en informasjonsmodell uten modellelementer, men med henvisning til hvor modellen finnes. Eksemplet viser også bruk av alle egenskapene (dvs. ikke bare obligatoriske).

Se [eksempel 2 i RDF Turtle](#), som er en informasjonsmodell med et modellelement. Eksemplet viser bruk av kun obligatoriske egenskaper.

Hva gjør du når du skal gjøre din informasjonsmodell tilgjengelig?

To tilfeller:

- Uten modellelementer, men med lenke til hjemmeside der selve modellen finnes, se eksempel 1 ovenfor.
- Med modellelementer, slik at hele modellen blir tilgjengeliggjort for automatisk høsting til modellkatalogen, se eksempel 2 ovenfor.

Modellinnhold

<tekst kommer>

Modellkatalog

Hva bruker du en modellkatalog til?

Informasjonsmodeller fra en eller flere virksomheter samles i en modellkatalog. En modellkatalog kan ses som et omslag til eller innbinding av modellene. En modellkatalog skal ha én utgiver, og en og samme virksomhet kan være utgiver av flere modellkataloger.

Hvordan beskriver du en modellkatalog?

Som minimum skal følgende egenskaper ha verdier:

- beskrivelse (`dct:description`): En kort og presis beskrivelse av modellkatalogen skal gjøre det lett for andre å se hva det inneholder. Beskrivelse er et obligatorisk felt. Bør gjentas når beskrivelsen finnes i flere ulike språk/målformer.
- identifikator (`dct:identifier`): Obligatorisk (se [Informasjonsmodell](#)).
- tittel (`dct:title`): Kortfattet om katalogen. Angi, uten å liste, hvilke modeller den omfatter, f.eks. «Informasjonsmodellene til Digitaliseringsdirektoratet». Bør gjentas når tittelen finnes i flere ulike språk/målformer.
- modell (`modelldcatno:model`): Referanser til en eller flere informasjonsmodeller som er inkludert i modellkatalogen.
 - Kommentar: I [UML-diagrammet](#) står det at `dct:hasPart` er obligatorisk mens `modelldcatno:model` er anbefalt. `dcat:Resource` som `dct:hasPart` peker på er en abstrakt klasse og skal erstattes med instanser av subklassen `modelldcatno:InformationModel` eller `dcat:Catalog`. Bortsett fra tilfeller der man beskriver en modellkatalog bestående av andre modellkataloger, skal en modellkatalog ha minst én informasjonsmodell.
- utgiver (`dct:publisher`): Se [Om bruk av egenskap «utgiver»](#) (`dct:publisher`) og «produsent» (`dct:creator`).

Eksempel på en modellkatalog (i RDF Turtle)

Se [eksemplet i RDF Turtle](#).

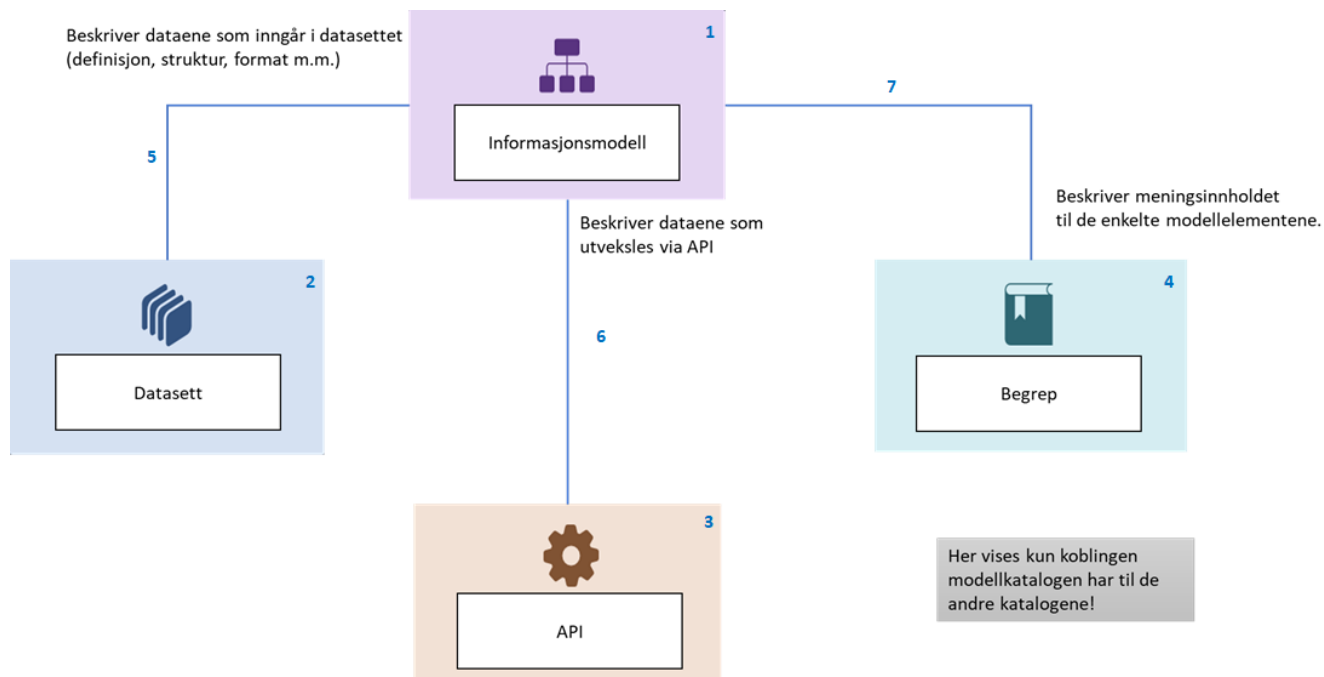
Hva gjør du når du skal publisere modellkatalogen din i Felles datakatalog?

Når du skal publisere katalogen din i Felles datakatalog, må modellene mappes iht. ModelDCAT-AP-NO til et format som kan høstes. Pr. i dag støtter Felles datakatalog følgende RDF-serialiseringer: RDF/XML, Turtle og JSON-LD.

Modellkatalogen må da legges ut som en nettressurs, f.eks. i Github. URIen til nettressursen registreres i Felles datakatalog, og katalogen høstes så automatisk.

Informasjonsmodell og relasjoner til datasett, datatjenester og begreper

Figuren nedenfor viser sammenhengen mellom informasjonsmodell, datasett, API (datatjeneste) og begreper:



Figur 1. Sammenheng mellom informasjonsmodell, datasett, datatjenester og begreper

Nummerering	ModellDCAT-AP-NO
1	Informasjonsmodell (dct:InformationModel)
2	Datasett (dcat:Dataset)
3	Datatjeneste (dcat:DataService)
4	Begrep (skos:Concept)
5	i samsvar med (dct:conformsTo)
6	i samsvar med (dct:conformsTo)
7	begrep (dct:subject)

Som vi har sett, kan vi referere fra informasjonsmodeller, modellelementer, egenskaper eller kodeelementer til begreper ved bruk av egenskapen begrep (**dct:subject**). Se beskrivelser og eksempler under [Informasjonsmodell](#) og [Begrepsreferanser](#).

Når det gjelder et datasett eller datatjeneste, refererer man fra disse til en informasjonsmodell. I ModellDCAT-AP-NO er klassen Informasjonsmodell subklasse av **dct:Standard**. Ved å bruke egenskapen **dct:conformsTo**, kan et datasett eller en datatjeneste referere til en informasjonsmodell.

Eksempler i RDF Turtle:

Datasett til informasjonsmodell:

```
<https://examples.com/infomoc/exdataset> a dcat:Dataset ;  
    dct:conformsTo <https://github.com/Informasjonsforvaltning/modelldcat-ap-  
no/examples/testMod1> .
```

Datatjeneste til informasjonsmodell:

```
<https://examples.com/infomoc/exdataservice> a dcat:Dataservice ;  
    dct:conformsTo <https://github.com/Informasjonsforvaltning/modelldcat-ap-  
no/examples/testMod2> .
```

Referanser

DCAT-AP-NO: [Standard for beskrivelse av datasett, datatjenester og datakataloger \(DCAT-AP-NO\)](#)

Github-repository for ModellDCAT-AP-NO, <https://github.com/Informasjonsforvaltning/modelldcat-ap-no>

ModellDCAT-AP-NO: [Spesifikasjon for beskrivelse av informasjonsmodeller \(ModellDCAT-AP-NO\)](#)

[RDF Turtle](#)

[Resource Description Framework \(RDF\)](#)

SKOS-AP-NO: [Forvaltningsstandard for tilgjengeliggjøring av begrepsbeskrivelser basert på SKOS](#)

[Standard for pekere til offentlige ressurser på nett](#)

Hjelpemidler

Bibliotek for å mappe en modelldcatno-modell til RDF, [Modelldcatnotordf](#). Kildekode: <https://github.com/Informasjonsforvaltning/modelldcatnotordf>.

Valideringsregler (shacl), https://github.com/Informasjonsforvaltning/modelldcat-ap-no/blob/develop/shacl/modelldcat-ap-no_v1_SHACL-shapes.ttl

Validator: <https://data.norge.no/validator>

IDLab Turtle Validator, <http://ttl.summerofcode.be>

Akronymer

AP – Application Profile

DCAT – Data Catalog

FDK – Felles datakatalog

JSON – JavaScript Object Notation

JSON – LD – JSON for Linked Data

NO – Norwegian

RDF – Resource Description Framework

SKOS – Simple Knowledge Organisation System

URI – Uniform Resource identifier

XKOS – eXtended Knowledge Organisation System

XML – eXtended Markup Language

Informasjonsmodellering enkelt forklart

Modellering – et enkelt utgangspunkt

Vi lager modeller for å strukturere kunnskap. Vi deler opp informasjonen i tenkte enheter: ting. Disse tingene omtales litt forskjellig avhengig av tradisjon og sammenheng, noen sier klasser, noen sier konsepter og i ModellDCAT-AP-NO snakker vi om modellelementer. Uavhengig hva vi omtaler det som er klassen eller modellelementet en måte å samle informasjon om et konsept, noe konkret eller abstrakt.



Figur 2. Et konsept

Denne informasjonen som beskriver «tingen» kan vi kalle egenskaper.

Alt vi kan uttrykke for å beskrive «tingen» er egenskaper, enten det er enkle fakta som en farge, *Enkeltype*, eller en struktur av opplysninger slik som en adresse, *Datatype*, som har flere felter, slik som gatenavn, husnummer, postnummer og poststed. Allerede med disse tre byggesteinene, klasse, enkelttype og datatype, kan man beskrive en datastruktur.



Figur 3. En datatype

Relasjoner

En modell består som regel av mer enn ett konsept. Relasjonen mellom konsepter er i denne sammenhengen også en egenskap.

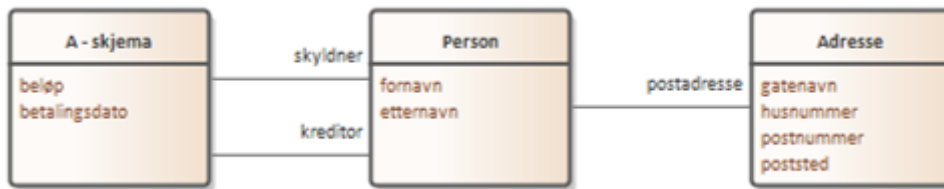
Hvilken type egenskap er litt avhengig av hva vi fokuserer på i modellen.

I en konseptuell modell der man f.eks. setter begreper i en sammenheng vil en kanskje kalle dette forholdet for en *Assosiasjon*.

<@@@@ sett inn figuren her, "Skyldner" → "Kreditor">

Her kan man si at en skyldner kjennetegnes av at man som en av egenskapene har en betalingsforpliktelse til en kreditor.

I modeller som beskriver konkrete forhold, logiske eller fysiske modeller, er det vanlig å beskrive relasjonen mellom to objekter utfra hva det ene objektet betyr for det andre. Rettere sagt, hvilken rolle spiller B for A.



Vi kan tenke oss et skjema (A) der man i tillegg til enkle egenskaper som beløp og datoer ønsker å beskrive en person med navn og adresse. Person er her et selvstendig konsept eller en klasse som blir brukt av skjemaet, i tillegg er det heftet på adresseinformasjon som en egen klasse.

Skjemaet som skal beskrives, inneholder informasjon om en som skylder penger, derfor kan man tenke seg at dette beskrives som skjema (A), med to egenskaper som relaterer seg til en person, i det ene tilfelle skyldner, i det andre tilfelle kreditor. Dette er det samme forholdet som var en assosiasjon i den konseptuelle modellen, men som uttrykkes som et rolle-forhold i den logiske eller fysiske modellen.

Ved hjelp av konseptet Modellelement, egenskapene rolle, der ulike Modellelementer brukes for å beskrive ulike deler av informasjonen; og enkelttype og datatype for data om konseptet, kan man uttrykke enhver tre-basert datastruktur.

ModellDCAT-AP-NO har et større mulighetsrom, noe du kan se tidligere i denne veilederen, men som en enkel start er disse fire elementene nok til å beskrive datastrukturer.

Metode

Ulike arkitekter vil ha ulike svar på hva de gjør når de modellerer, men her er en metode som virker:

Finn interesseområdet og omfanget av modellen

Det er sagt at alt henger sammen med alt, men skal man lage en modell er det viktig å klargjøre akkurat hva modellen skal gi en oversikt og innsikt i, og hvem som skal bruke modellen.

Finn mulige gjenbrukbare modeller

Ikke gjør dobbeltarbeid, hvis du kan bruke ting du eller andre allerede har definert betyr det ikke bare mindre arbeid for deg som arkitekt, det betyr også mindre arbeid for de som skal omsette den konkrete modellen til en applikasjon. Gjenbruk av de samme basiskonseptene letter den gjennomgående forståelsen og sikrer at man beskriver de samme tingene på en ensartet måte.

Finn de viktige objektstypene

Også kalt klassene, tingene eller konseptene. Pek ut hva som er viktigst i modellen, og start med disse og bygg ut derifra.

De viktige konseptene kjennetegnes ofte ved at ord som betegner konseptet går igjen i beskrivelsen av systemet. Les igjennom sentrale dokumenter som beskriver feltet og vurder viktigheten av alle substantiver.

I en konseptuell modell vil ofte de objektene som har flest tilknyttede relasjoner være de viktigste. La en gruppe mennesker som kjenner område skrive ned på gule lapper hva de forbinder med området som skal beskrives med ett ord. De sentrale objektene er ofte de som beskrives av flest lapper.

Finne sammenhengen mellom generelle og spesielle objekttyper

Noen konsepter er slik at det er en utdyping av et annet: loff er en type brød, da kan man si at brød er generelt og at loff er en spesialisering av brød. Med andre ord, vurder om en objekttype er en spesialisering eller en generalisering av en annen objekttype.

Definer egenskapene til objekttypene og beskriv tillatte verdier

Det er her informasjonen kommer inn i modellen. For hver objektstype, finn de egenskapene som beskriver objektet nærmere og legg dem til i form av egenskaper. Identifiser om egenskapen uttrykkes på en spesiell måte, er det et heltall, en streng, og har det noen spesifiserte minimum og maksimum verdier. F eks at en gitt egenskap er et heltall mellom 10 og 50.

Beskriv forhold mellom objekttypene

Her beskrives forholdene mellom objektene i modellen. Avhengig av hvilken type modell det er kan man beskrive dette som en assosiasjon, en rolle eller noen av de andre relasjonstypene som er beskrevet senere i dette dokumentet. Forholdet mellom objekttypene i modellen gir struktur.

Beskriv verdidomener i form av kodelister

Dersom man vil beskrive en egenskap som noe mer spesifikt enn at egenskapen har et gitt format, men ønsker at feltet skal inneholde en konkret verdi fra en liste, da definerer man en kodeliste. Slike kodelister kan beskrive valgmuligheter som har en lokal gyldighet i en gitt modell, eller være større lister som brukes på tvers av ulike modeller. Dersom listene er store og har et globalt anvendelsesområde, bør de publiseres som egne dokumenter og kun bli referert til fra modellen. Små lokale kodelister kan defineres som en del av modellen.