

# Strukturella Metadata - Referensgrupp 18

## nov

### Närvarande

Andrew Mercer (Riksantikvarieämbetet)  
Benny Lund (Bolagsverket)  
Cilla Öhnfeldt (Naturvårdsverket)  
Erik Mossing (Bolagsverket)  
Fredrik Emanuelsson (Riksarkivet)  
Fredrik Klingwall (Kungliga biblioteket)  
Fredrik Persäter (Lantmäteriet)  
Johan Oelrich (Arbetsförmedlingen)  
Kristofer Gäfvert (Lantmäteriet)  
Lars Näslund (Trafikverket)  
Marcus Smith (Riksantikvarieämbetet)  
Marjan Akhavan (E-hälsomyndigheten)  
Martin Brandhagen (Vetenskapsrådet)  
Matthias Palmér (Digg)  
Mattias Ekhem (Digg)  
Michalis Vassilas (Digg)  
Olov Johansson (SGU)  
Olof Olsson (Svensk nationell datatjänst)  
Patrik Wahlgren (SCB)  
Stefan Jakobsson (Svensk nationell datatjänst)  
Ulrika Domellöf Mattson (Digg)

## Bakgrund

Syftet med referensgruppen är att arbeta fram en profil för att främja interoperabilitet vid datadelning genom återanvändning av klasser, egenskaper och koncept. Klasser och egenskaper definieras så att de blir enklare att återanvända och att kombinera på olika sätt. Att beskriva hur de kombineras kallar vi en applikationsprofil. Arbetet sker inom byggblock metadata, en del av Ena, Sveriges digitala infrastruktur och drivs av Myndigheten för digital förvaltning.

Allt material kommer att publiceras till Github: <https://github.com/diggsverige/interoperable-specifications>

Materialet som tagits fram är på engelska är för att underlätta dialog med bland annat SEMIC (Europeiska kommissionens initiativ för att förbättra den semantiska interoperabiliteten) på EU-nivå.

Referensgruppsdeltagarnas namn kommer att synas på Digg:s GitHub. Om ni inte vill att era namn ska synas på GitHub hör av er till Ulrika Domellöf Mattsson.

## Genomgång av konceptet (se även ppt)

Syftet med interoperabla specifikationer är att stödja återanvändning mellan specifikationer. Förhoppningen är bland annat att förenkla, effektivisera och förbättra kvaliteten i anslutning till att nya informationsmodeller skapas då myndighetsaktörer inte behöver uppfinna hjulet på nytt utan kan återanvända befintliga klasser.

**En specifikation** är en behållare med resurser. En samling av en mängd olika artefakter såsom informationsmodeller, guidelines diagram och exempeldata. Förr var specifikationer ofta i PDF-format (goda exempel, modell etc.).

**En interoperabel specifikation** i sin tur möjliggör återanvändning av klasser, egenskaper (attribut och relationer) samt begrepp. Klasser är i sammanhanget en kategorisering av data eller identitet med liknande egenskaper. Egenskaper som i UML betecknas som attribut (i RDFS talar man om properties). Man paketerar ofta klasser och properties i så kallade datavokabulärer. Termen kommer från RDFS (data vocabulary).

En traditionell informationsmodell är inte gjord för att återanvändas, för det ändamålet föreslår vi att vi använder en applikationsprofil. En interoperabel specifikation innehåller en applikationsprofil vars syfte är att återanvända och kontextualisera befintliga klasser, egenskaper och koncept.

I det här sammanhanget har vi större fokus på properties, då de definieras oberoende av klasser. Properties återanvänds direkt inne i applikationsprofilen och sub- och egenskapsklasser undviks. När man introducerar subklasser ökar komplexiteten och ansvaret trycks neråt.

Vi tillåter enligt modellen att egenskaper återanvänds i nya sammanhang vilket gör att vi måste vara tydliga. Det här är inte helt självklart. Ju fler subklasser vi introducerar desto mer kunskap behöver vi ha om subklasserna och vilket i sin tur ökar komplexiteten.

Utgångspunkt - Vi erbjuder ett stort utbud av legobitar istället för att skapa subklasser.

### Föreslagna standarder för att beskriva interoperabla specifikationer

- RDFS för vokabulär (klasser och egenskaper), är den standard som rekommenderas i det europeiska ramverket för interoperabilitet.
- SKOS för begrepp. Är ett domänoberoende språk för att uttrycka begrepp, används idag på Sveriges dataportal för begrepp. För att hålla det enkelt är förslaget att inte gå så långt som att försöka särskilja mellan är och är en del av.
- SHACL för att beskriva Applikationsprofiler (här är det lite svårare då det inte råder någon riktigt konsensus i frågan)
- SVG för att beskriva diagram då det finns allmänt stöd i alla webbläsare.
- PROF för att knyta ihop de olika specifikationsdelarna. Används redan idag för att beskriva specifikationer på Sveriges dataportal.

### Angående framtagna modell för applikationsprofiler

Applikationsprofiler baseras på andra applikationsprofiler (DCAT-AP-SE är en applikationsprofil som bygger på DCAT-AP). Detta gäller även klassprofiler. Kan inte vara så exakt att vi säger extends (utan man säger baserad på). Undantaget är klassprofiler där vi likt DCAP-AP-SE kanske enbart översätter en beskrivning och där relationen extends är tillämpbar (starkare än Based on (baseras på)).

### Diskussion

Ingår alla delar (klasser, egenskaper och begrepp) ingår i Applikationsprofilen (AP)? Tanken är att det är applikationsprofilen som knyter ihop alla delar. Det måste alltid finnas en applikationsprofil i en interoperabel specifikation. Datavokabulär måste också finnas. Hur är det med terminologi, är den valfri?

Bra att förklara AP med exempel. Den interoperabla specifikationen kan visa oss hur vi behöver uttrycka en specifikation kring exempelvis forminnen. Man kan se det som en textuell variant av diagram. Vad är skillnaden mellan en AP och en informationsmodell? En AP behöver vara mer noga eftersom delarna ska gå att återanvända. Större fokus på properties.

DCTAP för att uttrycka applikationsprofiler är kanske något vi kan titta närmare på? Tabulär data, ytterligare en möjlig ingång. Ett sätt att kommunicera med icke-datorvana.

DCAT-AP, Core Person, Dublin Core är några vokabulärer men det finns otroligt många Core Vocabularies, det finns ingen klar bild över hur många Core Vocabularies som ska vara kärnan och inte heller vem som ska lägga upp dem.

Vi behöver fundera kring vilka som lägger in Dublin Core som en Vocabulary Specifikation. Det kanske finns domänspecifika Vocabulary Specifikationer? Kanske lägga upp ett antal som är de viktigaste och sedan får de växa med tiden?

Angående verktyg för att titta på den här typen av specifikationer och hur olika specifikationer hänger ihop gjorde Digg en behovskartläggning i en förstudie:

<https://github.com/diggs sweden/information-models-investigation/tree/main> där finns verktyg som används idag beskrivna. Av behovsinventeringen framgick att en majoritet av tillfrågade aktörer ville använda befintliga EA-verktyg för att beskriva och visualisera informationsmodeller. Att ta fram ett gemensamt verktyg för informationsmodellering är ett för stort åtagande i nuläget. Digg kommer dock inom ramen för en begränsad pilot som pågår parallellt med referensgruppen titta på hur en applikationsprofil skulle kunna visualiseras på Sveriges dataportal.

Varför inte använda OWL och RDFS för att uttrycka en specifikation? Svårt att återanvända klasser i OWL. Man behöver hela tiden skapa nya klasser. Vi kanske behöver lägga till en förklaring kring varför inte OWL istället för SHACL för AP?

**Uppgift - Matthias Palmér ber Fredrik Klingvall lägga upp en issue kopplat till OWL och applikationsprofiler.**

**Uppmaning till övriga att lägga in liknande diskussionspunkter som ärenden i Github.**