

Kapitel 6 - Regler om generelle egenskaber

Oprindelig kilde	WorkZone - 1442-0818
Forfatter	Modelsekretariatet
Oprettet	Aug 31, 2016
Version	1.2
Ændret	Oct 30, 2023
Sidehistorik	• Se sidehistorik

Kort om Generelle Egenskaber

- [6.1 Alle modelentiteter skal modelleres med persistent, unik identifikation](#)
- [6.2 Alle modelentiteter skal modelleres med status](#)
- [6.3 Alle modelentiteter skal understøtte dobbelthistorik og angivelse af aktører](#)
- [6.4 Alle modelentiteter bør understøtte beskedfordeling](#)

[Til toppen](#)

6.1 Alle modelentiteter skal modelleres med persistent, unik identifikation

Regel

Alle entiteter skal modelleres med persistent, unik identifikation.

Rationale

Alle data-objekter, skal have et globalt unikt id, som ikke ændres i data-objektets levetid. Det er nødvendigt at have en unik identifikation af et data-objekt på tværs af Grunddatamodellen, for at sikre en fælles høj datakvalitet.

Ofte vil data-objektet, ud over den unikke identifikation, have en eller flere forretningsnøgler, for eksempel har en matrikel et matrikelnummer. Men forretningsnøglerne kan ikke stå alene, da Grunddatamodellen generelt skal understøtte historik, hvilket betyder, at data-objektet kan have forskellige forretningsnøgler over tid, ligesom den samme forretningsnøgle løbende kan indgå i flere forvaltningsobjekter.

Derfor er det essentielt, at objektets identifikation er persistent i hele data-objektets levetid.

Ved at lade ID være en HTTP-URI (se Implikation) bliver URI'en "opløsbare", se <https://arkitektur.digst.dk/node/588>

Dette giver ID'en to funktioner:

- Global entydig identifikation af en entitet på tværs af Grunddatamodellen og på tværs af øvrige modeller.
- Reference direkte til data-objektet, således at et givet data-objekt potentielt kan adresseres direkte - også ved hjælp af en generisk web-applikation.

Implikationer

Alle entiteter skal modelleres med attributen 'id'

Id	
Betydning	Unik identifikation af objektet
Værdi	Objektets unikke id
Datatype	Identifikation
Krav	Obligatorisk

Datatypen **Identifikation**.

Datatypen er en del af Grunddatatyperne, som publiceres og vedligeholdes af Grunddatasekretariatet; se <https://github.com/digst/grunddata-model-rules-tool-support>

Attributterne i Identifikation kan kombineres til en HTTP-URI, som specificeret i [IETF RFC 3986](#) (http-scheme). Denne vil være universelt unik, og vil være opløsbar.

En HTTP-URI kan sammensættes på følgende måde - baseret på IETF RFC 3986, afsnit 3, Syntax Components: Skemaangivelse + "://" + Autoritet + "/" + Sti + "#" + Objektidentifikator

Indhold	
Komponent	Værdi
Skemaangivelse	"http"
Autoritet	"data.gov.dk"
Sti	Valgfri, men se nedenfor
Objektidentifikator	Valgfri, men se nedenfor

HTTP-URI'en skal i sin helhed være universelt unik.

Dette kan opnås på en række måder:

- Manglende eller generisk Sti, universelt unik Objektidentifikator
Eksempel: http://data.gov.dk#e1f9a650-21c7-11e3-8224-0800200c9a66
- Unikt afgrænsende Sti, lokalt unik Objektidentifikator
Eksempel: http://data.gov.dk/lufthavnskoder/IATA#CPH
- Unikt afgrænsende Sti, universelt unik Objektidentifikator
Eksempel: http://data.gov.dk/person#08f88d52-4add-4ec6-9a80-18caaf0f0e7b

Det er domænets ansvar at sikre, at den samlede, identificerende HTTP-URI er universelt unik.

Det anbefales, at Objektidentifikatoren er af typen UUID (Universally Unique Identifier - [IETF RFC 4122](#)) - på den måde bliver identifikationen universelt unik uanset hvilket namespace, der anvendes - dette vil være relevant i et fremtidigt, løst koblet data-scenarie, hvorfor der også på sigt vil være behov for, at alle datakilder udstyrer data med UUID.

Komponenterne Skemaangivelse, Autoritet og Sti kan sammensættes til et Namespace, som lagres i den ene attribut - namespace i datatypen. Objektidentifikatoren lagres i den anden attribut, som lokalid

Attributter

namespaces

Betydning	Identifikation af et namespace inden for hvilket lokalId er unik
Værdi	En HTTP-URI uden Objektidentifikator (Skemaangivelse + “://” + Autoritet + “/” +Sti)
Datatype	CharacterString
Krav	Obligatorisk

localID	
Betydning	Identifikation af objektet
Værdi	Objektets id
Datatype	CharacterString
Krav	Obligatorisk

6.2 Alle modelentiteter skal modelleres med status

[Til toppen](#)

Regel

Alle modelentiteter skal modelleres med status, der klart angiver, hvor et forvaltningsobjekt er i sin livscyklus.

Rationale

Forvaltningsobjekter gennemløber typisk en livscyklus. For eksempel kunne en livscyklus for en bygning være: “Forslag > Under projektering > Under opførelse > Ibrug > Under nedrivning > Historisk”. Forretningsdomænet kan opstille regler for hvilke statusser, der er gyldige for et givet objekt, og for, hvordan forvaltningsobjektet kan gennemløbe disse.

Disse tilstande skal placeres og udstilles i et vedtaget udfaldsrum, som modelentiteten skal referere til.

Dataobjektets status udtrykker dataobjektets relevans for databrugeren. Forretningsmæssigt giver det god værdi at udstille et dataobjekts status eksplicit, frem for at lade databrugeren analysere sig frem til informationen. Anvendelsen af status er med til at sikre en høj datakvalitet og potentielt nedbringe udviklingsomkostninger, da der skal implementeres mindre forretningslogik og fejlhåndteringslogik.

Implikationer

Alle modelentiteter skal have attributten ‘status’

Status	
Betydning	Forvaltningsobjektets status
Værdi	En status
Datatype	enumeration, kodeliste, klassifikation
Udfaldsrum	Domænespecifik liste, værdien må ikke være tom
Krav	Obligatorisk

Der skal i domænemodellerne defineres statustilstande for alle forvaltningsobjekter. Disse statustilstande defineres af den modelansvarlige, og publiceres i den fælles model. Tilstandene kan modelleres som enumeration, kodeliste eller klassifikation (se [Note om kodede værdier i afsnit 5.6](#)).

NOTE

Ideelt bør statusser betegnes med de reelle tilstande, ikke senest overståede milepæl. For eksempel "i brug" frem for "ibrugtaget" - ibrugtaget vil objektet jo vedblive at være.

[Til toppen](#)

6.3 Alle modelentiteter skal understøtte dobbelthistorik og angivelse af aktører

Regel

Alle modelentiteter skal modelleres med angivelse af registrering, virkning og aktører

Rationale

Dobelthistorik:

På tværs af Grunddata er der behov for at implementere dobbelthistorik, for at kunne understøtte et samlet krav om revisionsspor. Dataobjekter skal med andre ord kunne rekonstrueres på en måde, hvor der er styr på objektets sammensætning eller tilstand til et givet tidspunkt. Formålet hermed er blandt andet at dokumentere det konkrete historiske beslutningsgrundlag i forbindelse med for eksempel sagsbehandling. Dobelthistorik modelleres ved hjælp af bitemporale egenskaber. Det dobbelte består i, at to tidsaspekter "virkningstid" og "registreringstid" håndteres i sammenhæng.

Registreringstid:

Tidsrummet fra versionen registreres i databasen, indtil den enten erstattes af en nyere version eller afregistreres.

Virkningstid:

Tidsrummet, hvor en given version af data svarer til de forhold i virkeligheden, som versionen afbilder.

Aktører:

Oplysning om hvilke aktører, der er ansvarlig for datas indhold, tilfører sporbarhed i forbindelse med revision og anvendelse af data. Aktøren kan være en af en række forskellige typer, fx en organisation, et it-system, en arbejdsfunktion eller en konkret bruger.

Implikationer

Et data-objekt kan bestå af en række (1-*) versioner (ændres en enkelt attributværdi, betragtes dataobjektet som ændret og dermed versioneret). Alle versioner betragtes som dele af et "stamdataobjekt", og har den samme Identifikation.

Alle versioner skal være karakteriseret ved hjælp af deres registreringstid og deres virkningstid. Disse tidsaspekter modelleres ved anvendelse af attributterne registreringFra, registreringTil, virkningFra og virkningTil.

Enhver version af et stamdataobjekt identificeres entydigt af objektets unikke identifikation kombineret med attributten registreringFra.

Når en bruger til et bestemt formål skal fremfinde den eller de relevante versioner af et objekt, anvendes objektets identifikation samt en kombination af registreringstid, virkningstid og/eller objektets status.

Virkningstidsrummet skal fortolkes sådan, at virkningsintervallet er inklusiv starttidspunktet men eksklusiv sluttidspunktet.

Hvorvidt der kan være "huller" i Virkningstiden - perioder hvori virkeligheden ikke er afspejlet i data - er op til det enkelte datadomæne at beslutte og dokumentere.

Til hver version af et dataobjekt skal der knyttes aktører i betydningen:

- Reference til den aktør, der afstedkommer iværksættelse af virkningsperioden
- Reference til den aktør, der har foretaget registreringen

Værdien kan være en reference til fx en organisation, et system eller en sagsbehandler se [afsnit 3.3.1](#) og [regel 5.10](#).

Fuld temporal rekonstruktion af dataobjekter forudsætter, at alle ændringer i dataobjektet gemmes med de nødvendige tidsmarkeringer. Derfor indeholder denne regel et krav om, at data opbevares i registre, som er implementeret på en måde som gør dette muligt. En sådan implementering indbefatter dels en fysisk datamodel som tilføjer en ny række til datatabellen for hver ny version af data, på en måde så alle ændringer kan genfindes dels forretningsregler for, hvordan tidsmærkerne tilknyttes.

Attributter:

Hver modelentitet med stereotypen DKObjekttype - modsvarende et forretningsobjekt - modelleres med følgende attributter:

registreringFra	
Betydning	Tidspunktet hvor registreringen er foretaget
Værdi	Tidspunkt
Datatype	DateTime (ISO 8601), værdien må ikke være tom
Krav	Obligatorisk

registreringTil	
Betydning	Tidspunktet hvor en ny registrering er foretaget på dataobjektet, og hvor denne version således ikke længere er den seneste.
Værdi	Tidspunkt
Datatype	DateTime (ISO 8601), værdien kan være tom
Krav	Obligatorisk

registreringsaktør	
Betydning	Den aktør der har foretaget registreringen
Værdi	Angivelse af en aktør fx som en reference til en organisationsmodel (se regel 5.10)
Datatype	Domænespecifik aktør, værdien må ikke være tom
Krav	Obligatorisk

virkningFra	
Betydning	Tidspunktet hvorfra forvaltningsobjektet har virkning
Værdi	Tidspunkt - virkningsperioden inkluderer dette tidspunkt
Datatype	dateTime (ISO 8601), værdien må ikke være tom

Krav	Obligatorisk
------	--------------

virkningTil	
Betydning	Tidspunktet hvor forvaltningsobjektets virkning ophører
Værdi	Tidspunkt - virkningsperioden stopper umiddelbart før dette tidspunkt
Datatype	dateTime (ISO 8601), værdien kan være tom
Krav	Obligatorisk

virkningsaktør	
Betydning	Den aktør der har afstedkommet forvaltningsobjektets virkning
Værdi	Angivelse af en aktør fx som en reference til en organisationsmodel (se regel 5.10)
Datatype	Domænespecifik aktør, værdien må ikke være tom
Krav	Obligatorisk

Undtagelse

CVR's modeller er undtaget fra regel 6.3 for så vidt angår attributterne registreringFra og registreringTil, da CVR ikke indeholder de pågældende data.

[Til toppen](#)

6.4 Alle modelentiteter bør understøtte beskedfordeling

Regel

Modelentiteterne i grunddata bør modelleres således, at dataobjektet kommer til at indeholde informationer, som kan forbedre kvaliteten af hændelsesbeskeder, der udsendes i forbindelse med opdatering af dataobjektet. Disse informationer omfatter den forretningsmæssige kontekst, hvori dataobjektet opdateredes, samt den bagvedliggende forretningsmæssige årsag til opdateringen.

Rationale

Automatiseret beskedfordeling beskrives i [kapitel 3.3.3](#). Der findes på nuværende tidspunkt ikke et samlet billede af, hvordan beskedfordeling skal foregå, hvorfor denne regel er formuleret med det formål, at beskrive rammerne for de data der er nødvendige for beskedfordeling. Yderligere arbejde med fællesoffentlig beskedfordeling kan meget vel resultere i justering af denne regel. Automatiseret beskedfordeling forudsætter, at det for hver ændring af data registreres hvilken forretningssammenhæng, der ligger til grund for ændringen.

Forretningssammenhængen beskrives med tre parametre:

- forretningshændelse, som betegner den begivenhed i virkeligheden ([se afsnit 1.3.2](#)), som udløste ændringen i data
- forretningsområde - den del af den offentlige forretning, som håndterer hændelsen og derved udvirker ændringen i data

- forretningsproces - den manuelle eller it-understøttede proces, hvori forretningsområdet håndterer hændelsen

Hændelse, område og proces er udfaldsrum, som typisk vil skulle modelleres af domænet

- Forretningshændelser vil typisk (ligesom status - [se regel 6.2](#)) være specifikke for det enkelte forvaltningsobjekt - der skal opbygges en datastruktur, der afspejler forretningens viden om hvilke hændelser, der kan påvirke et forvaltningsobjekt
- Forretningsområdet kan specificeres ud fra FORM
- Forretningsprocesser forudsætter en kortlægning af domænets processer

De tre udfaldsrum kan modelleres mere eller mindre komplekst - som simple lister, indlejret i modellen eller som reference til eksterne data, enten i form af kodelister eller klassifikationskomponenter - se NOTE om kodede værdier i [afsnit 5.6](#) samt [afsnit 3.3.2](#). Den modelansvarlige må afgøre på hvilket niveau modelleringen bedst muligt modsvarer forretningskravene.

Implikationer

Hver modelentitet med stereotypen DKObjekttype - modsvarende et forretningsobjekt - kan modelleres med følgende attributter:

forretningsområde	
Betydning	Det forretningsområde som har opdateret dataobjektet
Værdi	Specifikation af et forretningsområde. Domænespecifik liste, værdien kan være tom
Datatype	enumeration, codeList, klassifikation
Krav	Valgfri

forretningsproces	
Betydning	Den forretningsproces, som har opdateret dataobjektet
Værdi	Specifikation af en forretningsproces. Domænespecifik liste, værdien kan være tom
Datatype	enumeration, codeList, klassifikation
Krav	Valgfri

forretningshændelse	
Betydning	Den forretningshændelse, som afstedkom opdateringen
Værdi	Specifikation af en forretningshændelse. Domænespecifik liste, værdien kan være tom
Datatype	enumeration, codeList, klassifikation
Krav	Valgfri
