

Att skriva RDF-trippler med Turtle-notation

Maria Idebrant

18 augusti 2017

Ett av kursmålen är att du ska kunna ”överföra en bibliografisk analys till RDF-form”. Kursboken tar upp RDF och olika skrivsätt för RDF kortfattat på några olika ställen. Som komplement följer här en samlad beskrivning av hur du gör för att skriva i RDF-form med notationen Turtle, vilket är den notation vi kommer använda i kursen.

Jag utgår från att jag vill beskriva vad vår kursbok har för titel och författare och börjar med en allmän förklaring om RDF. I slutet kommer jag in specifikt på Turtle och hur det skrivs.

Samma innehåll kan skrivas på olika sätt

Om jag ska beskriva någonting för dig, till exempel en bok, kan jag berätta att den heter ”The discipline of organizing” och är skriven av ”Robert J. Glushko” och så vidare. Jag kan också skriva det i text, som jag gjorde nu. Men om jag också vill att ett datorprogram, till exempel ett bibliotekssystem, ska kunna använda beskrivningen, då måste jag skriva på ett speciellt sätt.

Hur då? Ja, det beror på programmet. Olika program vill ha beskrivningar skrivna på olika sätt. Det går att dela in olika sätt att skriva beskrivningar i generella grupper. Dessa grupper innebär fortfarande inget bestämt sätt att skriva, men kan vara användbara för att diskutera för- och nackdelar med olika varianter. Glushko kallar det för att det finns olika strukturer - se 8.2.1 Kinds of Structures.

En av dessa strukturer är grafen. I en graf kan olika beskrivningar dela samma värde. Det innebär att om två olika manifestationer har samma författare behöver inte författaren beskrivas en gång per manifestation, utan båda resursbeskrivningar för manifestationerna kan hänvisa till samma beskrivning av författaren.

Grafer kan skrivas som trippler med subjekt, predikat och objekt

Ett speciellt sätt att skapa en graf är att göra det i RDF-form. RDF (Resource Description Framework) är en standard som bestämmer att påståenden ska göras i form av trippler.

En trippel består av ett subjekt, ett predikat och ett objekt. Om vi vill skriva påståendet "Manifestation 1 har titeln The discipline of organizing" i trippelform så måste vi börja med att hitta de tre delarna.

Subjektet är alltid den resurs vi vill beskriva. I detta fall "Manifestation 1".

Predikatet berättar vad vi vill säga om subjektet. I detta fall vill vi säga att manifestationen har en viss titel. Predikat är alltså "har titeln".

Objektet är det vi vill säga om resursen, i det här fallet vad titeln är: "The discipline of organizing".

Skriver vi RDF måste vi använda URI:er

Enligt RDF måste vi använda en unik identifierare för varje resurs vi vill beskriva, en så kallad URI (uniform resource identifier). Var får man tag på den? Ja, ska man skapa egna finns det flera saker att tänka på. Till exempel måste den vara unik globalt och inte ändras över tid. I denna kursen behöver du inte kunna skapa egna URI:er, vi lånar istället från andra eller använder förenklade versioner som låtsas-URI:er.

Ett vanligt sätt att skapa en URI är att använda sin webbadress följt av ett id-nummer för resursen. Det gör till exempel Kungliga biblioteket. Deras URI för manifestationen jag vill beskriva är <http://libris.kb.se/resource/bib/16634377>. Vi lånar den i det här exemplet.

Vi måste också använda en unik identifierare för predikatet. Den hämtar man från någon metadatavokabulär, det vill säga en lista med etiketter/URI:er och vad de står för. Dublin Core är en sådan lista, den innehåller 15 olika element, bland annat "title" och "creator". För mitt exempelpåstående passar "title", då title har definitionen "A name given to the resource.". URI är: <http://purl.org/dc/elements/1.1/title>. Du behöver inte kunna hitta URI:er på egen hand i denna kurs, men vill du se hela listan med element i Dublin Core så finns de på <http://dublincore.org/documents/dces/>.

Ibland vill man använda en URI även för objektet. Det är i de fall objektet är en annan entitet/resurs som har en egen beskrivning. I detta fallet är det inte en resurs som är på objektspositionen, utan resursens titel. Då använder vi ett textvärde, i det här fallet "The discipline of organizing".

De tre delarna i vår första trippel är alltså
Subjekt: <http://libris.kb.se/resource/bib/16634377>
Predikat: <http://purl.org/dc/elements/1.1/title>
Objekt: "The discipline of organizing"

Vill vi däremot ange att manifestationen är framställd av en viss person, då passar det bättre med en URI på objektspositionen. Det gör att vi inte behöver upprepa information som namnet i flera olika resursbeskrivningar, utan kan hämta den från den beskrivning

som finns för just den personen. Vi behöver också byta ut predikatet, så att det berättar att objektet är en upphovsperson.

Predikatet hämtar vi återigen från Dublin Core. Där finns elementet "creator" som har definitionen "An entity primarily responsible for making the resource." Det passar bra för att beskriva en författare. URI är: <http://purl.org/dc/elements/1.1/creator>

Glushko har ingen URI i Libris (än), för exemplets skull använder vi istället: <http://libris.kb.se/resource/auth/siffror> (det är så hans URI skulle sett ut om den fanns, men där "siffror" varit hans id-nummer i Libris)

De tre delarna i vår andra trippel är alltså
Subjekt: <http://libris.kb.se/resource/bib/16634377>
Predikat: <http://purl.org/dc/elements/1.1/creator>
Objekt: <http://libris.kb.se/resource/auth/siffror>

RDF kan skrivas på flera sätt där Turtle är ett

RDF handlar bara om vilken struktur vår beskrivning har, det vill säga att vi gör beskrivningen i trippel-form. RDF bestämmer inte vilka tecken vi ska använda för att skriva trippeln, med andra ord: vilken syntax som ska användas. Därför behöver man också bestämma sig för en syntax. För att göra det lättare att överföra data mellan olika program kommer olika organisationer ibland överens om standarder, det vill säga regler för hur man ska skriva en viss sak. Turtle är en sådan standard för att skriva RDF. Det finns andra, till exempel går det att skriva RDF i XML-form (vilket vi gjorde/kommer göra i KO2-kursen).

Okej, kom till sak - hur skriver man Turtle då?

URI:er skrivs alltid inom <>, exempel:

```
<http://libris.kb.se/resource/bib/16634377>
```

Textvärden skrivs alltid inom "", exempel:

```
"The discipline of organizing"
```

För att skriva trippeln som helhet finns två varianter.

Variant 1: En trippel per rad

Det lättaste sättet att skriva Turtle är att ange en trippel per rad. Varje rad skrivs i ordningen: subjekt predikat objekt. Det är mellanslag mellan varje del och raden avslutas alltid med punkt. Våra två tripplar blir då:

```
<http://libris.kb.se/resource/bib/16634377> <http://purl.org/dc/elements/1.1/title> "The discipline of organizing".  
<http://libris.kb.se/resource/bib/16634377> <http://purl.org/dc/elements/1.1/creator> <http://libris.kb.se/resource/auth/siffror>
```

Variant 2: Förkortad turtle

Om man ska skriva flera trippler som handlar om samma subjekt kan man skriva förkortad Turtle. Då skriver man:

subjekt predikat objekt

som innan, men istället för att avsluta med punkt avslutar man med semikolon ;

Det betyder att nästa rad kommer att handla om samma subjekt. På nästa rad skriver man bara

predikat objekt

och avslutar med semikolon om det kommer fler rader om samma subjekt, annars med punkt.

Exemplet ovan som förkortad Turtle:

```
<http://libris.kb.se/resource/bib/16634377> <http://purl.org/dc/elements/1.1/title> "The discipline of organizing";  
<http://purl.org/dc/elements/1.1/creator> <http://libris.kb.se/resource/auth/siffror>.
```

Här avslutas första raden med semikolon eftersom det kommer en till rad om samma subjekt. Andra raden avslutas med punkt eftersom det inte kommer någon ny rad om samma subjekt efteråt. Det räcker för övrigt med att man missar ett enskilt tecken i trippeln för att det ska bli obegripligt för program som ska läsa Turtle-koden.