# Kapitel 7: Det visuelle udtryk

Af hensyn til brugbarheden bør man gøre modellerne så læsevenlige som kompleksiteten i modellen tillader det, og helt overordnet anbefales det at det visuelle udtryk i UML-modellerne ensrettes så meget som muligt.

Det visuelle udtryk af modellen understøtter kommunikation mellem forskellige interessenter såsom arkitekter, udviklere og forretningseksperter, samt på tværs af organisationer med forskellig dokumentationspraksis. Det visuelle udtryk kan dermed betragtes som UMLs største fordel, og denne udnyttes bedst, hvis diagrammerne er nemme at afkode og behagelige at kigge på. Derfor giver det mening at følge fælles principper for udarbejdelsen af diagrammer, for at opnå et genkendeligt visuelt udtryk. Dette gælder uanset om der er tale om begrebsdiagrammer, diagrammer for informationsmodeller eller logiske datamodeller.

De generelle anbefalinger nedenfor betyder ikke at selve indholdet af UML-modellen bliver bedre, men at det bliver nemmere at kommunikere modellen til interessenter.

## Anbefalinger til grafisk opsætning og præsentation

### Undgå krydsende streger hvor det er muligt

Gør dit UML-diagram lettere at læse og forstå ved at undgå at stregerne krydser hinanden. Ligeledes bør streger ikke krydse klasser, enumerationer eller strukturerede datatyper.

Hvis dit UML-diagram er så omfattende at det kan være svært at undgå dette, så kan dette være et tegn på at modellen med fordel kan opdeles i deldiagrammer.

### Foretræk vinkelrette knæk frem for skrå streger

Det kan give et mere ordnet og roligt udtryk hvis modellens streger ikke tegnes skråt fra element til element, men at de derimod knækkes vinkelret.

### Ensret klassernes størrelser og justeringer

Ensret UML-klassernes størrelser og justeringer i det omfang det er muligt.

I begrebsdiagrammer: lav de klasser der repræsenterer begreberne lige høje og hvor muligt lige brede

I informationsmodeller og logiske datamodeller: lad gerne klasser være lige brede, eller brug evt. 2 faste bredder.

Man kan evt. vælge at lade diagrammets mest centrale klasse/begreb være større end resten, dels for at give en visuel indikation af at det er centralt, dels for at give bedre plads til mange associationer. At lade et mindre centralt element være større vil derimod virker forvirrende.

Det giver et mere roligt udtryk hvis klassers kanter, hvor muligt er justeret så kanterne flugter med hinanden.

### Sørg for at al tekst er læsbar

Navne på associationer og associationsender skal være fri af andre elementer så hele navnet er synligt og altså ikke ligge oven på hinanden, .

Tag også hensyn til at indholdet af klasser (klassenavn, attributter og datatyper) bedst kan læses hvis der er plads mellem tekst og kanten af klassen. Dette kan gøre det nødvendigt at bryde ovenstående princip om ensretning af af klassers størrelse.

Desuden kan det være en god ide at øge tekststørrelse eller gøre skrift fed for at øge læsbarheden.

### 

### Brug anbefalede farver

Farver skal anvendes til fremhæve information der allerede er tilstede i modellen. Når der anvendes farver bør man bruge dem, der anbefales i modelreglerne og dem der typisk anvendes til specifikke modelelementtyper.

Modelreglerne fastlægger for eksempel at genbrugte/indlånte elementer fremhæves i kernemodeller med en visuel markering. Her anbefales farven blå. Informationen om at et givet modelelement er genbrugt og defineret i en anden model vil allerede være til stede i form af elementets URI og evt. kvalificerede elementnavne, men fremhæves yderligere med den visuelle markering.

Grunddatamodellering har en praksis vedr. enumerationer og datatyper som også anbefales anvendt bredt, selvom farveanbefalingerne her ikke indgår i Modelreglerne.

Der anbefales følgende farver til modelelementer:

* Sandfarve/beige til egendefinerede klasser
* Blå til genbrugt/fremmede/indlånte klasser/begreber fra andre kernemodeller
* Grøn til klassifikationer i form af enumerationer og klassifikationsklasser
* Gul til strukturerede datatyper
* Lysegrå til objekter

De valgte farver skal egne sig til både skærm og print, og tekst placeret på disse farver skal være let læselig.

Til orientering anvendes der i grunddatamodellering disse specifikke farver:

* Klasser: Sandfarvet (RGB: 254,250,247, HEX: #FEFAF7)  
  eksempel på sandfarve 
* Indlånte klasser: Blå (RGB: 135,205,235, HEX: #87CDEB)  
  #Decorative
* Strukturerede datatyper: Gul (RGB: 251,249,198, HEX: #FBF9C6)  
  #Decorative
* Enumerationer: Grøn (RGB: 232,253,227, HEX: #E8FDE3)  
  #Decorative

Man kan eventuelt anvende yderligere visuelle markeringer, der fremhæver aspekter af modelleringen. I så fald bør disse markeringer:

* udelukkende fremhæve - ikke tilføje - information
* supplere - ikke erstatte - de fælles anbefalede farver
* anvendes konsistent
* forklares på diagrammet

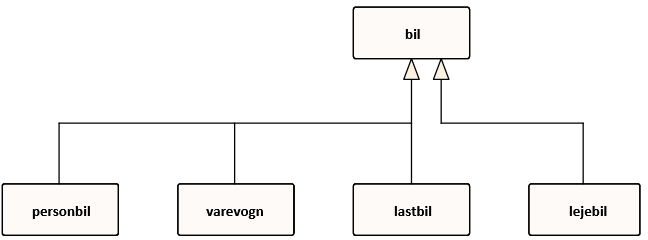
### Placer overordnede klasser over deres specialiseringer

Lad så vidt muligt generaliseringspilene pege opad.

Når man anvender en hierarkisk relation som en specialisering, så er det lettere at afkode modellen hvis overordnede klasser også er placeret over underordnede klasser. Er dette ikke muligt, fx i forbindelse med flere omfattende generaliseringssæt, er det næstbedste at placere specialiseringerne til højre for den overordnede klasse, således at den overordnede klasse kommer først i læseretning.

### Anvend generaliseringssæt

Generaliseringssæt (hvor flere generaliseringspile samles i en spids) kan bruges til at vise at specialiserede klasser/begreber er specialiserede efter samme inddelingskriterium, fx formål eller funktion. Dette giver diagrammer mere information samtidigt med at det bliver mere roligt at se på. Læs mere om generaliseringssæt i [afsnittet om UML-modellering](#_ma2uftknzky).



*Figur xx: Eksempel på anvendelse af generaliseringssæt i begrebsdiagram*

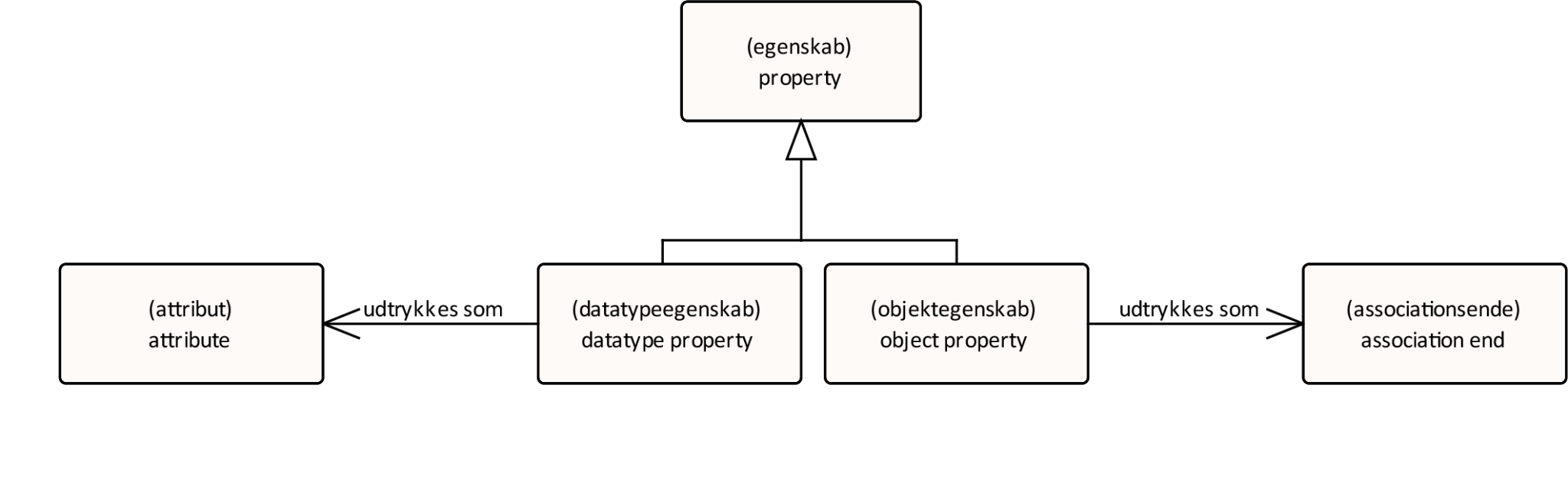
### Angiv retning på begrebsrelationer

UML-associationer kan angives med eller uden navigabilitet, der vises som en pil i enden af associationer. Dette omtales ofte mindre præcist som retning. Navigabilitet er ikke det samme som, men ofte sammenfaldende med læseretning.

I begrebs- og informationsmodeller giver det ikke meget mening at tale om navigabilitet, og associationer bruges specifikt til at udtrykke relationer mellem begreber. Derfor kan pilene benyttes til at angive relationens retning. Man kan også anvende læseretningspile i forbindelse med associationsnavnet. For mange, især de uden stor erfaring med UML, er det dog lettere at læse modeller med retning på relationerne. Desuden er det i nogle værktøjer svært at styre retningen på læseretningspile præcist.

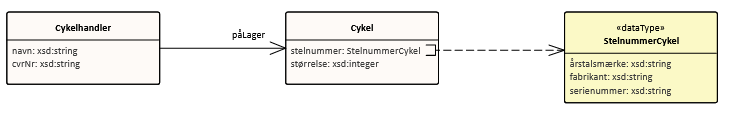
### Brug associationer til klasser og attributter til datatyper

I UML er både attributter og associationsende egenskaber og kan teknisk have både datatyper og klasser som udfaldsrum. Det giver dog mere ensartede og letlæselige modeller hvis objektegenskaber modelleres som *associationsender* og datatypeegenskaber som *attributter*.



Objektegenskaber har klasser som udfaldsrum, og datatypeegenskaber har datatyper som udfaldsrum. Her kan kan være tale som primitive datatyper, strukturerede datatyper eller enumerationer. For strukturerede datatyper og enumerationer hjælper det læseren hvis der oprettes en dependency mellem atrributtet og datatypen.

Nedenfor ses objektegenskaben ‘påLager’ med klassen ‘Cykel’ som udfaldsrum modelleret som associationsende. Datatypeegenskaberne ‘navn’, ‘cvrNr’, ‘stelnummer’ og størrelse’ er modelleret som attributter. Heraf har ‘stelnummer’ en struktureret datatype som udfaldsrum og sammenhængen mellem elementerne er illustreret med en dependency.



*Figur xx : objektegenskaber som associationsender og datatypeegenskaber som attributter*.

## Eksempel på UML-diagram

## UML-diagram uden krydsende associationer, med anbefalede farver og specialiseringer placeret under det de er en specialisering af.

*Figur xx : UML-diagram hvor anbefalinger følges*

## 

## Anbefalinger til organisering og kommunikation

### Skab overblik med opdeling og overordnede diagrammer

I forbindelse med formidling af en større model anbefales det at lave en slags oversigts- eller indflyvningsdiagram, der indeholder de mest centrale elementer (uden eventuelle attributter) og relationer, og som kan fungere som en introduktion til modellen.

Det kan også være hensigtsmæssigt at dele en stor model op i flere deldiagrammer. Generelt bør et diagram kunne være på en A4-side (gerne med landskabsorientering) eller en Powerpoint-slide, samtidigt med at det har en størrelse hvor al tekst er nemt læsbart.

Der findes diagrammer hvor elementerne naturligt falder ind i forskellige grupperinger. Det kan være indflyvningsdiagrammer hvor et par elementer repræsenterer et emne der udfoldes i et andet diagram eller anvendelsesmodeller der sammensætter elementer fra forskellige kernemodeller. Her kan det være en god idé at visualisere grupperingerne ved at indramme elementer der hører sammen, gerne med navngivelse af rammen.

I fortællingen om hvilke kernemodeller der indgår i en en given anvendelsesmodel kan det være en idé at udarbejde et pakkediagram som viser hvilke kernemodeller en anvendelsesprofil sammensætter og/eller indramme elementer fra samme kernemodel med pakken på diagrammet.

### Organiser diagrammet ift. rækkefølge hvor relevant

I nogle situationer er der en indbygget logisk rækkefølge eller kronologi mellem elementerne.. Fx kommer en ansøgning om en ydelse før bevillingen af ydelsen, som igen kommer før udbetalingen af ydelsen. I disse tilfælde kan det lette læsningen af diagrammet, hvis rækkefølgen følger almindelig læseretning, således at det der kommer først står længst til venstre og det der kommer sidst står længst til højre. Organiseringen i rækkefølge har dog ingen reel betydning – det er udelukkende en læsehjælp. Ønsker man at modellere et flow eller lignende anvendes en anden modeltype, fx et procesdiagram.

### Lav også alternative visuelle diagrammer til kommunikationsformål

Det kan være relevant at formidle det centrale indhold af en begrebs- eller datamodel visuelt med andre grafiske metoder til specifikke kommunikationsformål.

Det er typisk uformelle modeller med kasser, figurer og streger eller illustrationer med ikoner eller andre grafiske illustrationer der repræsenterer modellens elementer og relationer. Se eksempelvis denne [illustration fra Ejendomsinfo.dk](http://ejendomsinfo.dk/grunddata-og-bbr) som viser hvordan grunddata hænger sammen på et overordnet niveau.

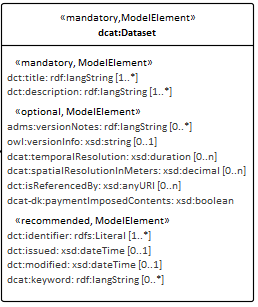
En anden måde at lave diagrammer til kommunikation er at lave simplificerede views at UML-modellen. Sådanne diagrammer kan være velegnede i forbindelse med forretningsgodkendelse af modellen. Det kan fx være nyttigt at lave diagramvisninger, der ikke viser stereotyper, datatyper, prefixes eller aliaser på et andet sprog, selvom dette er en del af den bagvedliggende model.

## Yderligere best practices i international modellering

### Angivelse af kravniveau med supplerende stereotyper

Hvis man i en UML-model ønsker at tydeliggøre de forskellige krav til anvendelse af de forskellige elementer i modellen, bør man anvende supplerende stereotyper til dette. Eksempelvis kan stereotyperne 5<<mandatory>>, <<recommended>>, og <<optional>> anvendes til at angive om udfyldelse af et givet felt er obligatorisk, anbefalet eller valgfrit.

Disse stereotyper anvendes også typisk i anvendelsesprofilerne der udgives af Europa-Kommissionen. Se fx [UML-diagrammet til DCAT-AP 2.0.1](https://joinup.ec.europa.eu/sites/default/files/distribution/access_url/2020-06/8d26212c-3ed4-4c9d-b5a5-0b5934daae34/DCAT_AP_2.0.1.png). I [INSPIRE](https://inspire-regadmin.jrc.ec.europa.eu/dataspecification/themes/br/Chapter5.pdf) data specifikationerne anvendes også stereotypen <<voidable>> til at angive at der ikke eksisterer eller kan afledes en tilsvarende værdi i kildedatasættet.



*Figur xx: Udsnit fra UML-diagram til DCAT-AP 2.0.1*

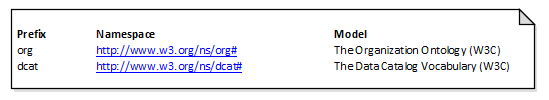
### Angiv gerne namespaces og præfikser i logiske datamodeller

Ved at anvende en forkortelse for namespacet på den model hvor modelelementer er defineret som præfiks for UML-elementnavne, er det til enhver tid tydeligt hvor et givet element kommer fra. Se fx [W3Cs UML-visualisering af DCAT](https://www.w3.org/TR/vocab-dcat-2/). Det er især relevant i anvendelsesmodeller som sammensættes af modelelementer fra forskellige kernemodeller. I kernemodeller kan man eksempelvis også vælge alene at forsyne de indlånte elementer med prefixes så defaultlæsningen bliver at de øvrige klasser er modellens egne.



*Figur xx: ??*

For at gøre det let se hvilke modeller der er genbruges i given model, bør man også indsætte en tekstboks med anvendte prefixes og namespaces i modellen, fx



*Figur xx: Tekstboks med anvendte prefixes*

# 