

Interoperabla specifikationer

En utredning inom byggblock metadata

Matthias Palmér
Michalis Vassilas

(metadataexpert för Digg)
(byggblocksansvarig)

Byggblock metadata

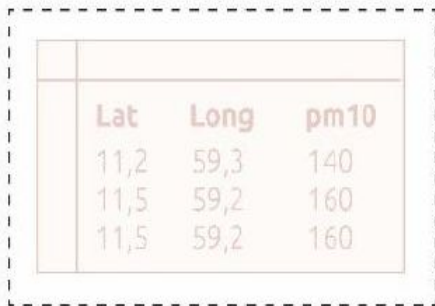
Myndigheten för digital förvaltning, Digg leder arbetet med att etablera en förvaltningsgemensam digital infrastruktur (Ena) för att information ska kunna utbytas på ett säkert och effektivt sätt.

- Samverkan
- Nationella grunddata (ramverk)
- Kompetensområden
- Förutsättningsskapande byggblock

<https://www.digg.se/ledning-och-samordning/ena---sveriges-digitala-infrastruktur>

Beskrivande metadata (Fokus på datamängden)

Titel: Luftkvalitet
Utgivare: Naturvårdsverket
Uppdat: Dagligen

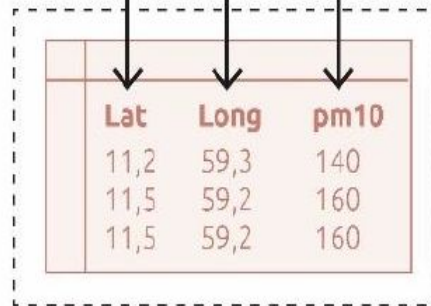


A dashed box containing a table with three columns: Lat, Long, and pm10. The table contains three rows of data. An arrow points from the descriptive metadata box above to the top of this table.

Lat	Long	pm10
11,2	59,3	140
11,5	59,2	160
11,5	59,2	160

Strukturella metadata (Fokus på datauttrycket)

Kolumn: lat
long
pm10
Label: latitud
longitud
partiklar
Datatyp: decimaltal
decimaltal
heltal



A dashed box containing a table with three columns: Lat, Long, and pm10. The table contains three rows of data. Three arrows point from the structural metadata box above to the column headers of this table: 'lat' to 'Lat', 'longitud' to 'Long', and 'heltal' to 'pm10'.

Lat	Long	pm10
11,2	59,3	140
11,5	59,2	160
11,5	59,2	160

Lite förenklat motsvarar beskrivande metadata FAIR principerna Findable och Accessible medan strukturella metadata motsvarar Interoperable och Reusable.

Syfte med referensgruppen

Syftet med referensgruppen är att arbeta fram en profil för att främja interoperabilitet och återanvändning av delar i en specifikation.

En specifikation kan innehålla en blandning av bakgrund, motivering och mer formella beskrivningar. Det är ett paket med olika delar/resurser, vissa riktade mot mänsklig konsumtion, andra maskinläsbara.

Genom att ge användare möjlighet att söka djupare i en datamängd, på den bakomliggande olika delarna möjliggörs interoperabilitet och återanvändning av dem.

Tidigare förstudie: <https://github.com/diggsweden/information-models-investigation>

Referensgruppsarbete

Material:

- Allt läggs up på Github
<https://github.com/diggsweden/interoperable-specifications>
- Bakgrundsmaterial, specifikation, exempel etc.
- Material på engelska då vi förutser Nordisk / Europeisk samverkan

Process:

- Ärendehantering på Github - mallar för styra kommunikationen
- Ändringsförslag (PR = Pull Requests)
- Referensgrupp cirka 6 möten - deltagares namn visas på github
- Referensgruppsarbete November 24 - Februari 25

Interoperabla specifikationer återkoppling

- Återkoppling från SEMIC
- Ändringsförslag: refine och variant
- Ändringsförslag: klargörande kring dubbelanvändning av properties
- Återkoppling från referensgruppen

Återkoppling från SEMIC

- [Översiktlig presentation](#) på engelska
- Mottogs väl, diskussioner om detaljer
- Nästa steg identifierades:
 - Undersöka om vi kan lyfta specifikationen till EU nivå (under SEMICs paraply)
 - Samarbete kring hur OSLO ramverket kan brygga UML till INSPEC
 - Vidare diskussion om [Data Specification Vocabulary](#) kan vara en ytterligare källa (Se [issue 18](#))

En del av diskussionen ledde vidare till ändring om hur man förfinar applikations profiler, dvs hur uttrycker man att något är en förfining.

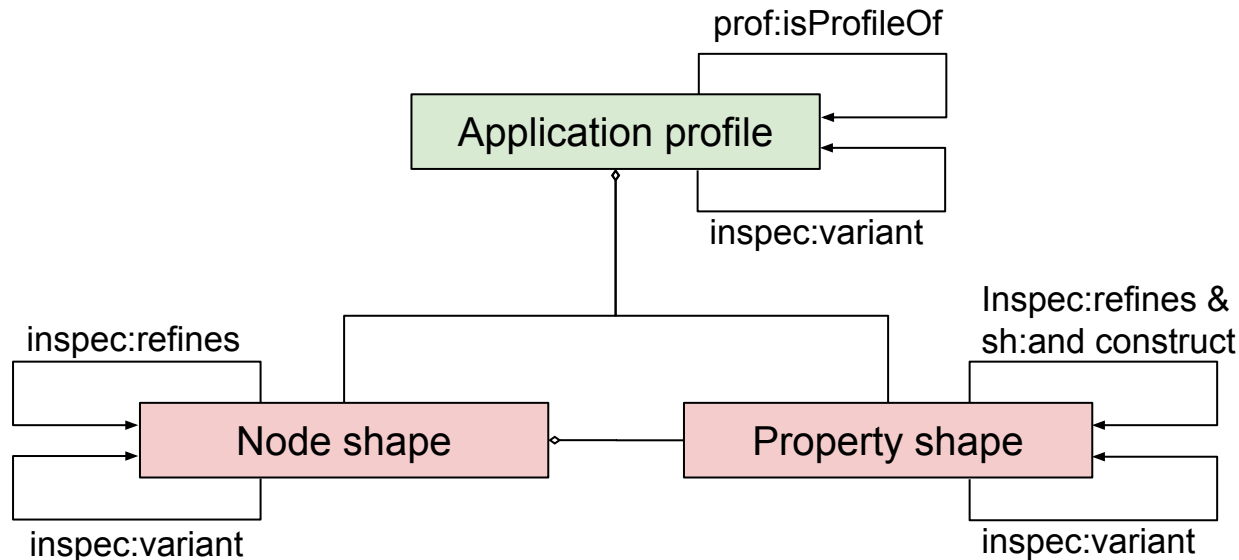
Förslag om refine och variant

Tidigare förslag använde `dcterms:isVersionOf` och `sh:and` konstruktionen.

Nytt förslag:

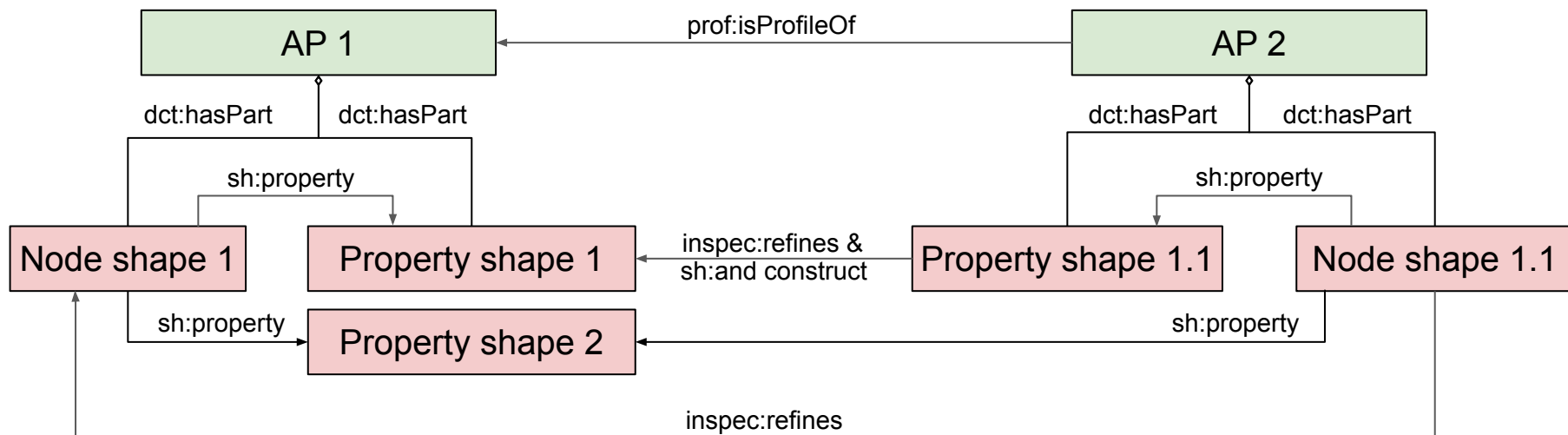
- Introducera properties: **`inspec:refine`** och **`inspec:variant`**
- Dubblera med `sh:and` konstruktion för property shapes (kompatibilitet med SHACL för validering)
- På applikationsprofil nivå används **`prof:isProfileOf`** och **`inspec:variant`**
- Tydliga regler vad det innebär

Uppdaterad övergripande modell



Både isProfileOf och variant är designade så man kan göra en diff mellan applikationsprofiler för att få en översikt vad som skiljer

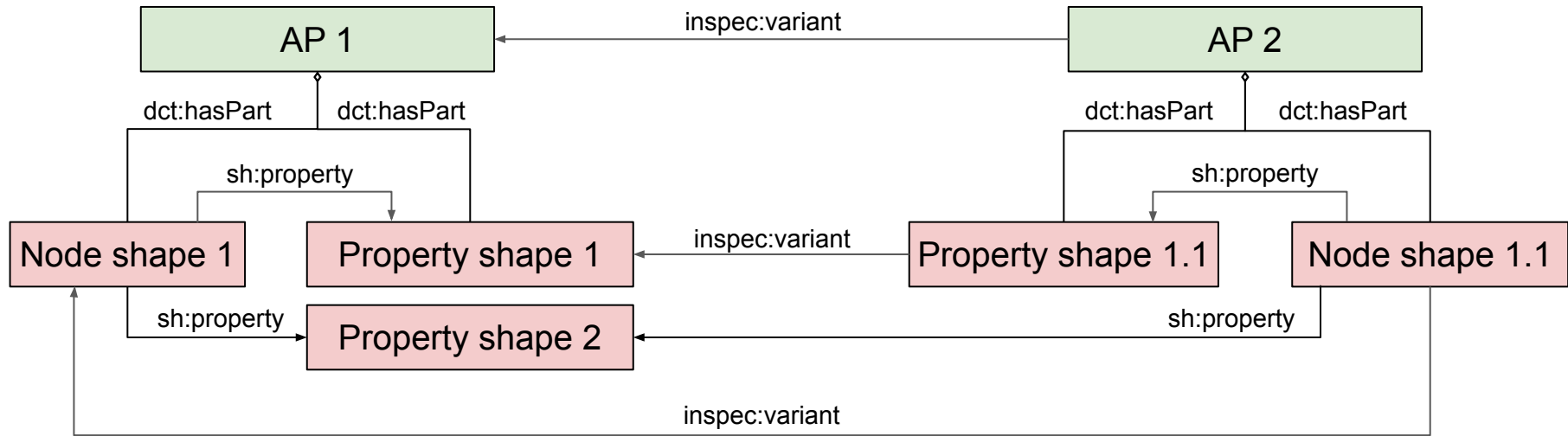
Förfina applikationsprofil



Förfining - all data enligt AP2 är också enligt AP1

Alla property shapes från AP 1 förekommer direkt eller via refines i AP2.

Variant på applikationsprofil



AP2 är en variant av AP1 om alla property shapes från AP 1 förekommer direkt, via refine eller via variant i AP2 (minst en variant, annars är det en förfining).

Återstår kring refine och variant

- Definiera `inspec:refine` och `inspec:variant` mer formellt i RDFS
- Diskutera med SEMIC gruppen om lösningen är tillräcklig

Dubbelanvändning av properties

Det finns situationer när man vill slippa skapa egna properties:

1. Kostnad att etablera ny semantik, samma diskussion som subklassning
2. Kostnad att etablera ny namnrymd och nya properties
(Och ibland är semantiken exakt densamma, bara en annan värdemängd)

Det går att återanvända samma property flera gånger för olika syften, t ex:

```
ex:book    dct:subject    "Miljö"@sv ;  
           dct:subject    po:ENVI .
```

När får man dubbelanvända

1. När man har en annan nodtyp, literal, literal med datatyp, URI eller blank
2. Olika datatyper
3. URI:er som kan separeras utifrån mallar (reguljära uttryck)
4. Fåtal URI:er som kan enumereras (färre än 50)
5. Blanka noder om de har direkta utgående properties som är olika, t.ex. olika `rdf:type`.

Undvik:

- Olika språk
- Utgående properties på URI:er

Återkoppling från referensgruppen

- Språkliga justeringar från Stefan Jakobsson (PR + issue)
- Förslag kring formuleringar från Marjan Akhavan

Dessa håller på att bearbetas