

Joachim Florentz Boye

Fra: Jørn Wildt [jw@cbrain.com]
Sendt: 11. juli 2011 16:38
Til: I-CDI - Sagogdokument
Cc: Michael Madvig; Thomas Qvist
Emne: Svar på høring over udkast til standarden "Person"

Kære IT- og Telestyrelsen

Hermed fremsendes et par kommentarer til standarden "Person" iflg.

<https://www.borger.dk/Lovgivning/Hoeringsportalen/Sider/Fakta.aspx?hpid=2146002719>

Personrelationer

Standarden antager et fast defineret antal relationer mellem personer, f.eks. Moder, Fader og Bopælssamling med flere. Alle disse relationer ser ud til at være samlet i én record med ét felt pr. relation. Iflg. standarden kan listen udvides lokalt efter behov, hvilket er ganske fint.

Jeg vil dog foreslå en lidt mere fleksibel struktur, hvor en person har en liste af relationer (med "virkning") til andre personer, og at hver af disse relationer dekorerer med en type, der afgør hvilken form for relation der er tale om. Typen kan gøres unik og kan udvides lokalt hvis der anvendes en URL, som det kendes fra mange andre standarder på nettet.

Eksempel: Personen "Gitte" har moderen med ID = 1, faderen med ID = 2, og nærmest-kontaktperson med ID = 3 (dette er en ny type relation). Det kunne repræsenteres ved:

```
Gitte.Relationer[0] = { Type = http://itst.dk/oio/personrelationer/moder, PersonId = 1 }
```

```
Gitte.Relationer[1] = { Type = http://itst.dk/oio/personrelationer/fader, PersonId = 2 }
```

```
Gitte.Relationer[2] = { Type = http://andetprodukt.dk/oio/personrelationer/narmestekontakt,  
PersonId = 3 }
```

Med ovenstående repræsentation, så kan man præsentere brugeren af systemet for en liste af personens relationer uden at kende samtlige muligheder for relationstyper.

Man kan yderligere udvide relationen med et "Brugervendt titel" felt, så man også i systemets GUI kan skrive en pænt formateret beskrivelse af relationen. Se f.eks. hvordan det er gjort med ATOM's relationslink (<http://www.ietf.org/rfc/rfc4287.txt> afsnit 4.2.7).

Kontaktkanaler

Kontaktkanaler kan transformeres på samme måde som personrelationer nævnt ovenfor, hvormed man opnår en mere generisk tilgang til kontaktkanaler.

REST API

Jeg har ikke tænkt mig at prædike et skift fra SOAP webservices til REST, men vil gerne påpege nogle ting i standarden som ville forbedres ved at inddrage elementer fra REST. Det er først og fremmest problemet med at fremfinde en given entitet ud fra et UUID. Hvis systemet f.eks. skal hente en person med UUID = 1, så skal den på forhånd vide hvilket serviceinterface/WSDL, der håndterer personer, og derefter vide hvilken metode der skal kaldes for at hente personen. Det gør det umuligt at have flere forskellige kilder for personer, og kræver at alle del-systemer skal konfigureres med WSDL-referencen. Havde man derimod angivet en person-reference som en komplet URL, så var der ingen tvivl – så var det bare at

lave en HTTP GET på den angivne URL.

Hvis man så ændrede alle relationer til at være URL'er i stedet for UUID'er, ville man opnå at få et API, som kunne "browses" med en helt almindelig browser – hvilket ville gøre debugging en del nemmere. Listen af fordele ved REST er længere endnu, men det var vist ikke det I oprindeligt spurgte om 😊

Vi har selv hos cBrain lavet et REST API ovenpå Sag/dokument-standard, hvorfra der kan laves fritekstsøgning og bladres frit mellem forskellige sager, akter, dokumenter og parter (selvfølgelig med behørig adgangskontrol). Det giver vi gerne en demo af.

Mvh Jørn Wildt
cBrain