

# Human IT

Tidskrift för studier av IT  
ur ett humanvetenskapligt perspektiv

## Resource Description Framework

Metadata för Internet

av [Ulf Kronman](#) och [John Parnefjord](#)

---

*For quite some time now, there has been a need for more organized descriptions of the information resources on the Internet. Several standards for metadata have been developed but not reached a large group of users since They have been too limited or too specialized.*

*During 1999 the Web Consortium W3C has developed the Resource Description Framework – RDF - as a standardized framework for metadata on the Internet. RDF is based on the Extensible Markup Language - XML - and is destined to be the unified standard for metadata on the Internet in the near future. RDF is designed to be an expandable framework where several systems of metadata can coexist and be used to describe the same resource at the same time. In RDF these metadata systems are called schemas. Schemas can be developed locally without any central coordination and control as long as they follow the RDF specification. The modularity of RDF will probably ensure the future of the metadata standard Dublin Core, as RDF permits the addition of local, specialized schemas, without any changes having to be made in the Dublin Core element set. RDF is still immature. Tools for production and interpretation of RDF have yet to be developed before it can become a working standard. For organizations that are about to develop their own systems for metadata still, it is recommendable to start using RDF as soon as possible.*

---

### Innehåll

- [1. Vad är metadata?](#)
- [2. Behövs metadata på Internet?](#)
- [3. Metoder för metadata på Internet](#)
- [4. HTML META](#)
- [5. Dublin Core](#)
- [6. RDF – Resource Description Framework](#)
- [7. RDFs datamodell](#)
- [8. RDF syntax](#)
- [9. RDF schema](#)
- [10. Dublin Core i RDF](#)
- [11. Ramverk](#)

[12. Innehåll](#)

[13. Utbyggd RDF](#)

[14. RDF i HTML](#)

[15. RDF i praktiken](#)

[16. Slutsatser](#)

[Om författarna](#)

---

## 1. Vad är metadata?

Prefixet *meta* brukar i allmänhet användas för att beteckna att man talar *om* något – att man höjer abstraktionsnivån och försöker beskriva eller förklara något. Med den definitionen betyder metadata att man använder data för att beskriva eller förklara innebörden i vissa andra data. Metadata är således data med *information om* data.

Metadata är något vi använder dagligen, oftast utan att tänka på det. Ta följande exempel: siffrorna 6453782 betyder troligen ingenting för dig, men om vi tillför lite metainformation genom att skriva *Tel: 6453782* blir det genast klart för de flesta av oss att siffrorna ska tolkas som ett telefonnummer. Datat *Tel:* har alltså givit datat 6453782 en innebörd, en tolkning. Man kan alltså säga att metadata hjälper oss att omvandla data till meningsfull *information*.

Ett traditionellt bibliotek är ett annat bra exempel på användning av metadata. Bibliotekskatalogen är ju inget annat än en stor samling metadata över de böcker och tidskrifter som finns i biblioteket. Utan uppgifterna i katalogen skulle det bli mycket svårt och tidsödande att hitta det man söker.

[\(Åter till början av artikeln\)](#)

## 2. Behövs metadata på Internet?

Med sina en miljard webbsidor är Internet idag en ovärderlig källa till information. Det är dock ofta svårt att hitta det man söker ute på nätet. Det vanligaste sättet att navigera på nätet är att söka hos de webbplatser som samlar in webbsidor och erbjuder dem i söktjänster med namn som till exempel AltaVista, HotBot, Evreka, EuroSeek och Fast Search. De program som samlar in dokumenten från nätet och gör dem sökbara läser av och förtecknar varje ord i dokumenten. Det innebär att man oftast kan hitta ett dokument utan att några metadata behöver tillföras, till skillnad från en traditionell bibliotekskatalog.

Bristen på metainformation i de stora söktjänsterna leder dock ofta till att man får onödigt många sökträffar och låg precision i sin sökning. Om jag till exempel använder ordet *Strindberg* för att försöka hitta texter av Strindberg i söktjänsten Evreka får jag massvis med förslag på sidor. Eftersom August Strindberg är en berömd författare har det skrivits mycket *om* honom. Däremot är det mer sparsamt med sidor med texter där Strindberg själv är *författare*. Om jag skulle kunna tillföra lite metainformation till min sökning som sade att jag bara är intresserad av sidor med texter där Strindberg är författare skulle situationen genast bli bättre. En sådan sökning förutsätter dock att den som har publicerat en text av Strindberg på Internet också har tillfört den metainformationen till resursen, något som tyvärr är mycket ovanligt idag.

Vartefter mängden dokument på Internet ökar kommer alltså modellen med att söka fritt bland alla orden i dokumenten att bli allt mindre hållbar. För att söktjänsterna ska kunna erbjuda förbättrad sökning med hjälp av metainformation krävs då att dokumenten förses med metadata enligt någon fastställd standard.

[\(Åter till början av artikeln\)](#)

## 3. Metoder för metadata på Internet

Det har tagits ett flertal initiativ till metadatastandarder för dokument på Internet. Mångfalden visar att det finns ett stort behov av förbättrade beskrivningar av nätets resurser. Den rika floran har dock i sig hittills utgjort ett hot mot den praktiska användningen av databeskrivningar. Splittringen har lett till att det har saknats stöd både för produktion av metadata i publiceringsredskapen och för användning av dem i sökrederkapen. Det är inte *flera* standarder vi behöver utan *en* standard som majoriteten kan ansluta sig till.

De två vanligaste metoderna för att föga metadata till webbsidor är idag HTMLs inbyggda *META*-tagg och biblioteksvärldens initiativ *Dublin Core* (DC). Vi börjar med en kort beskrivning av dessa metoder och en diskussion av problemen med dem. Vi går sedan över till att beskriva det som ser ut att kunna bli den framtida samordnande standarden för metadata, nämligen RDF - Resource Description Framework.

[\(Åter till början av artikeln\)](#)

## 4. HTML META

I standardkodningen av webbdokument finns ett grundläggande stöd för metainformation i dokumentets sidhuvud. Metainformationen utgörs av två olika HTML-taggar – *TITLE* och *META* – som beskriver dokumentets titel, innehåll, upphovsman och några nyckelord. Dessa metadata presenteras inte på själva webbsidan, utan är enbart avsedda att vara tillgängliga för sökning. Söktjänster som stödjer dessa metadata använder dem för att förbättra sökresultat och presentation i träfflistor. Exempel på sådana tjänster är AltaVista, Evreka, HotBot och InfoSeek.

*TITLE* är den enda metainformation som är "obligatorisk" enligt HTML-standard. Övrig metainformation läggs till med hjälp av HTML-taggen *META*, som används för att ange både författare, beskrivning och nyckelord. En beskrivning av en webbsida skulle med HTML *META* kunna se ut på följande vis:

```
<HEAD>
  <TITLE>Metadata för medicinskt bruk</TITLE>
  <META NAME="author" CONTENT="John Parnefjord, Ulf Kronman, Elisabet Frigell">
  <META NAME="description" CONTENT="Metadata för medicinskt bruk">
  <META NAME="keywords" CONTENT="Metadata, Resource Description Framework,
RDF">
</HEAD>\*
```

Av exemplet framgår att HTML *META* har begränsade möjligheter till att beskriva ett dokument uttömmande. Här saknas till exempel uppgifter om utgivare, upphovsrätt och publiceringsdatum. Standarden säger heller inget om hur innehållet i taggarna ska vara utformat. En undersökning gjord av tidskriften Nature [[Nature](#)] visar att en tredjedel av dokumenten på webben har metadata i form av HTML *META*. I undersökningen återfanns dock 123 olika varianter på HTML *META*, varav flertalet inte användes för att beskriva själva innehållet på sidorna, utan för interna administrativa ändamål.

Problemen med HTML *META* kan således sammanfattas med:

- För yttlig beskrivning av dokumenten
- Icke-standardiserad utformning av innehållet
- Avsaknad av utbyggbarhet

Till försvar för HTML *META* kan dock anföras att det är en fungerande och väl spridd standard som stöds av många söktjänster.

[\(Åter till början av artikeln\)](#)

## 5. Dublin Core

För att avhjälpa bristerna i HTML *META* togs år 1995 ett initiativ för att förbättra informationsåtervinningen av nätverksbaserade resurser, med särskilt fokus på webben. Strävan var att skapa samförstånd kring ett generellt metadata-system som präglas av enkelhet. Systemet kom sedermera att få namnet *Dublin Core*, ofta förkortat DC [[DC](#)]. Dublin Core använder 15 standardiserade metadataelement för att beskriva de centrala aspekterna hos resurser på webben. Elementens betydelse är

avsiktligt brett dennerade för att kunna tillgodose olika intressegruppers behov. Tabellen nedan visar elementen indelade i kategorier utifrån vilka aspekter de avser att beskriva. De tre kategorierna är innehåll, upphovsrätt samt manifestation.

Innehåll	Upphovsrätt	Manifestation
Title	Creator	Date
Subject	Publisher	Format
Description	Contributor	Identifier
Source	Rights	Type
Language		
Relation		
Coverage		

När man beskriver en webbsida väljer man fritt bland de olika elementen. Det är också tillåtet att upprepa samma element flera gånger i beskrivningen. Användningen av Dublin Core i HTML-sidor bygger på en utvidgning av HTMLs META-standard. Genom att använda attributen NAME och CONTENT till HTMLs META-taggar specificerar man olika element.

En beskrivning av ovan nämnda webbsida skulle enligt Dublin Core kunna se ut på följande vis:

```
<HEAD>
<TITLE>Metadata för medicinskt bruk</TITLE>
<META NAME="DC.Title" CONTENT="Metadata för medicinskt bruk">
<META NAME="DC.Subject" CONTENT="Metadata, Resource Description
Framework, RDF">
<META NAME="DC.Description" CONTENT="En rapport om användningen av
Resource Description Framework för användning av metadata för
nätverksbaserade resurser">
<META NAME="DC.Creator" CONTENT="Frigell, Elisabet">
<META NAME="DC.Creator" CONTENT="Parnefjord, John">
<META NAME="DC.Creator" CONTENT="Kronman, Ulf">
</HEAD>
```

För att möjliggöra fördjupade beskrivningar erbjuder Dublin Core även påbyggnad av underordnade element, så kallade *kvalifierare*. Kvalifierare syftar till att ytterligare precisera eller specificera innehållet i ett DC-element. Vid angivande av upphovsman kan det till exempel vara intressant att skilja mellan författare, illustratör och fotograf. Användning av kvalifierare i Dublin Core kan exempelvis se ut på följande sätt:

```
<META NAME="DC.Creator.Author" CONTENT="Frigell, Elisabet">
```

Dublin Core löser en del av problemen med HTML META, men skapar också ett antal nya problem. Det största problemet är Dublin Cores brist på utbyggbarhet i kombination med dess tolkningsfrihet, som har lett till att olika organisationer har använt det på helt olika sätt. Dublin Core framställs som ett väl etablerat metadatasystem, representerat på 21 språk, men Natures undersökning visar att endast 0,3% av sidorna på 2500 undersökta webbservrar använde Dublin Core [[Nature](#)].

De stora söktjänsterna på nätet stödjer inte heller metadata enligt Dublin Core, varför den som vill få sina sidor väl synliga måste lägga till metadata enligt HTML META. I vissa fall verkar det vara så illa att användningen av Dublin Core tillsammans med HTML META av vissa söktjänster uppfattas som så kallad "keyword spamming" – dubblering av sökord för att hamna högt i träfflistorna. Detta leder istället till att sidorna rensas ut ur sökindexen.

Problemen med Dublin Core kan sammanfattas med:

- För generellt för många användningsområden
- Ingen standardiserad utbyggbarhet
- Ingen standardiserad syntax för kvalifiering
- Liten spridning
- Brist på stöd i söktjänster

## 6. RDF – Resource Description Framework

Vi ser alltså att det finns problem med de standarder för metadata som idag används på Internet: HTML META är relativt spritt men har stora begränsningar i sin beskrivning av dokumenten. Dublin Core är mycket mer detaljerat, men dåligt spritt och har svårt att vinna acceptans från de som har specialiserade behov med sin metadatahantering. För att råda bot på detta initierade webbkonsortiet W3C ett arbete för att ta fram en ny standard för metadata – *Resource Description Framework*.

Till grund för sitt arbete med RDF använde W3C webbindustrins tidiga metadatasystem Platform for Internet Content Selection [[PICS](#)] och Meta Content Framework [[MCF](#)]. Andra modeller som legat till grund för arbetet är Extensible Markup Language [[XML](#)] och Dublin Core [[DC](#)].

RDF är, precis som namnet anger, inte en metadatastandard i sig, utan ett utbyggbart *ramverk* för metadatastandarder. Inom detta ramverk kan man använda olika standarder för metadata som i RDF kallas för *scheman*. En finess med RDF är att man också kan använda flera olika metadatasystem i samma beskrivning, något som gör att RDF blir utbyggbart och därigenom anpassningsbart för olika behov. En av grundtankarna med RDF är att möjliggöra automatiserad behandling av metadata. Med tanke på webbens omfattning och webbsidors föränderliga natur är det orimligt att insamling och indexering ska ske manuellt. För att RDF-beskrivningar ska kunna tolkas maskinellt krävs mycket tydliga standarder. RDF definierar standarder på tre nivåer:

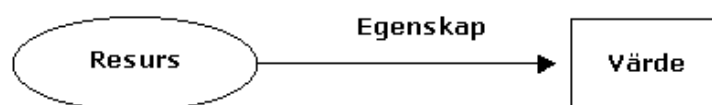
- **Struktur** – RDF *datamodell*
- **Syntax** – RDF *syntax*, uttryckt i Extensible Markup Language (XML)
- **Semantik** – RDF *schema*, till exempel Dublin Core

RDFs datamodell anger hur resurser struktureras och hur relationer mellan dessa kan beskrivas. För själva kodningen av metadata används RDF-syntax, som skrivs med hjälp av Extensible Markup Language (XML). Slutligen krävs en semantisk enhetlighet, vilket åstadkoms med hjälp av så kallade scheman. Dublin Core är ett bra exempel på ett sådant schema [[DC](#)].

De tre följande avsnitten kommer vi att ägna åt en grundläggande genomgång av RDFs tre byggstenar: datamodellen, syntaxen och schemat.

## 7. RDFs datamodell

Datamodellen i RDF utgår ifrån att det finns en *resurs* som är unikt identifierbar. Denna resurs har ett antal *egenskaper*. Var och en av dessa egenskaper har ett *värde*. Ett vanligt sätt att åskådliggöra RDFs resurser, egenskaper och värden visas i figur 1.



Figur 1. Grundläggande datamodell för RDF

### **Resurser**

En resurs kan vara en samling webbsidor, en enstaka webbsida eller en del av en webbsida. Även andra typer av objekt, som en tryckt bok eller tidskrift, kan vara en resurs. I RDF utgörs en resurs av allt som kan tillskrivas en unik identitet, en så kallad *Uniform Resource Identifier*, förkortat URI. På webben är URI:n liktydiga med resursens adress på Internet, den så kallade URL:en – *Uniform Resource Locator*.

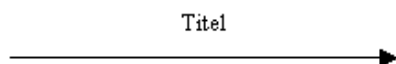
En URL består av tre delar:

1. Den metod som används för att nå resursen
2. Namnet på den dator där resursen finns
3. Namnet på resursen inklusive eventuell sökväg

Webbadressen `http://www.kib.ki.se/edu/index_se.html` är ett exempel på en URL. I detta exempel deklarerar vi att *HTTP-protokollet* ska användas för åtkomst. Datafilen som efterfrågas finns tillgänglig på en dator som har namnet `www.kib.ki.se`. Filen `index_se.html` är tillgänglig på denna dator via sökvägen `/edu/`.

### Egenskaper

Egenskaper utgör en aspekt, ett attribut eller en relation som kännetecknar en resurs. Varje egenskap är bärare av en specifik innebörd. Det går att kontrollera en egenskaps tillåtna värden, vilken typ av resurser egenskapen får beskriva samt vilka relationer den får ha till andra egenskaper.



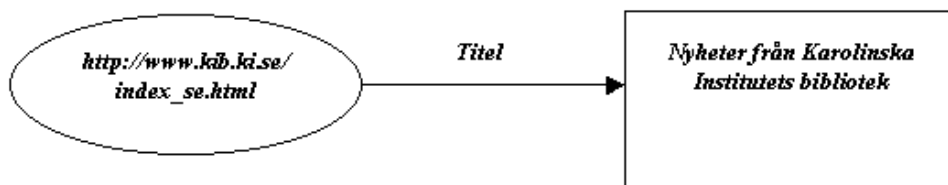
Figur 2. Exempel på egenskapen Titel

### Värde

Slutligen krävs att egenskapen har ett värde. I exemplet med webbsidan är värdet för egenskapen titel *Kurser vid Karolinska Institutets Bibliotek*.

### Uttryck

De tre komponenterna, resurs, egenskap och värde utgör i RDF tillsammans ett *uttryck*. I figur 3 förenas de ovanstående exemplen på resurs, egenskap och värde till ett uttryck.



Figur 3. Ett RDF-uttryck består av tre komponenter; resurs, egenskap och värde

[\(Åter till början av artikeln\)](#)

## 8. RDF syntax

För att uttrycka RDF används XML – *Extensible Markup Language*. RDF är en av de första praktiska tillämpningarna av XML.

XML är liksom RDF utvecklat av W3C och kan beskrivas som en kondenserad version av det mycket generella SGML – *Standard Generalised Markup Language*. Utvecklarna har tagit fasta på de vanligaste komponenterna i SGML och anpassat XML till hypertextmiljö. XML beskriver endast innehållet och strukturen i ett dokument, men säger inget om presentationen - dess utseende. Det innebär att innehållet i ett XML-dokument kan presenteras på många olika sätt utan att själva XML-koden behöver ändras. I korrekt XML ska en inledande `<tagg>` alltid avslutas med motsvarande `</tagg>`. Denna syntax med inledande och avslutande taggar benämns för *seriell* syntax och ser ut på följande vis:

```
<upphovsman>
  <efternamn>Parnefjord</efternamn>
</upphovsman>
```

Datamodellen i RDF uttrycks enklast med XMLs seriella syntax. Förutom den seriella XML-syntaxen kan

man använda så kallad *förkortad* XML-syntax. I den förkortade varianten av XML använder man enbart en inledande <tagg/>. Det avslutande / påvisar att taggen avslutas. Exemplet ovan motsvaras av följande förkortade syntax:

```
<upphovsman efternamn="Parnefjord"/>
```

Förkortad syntax är lämplig för att infoga RDF-beskrivningar i HTML-sidor, för att därigenom undvika att RDF-data presenteras i webbläsarens fönster. Vi kommer dock i fortsättningen huvudsakligen att använda seriell syntax i exemplen.

Genom att använda sig av XML kan RDF utnyttja den viktiga funktionen *XML namespace*. Det är med hjälp av XML namespace RDF kan begagna sig av kontrollerade *scheman*, som till exempel klassifikationssystem, thesauri eller ordlistor. I nästa avsnitt redogör vi för RDF-scheman.

[\(Åter till början av artikeln\)](#)

## 9. RDF schema

I vardagligt språk använder vi överenskomna ord för att uttrycka en särskild innebörd. Utan en tydlig definition kan innebörden av orden bli föremål för olika tolkningar, i synnerhet på en global arena som webben. Antag att vi har en RDF-beskrivning som lyder:

```
<?xml version="1.0"?>
<rdf:RDF>
  <rdf:Description
    about="http://www.kib.ki.se/kib/project/
    rdf/index.html">
    <title> Metadata för medicinskt bruk </title>
  </rdf:Description>
</rdf:RDF>
```

Vi kommer längre fram i artikeln att detaljerat beskriva den XML-kod som ingår i exemplet. För tillfället vill vi dock fokusera på raden med <title>, som innehåller själva värdet i metadatabeskrivningen.

Här framgår inte vad som egentligen avses med *title*. Innebörden av egenskapen title kan variera mellan olika användare och intressegrupper. Det krävs någon form av definition av elementet, ett tydliggörande av semantiken. Därför krävs en tydlig definition av de egenskaper och termer som används för att beskriva resurser.

I RDF införs semantik med hjälp av hänvisningar till scheman. Genom att till exempel specificera att title används i enlighet med Dublin Cores definition av title, klargör man dess semantik. RDF scheman tar fasta på de egenskaper som ett metadatasystem omfattar, deras begränsningar i användning samt en definition av respektive egenskap.

För att skilja mellan element hämtade från olika scheman föregås de i RDF av en *identifierare*. Till exempel kan egenskaper hämtade från Dublin Core inledas med identifieraren *dc:*. Identifierare kan väljas fritt, men *dc* är den rekommenderade identifieraren för Dublin Core. Då XML skiljer mellan gemener och versaler används *dc* genomgående, inte *DC* eller *Dc*. Likaså rekommenderas användning av gemener för Dublin Cores element. Därför ska till exempel *title* användas, ej *Title*.

För att göra exemplet ovan mer fullständigt ska syntaxen alltså vara:

```
<?xml version="1.0"?>
<rdf:RDF>
  <rdf:Description
    about="http://www.kib.ki.se/kib/project/
    rdf/index.html">
    <dc:title> Metadata för medicinskt bruk
  </dc:title>
  </rdf:Description>
</rdf:RDF>
```

Det är dock en komponent som saknas. Att förkortningen dc ska associeras med Dublin Core är inte allom



bekant. För att användare ska kunna läsa om schemats element och program ska kunna analysera dem krävs en URI som pekar ut det aktuella schemat. I RDF associeras prefixet *dc* med schemat Dublin Core genom att göra ett tillägg där förkortningen *dc* hänvisar till en URL, enligt följande:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description
    about="http://www.kib.ki.se/kib/project/
    rdf/index.html">
    <dc:title> Metadata för medicinskt bruk
    </dc:title>
  </rdf:Description>
</rdf:RDF>
```

På samma sätt associeras i alla RDF-beskrivningar identifikatorn *rdf* med ett schema för själva RDF-syntaxen. Man kan alltså säga att RDF använder sin egen schemametod för att beskriva sig själv. Det ser ut på följande vis:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description
    about="http://www.kib.ki.se/kib/project/
    rdf/index.html">
    <dc:title> Metadata för medicinskt bruk
    </dc:title>
  </rdf:Description>
</rdf:RDF>
```

Dublin Core och RDF är naturligtvis inte de enda scheman som kan användas. Vid beskrivning av en och samma resurs kan element från olika scheman användas genom att man pekar ut dem med XML namespace på ovanstående vis.

Funktionen med det avslutande #-tecknet i länken till schemat för RDF är tyvärr något oklart för närvarande. RDF-specifikationen [RDF-SPEC] lämnar ingen förklaring till användningen av detta tecken vid utpekande av RDF-schemat. Det verkar som om möjligheten att lägga till ett #-tecken står till buds för den som vill skapa pekare till enskilda element i sitt schema. Länken till dessa kan då konstrueras genom att man lägger till elementets namn efter #-tecknet:

```
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag"
```

Det förefaller inte som om användningen av # överensstämmer med övrig användning av namespace-funktionen inom XML [XML-NAME]. Detta kan alltså komma att se annorlunda ut när RDF verkligen

börjar användas i framtiden.

Vi går inte här in på hur man konstruerar själva schemat, då metoden för detta inte är helt fastställd ännu. Den blivande specifikationen för hur ett schema ska vara uppbyggt är ännu bara en "Proposed Recommendation" från W3C [RDF-SCHEMA], och fortfarande föremål för diskussioner (februari 2000). Vi kan ändå skaffa oss en uppfattning om hur det senaste schemat för Dublin Core är uppbyggt genom att följa DC-länken i våra RDF-exempel; <http://purl.org/dc/elements/1.1/>. Vi ser då att schemat för Dublin Core helt enkelt är en HTML-fil med textbeskrivningar av de olika elementen. Ett kort utdrag ur DC-schemat ser ut på följande vis:

**Element : Creator**

Name: Creator

Identifier: Creator

Definition: An entity primarily responsible for making the content of the resource.

Comment: Examples of a Creator include a person, an organisation, or a service. Typically, the name of a Creator should be used to indicate



... should be used to indicate the entity.

Följer vi istället länken till W3Cs schema för själva RDF; <http://www.w3.org/1999/02/22-rdf-syntax-ns#>, hittar vi ett XML-dokument, mer lämpat att läsas av program än människor. Början av RDF-schemat ser för närvarande ut på följande vis:

```
<?xml version="1.0"?>
<RDF
  xmlns="http://www.w3.org/1999/02/22-rdf-syntax-
  ns#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
  ns#"
  xmlns:s="http://www.w3.org/TR/WD-rdf-schema#">
<!--
  This is the RDF Schema for the RDF data model as
  described in the Resource Description Framework
  (RDF) Model and Syntax Specification
  http://www.w3.org/TR/REC-rdf-syntax -->

  <s:Class rdf:ID="Statement"
    s:comment="A triple consisting of a predicate, a
    subject, and an object." />

  <s:Class rdf:ID="Property"
    s:comment="A name of a property, defining specific
    meaning for the property" />

  <s:Class rdf:ID="Bag"
    s:comment="An unordered collection" />
```

Av skillnaderna i dessa exempel framgår alltså att det verkar finnas en hel del kvar att göra innan man kan påstå att RDF schema har fått någon enhetlig utformning.

[\(Åter till början av artikeln\)](#)

## 10. Dublin Core i RDF

Som vi tidigare nämnt är RDF ingen metadatastandard i sig utan ett ramverk för metadata. Det innebär att en RDF-beskrivning måste fyllas med metadata enligt ett vedertaget system – ett *schema* – för att beskrivningen ska bli meningsfull. Dublin Core har under hela utvecklingsarbetet framhållits som ett av de första standardiserade metadataschemata för att fylla RDF med innehåll. DC lämpar sig därför synnerligen väl att användas i kombination med RDF-datamodell och XML-syntax. RDF ska alltså inte betraktas som vare sig konkurrent eller ersättning till Dublin Core, utan som en generell utbyggnad som både behöver Dublin Core för sin användning och ger möjligheter att komplettera Dublin Core med andra schemata.

I följande avsnitt kommer vi att ge ett praktiskt exempel på en metadata-beskrivning utifrån den föregående teoretiska genomgången. Vi kommer att använda just Dublin Core till RDF-beskrivningen. Låt oss alltså börja från början, nu med fokus på det praktiska.

[\(Åter till början av artikeln\)](#)

## 11. Ramverk

Det första som måste klargöras i en RDF-beskrivning är att vi använder oss av XML-syntax. Man kan säga att en RDF-beskrivning fungerar som en slags XML-etikett på en resurs. Vi deklarerar användningen av XML genom att överst skriva:

```
<?xml version="1.0"?>
```

Att deklarera vilken version av XML som används är viktigt för de program som ska läsa filen. Med denna information kan program avgöra hur beskrivningen ska behandlas. Observera att den inledande deklARATIONEN inte är XML i sig utan enbart påvisar att dokumentet härifrån använder XML-syntax. Därmed behövs ingen avslutande tagg för denna deklARATION. Nästa moment är att visa att vi använder

RDF. Vi lägger därtill:

```
<?xml version="1.0"?>
<rdf>
```

Eftersom vi nu använder XML måste också en avslutande tagg tillfogas, alltså ska beskrivningen lyda:

```
<?xml version="1.0"?>
<rdf>
</rdf>
```

Inom dessa omslutande taggar kommer sedan hela RDF-beskrivningen att återfinnas. För att kunna verifiera att vi använder RDF korrekt krävs nu ett schema som anger vilka element som är tillåtna. Detta schema anropas med hjälp av XMLs namespace-funktion, enligt nedanstående modell:

```
<?xml version="1.0"?>
<rdf xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
</rdf>
```

Schemat för RDF är nu associerat med identifieraren rdf. Därmed kan de element som ingår i schemat användas. I schemat för RDF finns ett element som benämns just *RDF*. Därmed kan följande tillägg göras:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
  syntax-ns#">
</rdf:RDF>
```

Observera att vi nu har använt RDFs schemafunktion för att peka på ett schema för RDFs datamodell. Vi kan därigenom verifiera att vår RDF-beskrivning är korrekt. I vårt exempel kommer vi att beskriva en resurs med hjälp av schemat Dublin Core. Vi måste därför även peka ut Dublin Core-schemat. För detta använder vi återigen XMLs namespace-funktion:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
  syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
</rdf:RDF>
```

I detta fall associeras Dublin Core-schemat med förkortningen dc, som kommer att användas vid beskrivningen av de olika egenskaperna av resursen. Hittills har vi inte skapat någon beskrivning av resursen, vi har enbart förberett en ram för beskrivningen.

Vi kan nu börja med själva beskrivningen. På något sätt måste vi visa att beskrivningen börjar. Notera att XML åter kräver en avslutande tagg, varför exemplet kommer att se ut enligt följande:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
  syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description>
  </rdf:Description>
</rdf:RDF>
```

Inom den inledande <rdf:Description>-taggen och den avslutande </rdf:Description>-taggen kommer hela beskrivningen av resursens egenskaper att återfinnas. Men vad är det som ska beskrivas? Eftersom en RDF-beskrivning kan ligga helt separerad från en resurs är det nödvändigt att peka ut den resurs som RDF-koden ämnar beskriva. Detta gör man genom att ge Description-taggen ett *about*-attribut:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
  syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description
```

```
    about="http://www.kib.ki.se/kib/project/rdf/
    index.html">
</rdf:Description>
</rdf:RDF>
```

Förutom enskilda webbsidor har RDF även möjlighet att beskriva samlingar av dokument. Detta görs genom att man pekar ut en katalog med attributet *aboutEachPrefix*. Då kommer alla dokument i en katalog och katalogerna därunder ska betraktas som en gemensam resurs. För att göra detta skulle vi istället ha pekat ut katalogen på följande sätt:

```
aboutEachPrefix="http://www.kib.ki.se/kib/project/rdf/"
```

Vi har nu skapat ett komplett ramverk för vår RDF-beskrivning. Nästa steg är att fylla detta ramverk med metadata om resursen.

[\(Åter till början av artikeln\)](#)

## 12. Innehåll

Låt oss börja med att ange en titel för den aktuella resursen. I detta exempel är titeln på resursen *Metadata för medicinskt bruk*. För att påvisa att det är en titel vi avser beskriva infogar vi taggar för titeln. Observera att XML återigen kräver att det finns en avslutande tagg, varför vår åtgärd får följande konsekvens:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
  ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description
  about="http://www.kib.ki.se/kib/project/rdf/
  index.html">
  <title></title>
</rdf:Description>
</rdf:RDF>
```

Det finns nu en inledande och en avslutande tagg för titel. Ett känt problem infinner sig dock här. Det framgår inte vad som avses med titel i detta fall. Ambitionen är att beskriva resursen enligt Dublin Core, vars definition av titel faller väl inom ramen för denna beskrivning. På något sätt måste titel-taggen associeras med Dublin Core. Detta görs med hjälp av XML namespace mekanismen. Sedan tidigare är schemat för Dublin Core utpekade och associerat med förkortningen *dc*. Genom att använda *dc* som prefix till titel-taggen antyder vi att titel är ett element som är definierat enligt Dublin Core. Alltså bör beskrivningen bli:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
  ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description
  about="http://www.kib.ki.se/kib/project/rdf/
  index.html">
  <dc:title></dc:title>
</rdf:Description>
</rdf:RDF>
```

För att fullborda beskrivningen av titeln måste naturligtvis ett värde tilldelas, i detta fall *Metadata för medicinskt bruk*. Värdet placeras mellan *dc:title*-taggarna:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
  ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description
  about="http://www.kib.ki.se/kib/project/rdf/
```

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description
  about="http://www.kib.ki.se/kib/project/rdf/
index.html">
  <dc:title>Metadata för medicinskt bruk
  </dc:title>
</rdf:Description>
</rdf:RDF>

```

Naturligtvis finns det fler element som kan användas för att beskriva resursen, bland annat upphov. I Dublin

Core benämns detta element *creator*. Precis som för titel så ska upphov anges i enlighet med Dublin Cores definition. Med beskrivningen av titeln fortfarande i åminne är det inte vidare svårt att lista ut hur syntaxen ska se ut:

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description
  about="http://www.kib.ki.se/kib/project/rdf/
index.html">
  <dc:title>Metadata för medicinskt bruk</dc:title>
  <dc:creator>Frigell, Elisabet</dc:creator>
</rdf:Description>
</rdf:RDF>

```

Det finns dock flera författare till denna resurs. Att uttrycka detta är inga problem då element från Dublin Core kan användas ett obegränsat antal gånger och oberoende av ordning i en och samma beskrivning. Därmed är det bara att infoga ytterligare författare till beskrivningen enligt föregående exempel.

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description
  about="http://www.kib.ki.se/kib/project/rdf/
index.html">
  <dc:title>Metadata för medicinskt bruk
  </dc:title>
  <dc:creator>Frigell, Elisabet</dc:creator>
  <dc:creator>Parnefjord, John</dc:creator>
  <dc:creator>Kronman, Ulf</dc:creator>
</rdf:Description>
</rdf:RDF>

```

Andra typer av metadata som kan vara av intresse i en beskrivning och som omfattas av Dublin Core är ämnesord, utgivare, datum för skapande eller publicering, format, språk, typ av resurs samt en unik identitet för resursen. Genom att följa ovanstående modell kommer vårt exempel att se ut som beskrivningen nedan efter att vi infogat nämnda metadataelement:

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description
  about="http://www.kib.ki.se/kib/project/rdf/
index.html">
  <dc:title>Metadata för medicinskt bruk
  </dc:title>
  <dc:creator>Frigell, Elisabet</dc:creator>
  <dc:creator>Parnefjord, John</dc:creator>
  <dc:creator>Kronman, Ulf</dc:creator>
  <dc:subject> Metadata; Dublin Core; Resource Description
Framework; RDF </dc:subject>
  <dc:publisher> Karolinska Institutets Bibliotek </dc:publisher>

  <dc:date>1999-07-26</dc:date>
  <dc:format>text/html</dc:format>

```

```

    <dc:language>se-SE</dc:language>
    <dc:identifier>
    http://www.kib.ki.se/kib/project/rdf/
    </dc:identifier>
    <dc:type>text</dc:type>
</rdf:Description>
</rdf:RDF>

```

Vi har nu skrivit en kort RDF-beskrivning av vår rapport.

Den RDF-modell som vi visat ovan är ganska enkel till sin uppbyggnad och användbar i många sammanhang. Ibland kan man dock ha behov av särskilda beskrivningar som inte täcks av Dublin Core. Nästa avsnitt handlar om hur man kan lägga till egna scheman för dessa behov.

[\(Åter till början av artikeln\)](#)

## 13. Utbyggd RDF

I och med att RDF är utbyggbart öppnar sig till exempel möjligheten att beskriva medicinska resurser enligt *Medical Subject Headings* (MeSH). Att man vill använda MeSH betyder dock inte att man bör överge Dublin Core för den allmänna beskrivningen av resurserna. Den stora finessen med RDF är att man kan utnyttja ett schema så långt det räcker och sedan komplettera med ett mer ämnesspecifikt schema för sina speciella behov.

Vi kommer i detta avsnitt introducera ett fiktivt schema för MeSH i RDF-beskrivningen av vår rapport och föra den vidare mot en mer detaljerad beskrivning. Vi börjar med den allmängiltiga DC-beskrivningen av rapporten och lägger till en rad som introducerar ett nytt fiktivt MeSH-schema:

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
  ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:mesh="http://www.kib.ki.se/metadata/mesh/
  0.1/">
  <rdf:Description
    about="http://www.kib.ki.se/kib/project/rdf/
    index.html">
    <dc:title>Metadata för medicinskt bruk
    </dc:title>
    <dc:creator>Frigell, Elisabet</dc:creator>
    <dc:creator>Parnefjord, John</dc:creator>
    <dc:creator>Kronman, Ulf</dc:creator>
    <dc:subject>
      Metadata; Dublin Core; Resource Description Framework; RDF
    </dc:subject>
    <dc:publisher>
      Karolinska Institutets Bibliotek
    </dc:publisher>
    <dc:date>1999-07-26</dc:date>
    <dc:format>text/html</dc:format>
    <dc:language>se-SE</dc:language>
    <dc:identifier>
      http://www.kib.ki.se/kib/project/rdf/

    </dc:identifier>
    <dc:type>text</dc:type>
  </rdf:Description>
</rdf:RDF>

```

Vi kan nu använda oss av MeSH:s koder och textbeskrivningar för att ytterligare beskriva vår rapport med hjälp av till exempel `mesh:code` och `mesh:subject`:

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
  ns#"

```

```

    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:mesh="http://www.kib.ki.se/metadata/mesh/0.1/"
<rdf:Description
  about="http://www.kib.ki.se/kib/project/rdf/
index.html">
  <dc:title>Metadata för medicinskt bruk</dc:title>
  <dc:creator>Frigell, Elisabet</dc:creator>
  <dc:creator>Parnefjord, John</dc:creator>
  <dc:creator>Kronman, Ulf</dc:creator>
  <dc:subject>
    Metadata; Dublin Core; Resource Description Framework; RDF
  </dc:subject>
  <dc:publisher>
    Karolinska Institutets Bibliotek
  </dc:publisher>
  <dc:date>1999-07-26</dc:date>
  <dc:format>text/html</dc:format>
  <dc:language>se-SE</dc:language>
  <dc:identifier>
    http://www.kib.ki.se/kib/reports/metadata/rdf/
  </dc:identifier>
  <dc:type>text</dc:type>
  <mesh:code>L01.300.100</mesh:code>
  <mesh:subject>Abstracting and Indexing
  </mesh:subject>
</rdf:Description>
</rdf:RDF>

```

Fördelarna med att använda sig av ovanstående utbyggda modell istället för att skapa ett helt eget schema är stora. Om Dublin Core uttryckt som RDF blir en vedertagen standard som stöds av de stora söktjänsterna kommer denna utbyggda beskrivning att ge dem en meningsfull beskrivning av dokumentet, trots att vi har lagt till våra egna medicinspecifika metadata.

En söktjänst specialiserad på medicinska resurser kan nu specialindexera dokumentet med hjälp av MeSH. Det förutsätter naturligtvis att den medicinska söktjänsten stödjer RDF-beskrivningar uttryckta både med Dublin Core och MeSH.

[\(Åter till början av artikeln\)](#)

## 14. RDF i HTML

För att RDF ska bli praktiskt användbart på webben måste RDF-beskrivningen på något sätt kopplas till webbdokumentet den avser att beskriva. Det kan antingen göras genom att infoga RDF-koden i själva webbdokumentet, eller genom att länka en extern fristående RDF-beskrivning till webbdokumentet.

Att lägga RDF-beskrivningen i ett separat dokument är den enklaste modellen, eftersom man i detta fall kan använda XMLs seriella syntax för sin RDF-kod. Den kommer alltså att se ut som i vårt exempel ovan. I själva HTML-dokumentet lägger man sedan en länk till RDF-beskrivningen, som i princip kan finnas var som helst på nätet. Koden i HTML-dokumentets huvud kommer då att se ut på följande vis:

```

<HEAD>
  <TITLE>Metadata för medicinskt bruk</TITLE>
  <LINK REL="meta" SRC="http://www.kib.ki.se/kib/
project/rdf/index.rdf">
</HEAD>

```

Den utpekade filen index.rdf ska då innehålla en RDF-beskrivning uttryckt i XML. När ett program som kan utnyttja RDF läser HTML-dokumentet kommer det även att läsa in och tolka RDF-filen och använda innehållet för att presentera metainformation.

Om man istället vill infoga RDF-beskrivningen i själva HTML-dokumentet måste en särskild förkortad XML-syntax användas. Detta för att undvika att själva RDF-beskrivningen visas som en del av dokumentet i webbläsarens fönster. Ett komplett HTML-dokument med Dublin Core i RDF kan komma att se ut som nedan:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">
<HTML>
<HEAD>
<TITLE>Metadata för medicinskt bruk</TITLE>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description
rdf:about="http://www.kib.ki.se/kib/project/rdf/
index.html"
dc:title="Metadata för medicinskt bruk"
dc:creator="Frigell, Elisabet"
dc:creator="Parnefjord, John"
dc:creator="Kronman, Ulf"
dc:subject="Metadata; Dublin Core; Resource Description Framework;
RDF; MeSH"
dc:publisher="Karolinska Institutets bibliotek"
dc:date="1999-07-26"
dc:format="text/html"
dc:language="swe"
/>
</rdf:RDF>
</HEAD>
<BODY>
Här är själva webbdokumentet
</BODY>
</HTML>

```

[\(Åter till början av artikeln\)](#)

## 15. RDF i praktiken

Vad händer då med RDF i den praktiska vardagen på nätet idag (februari 2000)? Inte mycket, tyvärr. Just nu verkar det som om alla går och väntar på att specifikationen för RDF schema ska bli fastställd och att någon tar det första steget och skapar en tillämpning för att producera och presentera RDF-data.

Bland de fåtaliga exempel på praktiska tillämpningar av RDF vi har kunnat hitta finns en webbaserad tillämpning för generering av metadata i Dublin Core-format, som nu även kan producera Dublin Core-kod inbäddad i RDF [[DC-Dot](#)]. En av de mest intressanta användningarna av RDF hittar vi hos företaget Netscape, där man skapat ett register över webbsidor inom olika ämnen och metadataindexerat dem med en egen variant av RDF [[NS-ODP](#)]. Netscape arbetar också med att bygga in RDF-stöd i sin nya webbläsare Mozilla [[NS-MOZ](#)]. Den nuvarande situationen för metadata på Internet i allmänhet och RDF i synnerhet liknar den klassiska med hönan och ägget. Vilket ska komma först; ska metadata beskrivningar enligt RDF börja läggas in i dokument, trots att det inte finns något stöd för RDF bland indexeringsredskapen? Ska de stora webbindexen börja stödja RDF trots att det saknas dokument med RDF-beskrivningar?

Förutsättningarna för att RDF ska bli användbart kan sammanfattas i följande punkter:

- Det behövs stöd vid produktion av dokument, till exempel i form av editorer, konverterare och (halv-)automatiska indexerare.
- Det behövs stöd vid indexering och sökning av resurser genom att de stora söktjänsterna utnyttjar RDF vid insamling och presentation.
- En allmän användning av XML eller övergångsstandarden XHTML kommer troligen också att underlätta införandet av RDF [[XHTML](#)].

En möjlig lösning på problemet med vad som ska komma först är att de som utvecklar lokala sökfunktioner redan nu börjar använda metadata i RDF-format för sin egen indexering. Det innebär att RDF kommer att finnas i dokumenten den dag de stora söktjänsterna börjar stödja denna standard. I Sverige har vi goda möjligheter att förbereda oss för den kommande RDF-standarden genom att centrala söktjänster som till



exempel Svesök och SAFARI så snart som möjligt börjar använda sig av RDF för sina metadatabeskrivningar.

[\(Åter till början av artikeln\)](#)

## 16. Slutsatser

Resource Description Framework är ett utbyggbart ramverk där man samtidigt kan använda flera olika system för metadatasystem. RDF är så konstruerat att dessa metadatasystem kan utvecklas och användas utan central kontroll och koordinering. RDF har därför alla förutsättningar att kunna bli den framtida enhetliga standarden för metadatanhantering på Internet.

Metadatasystemet Dublin Core är ett av de centrala schemata för RDF. Utbyggbarheten hos RDF kommer sannolikt att säkra framtiden för Dublin Core, eftersom RDF tillåter att varje organisation lägger till sina egna system för specialiserade metadatasystem utan att förändra något av de 15 grundelementen i Dublin Core.

RDF framstår dock idag (februari 2000) som alltför ofärdigt och oprövat för att man ska kunna förorda en konkret användning för att underlätta informationssökning på Internet. Detta beror främst på att det ännu inte finns några sökredskap som kan utnyttja RDF. Det kommer troligen att ta ett till två år innan program för produktion och praktisk användning av RDF kommit ut på marknaden.

För den som idag avser att skapa egna metadatasystem är det dock rekommendabelt att så snart som möjligt börja använda sig av RDF.

[\(Åter till början av artikeln\)](#)

## Om författarna

Ulf Kronman började arbeta som systemerare vid Karolinska Institutets Bibliotek (KIB) 1987. Under åren 1995-97 var han webmaster vid Karolinska Institutet och ledde sedan institutets projekt för webbutveckling under ett år. Han är nu tillbaka på KIB och arbetar som webbutvecklare i bibliotekets nystartade webbyrå Vision. Ulf innehar examina som fysiker och dokumentalist och har bedrivit forskarstudier i musikkakustik.

John Parnefjord arbetar sedan 1998 vid Karolinska Institutets Bibliotek. Idag är han knuten till bibliotekets webbyrå Vision, med programmering, informationsarkitektur och användargränssnitt som främsta arbetsuppgifter. John innehar sedan tidigare en magisterexamen i biblioteks- och informationsvetenskap samt en kandidatexamen i socialantropologi.

Artikeln är en sammanfattning av rapporten *Metadatasystem för medicinskt bruk*, författad av Elisabet Frigell, John Parnefjord och Ulf Kronman. Rapporten har skrivits med stöd från BIBSAM inom ramen för Karolinska Institutets Biblioteks åtaganden som ansvarsbibliotek inom medicin och omvårdnad. Rapporten kommer att kunna hämtas från bibliotekets webbplats på adressen <http://www.kib.ki.se/kib/project/rdf/>.

[\(Åter till början av artikeln\)](#)

## Noter

\* OBS att detta exempel korrigerats i den digitala upplagan. [\(Åter till texten\)](#)

## Referenser

[DC] Dublin Core Metadata Initiative.  
<http://purl.org/dc/>

[DCDM] Dublin Core Metadata Initiative: Data Model Working Group.  
<http://purl.org/dc/groups/datamodel.htm>

[DCES] Dublin Core Metadata Element Set, Version 1.1: Reference Description.  
<http://purl.org/dc/elements/1.1/>

[DC-dot] DC-dot: Dublin Core generator.  
<http://www.ukoln.ac.uk/metadata/dcdot/>

[DC-HTML] Encoding Dublin Core Metadata in HTML. – Red. J. Kunze. - RFC 2731.  
<http://www.ietf.org/rfc/rfc2731.txt>

[DC-RDF] Guidance on expressing the Dublin Core within the Resource Description Framework (RDF). – Red. Eric Miller, Paul Miller och Dan Brickley.  
<http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/>

[DT-1] Tångring, Jan. - Med metadata är Internet en distribuerad databas // Datateknik 3.0. – Nr. 5, 1999. - S. 29.

[DT-2] Tångring, Jan. - Biblioteksfolket städar upp på nätet // Datateknik 3.0. – Nr 5, 1999. - S. 30 – 32.

[HTML] HyperText Markup Language: Home Page.  
<http://www.w3.org/MarkUp/>

[HTML-META] Meta data // HTML 4.01 Specification: W3C Recommendation 24 December 1999. – Red. Dave Raggett, Arnaud Le Hors och Ian Jacobs.  
<http://www.w3.org/TR/REC-html40/struct/global.html#h-7.4.4>

[IANNELLA] Iannella, Renato. - An Idiot's Guide to the Resource Description Framework.  
<http://archive.dstc.edu.au/RDU/reports/RDF-Idiot/>

[MCF] Guha, R.V. - Meta Content Framework.  
<http://www.xspace.net/hotsauce/mcf.html>

[Nature] Lawrence, Steve och Giles, C. Lee. - Accessibility of information on the web // Nature. – Vol. 400, nr 107, 1999. – S. 107-109.  
<http://www.nature.com/server-java/Propub/nature/400107A0.frameset?context=search>  
<http://www.nature.com/server-java/Propub/nature/400107A0.pdf>

[NS-MOZ] RDF in Mozilla.  
<http://rdf.netscape.com/rdf/doc/>  
<http://rdf.netscape.com/rdf/doc/api.html>

[NS-ODP] Open Directory Project.  
<http://dmoz.org/>

[PICS] Platform for Internet Content Selection (PICS).  
<http://www.w3.org/PICS/>

[RDF] Resource Description Framework (RDF).  
<http://www.w3.org/RDF/>

[RDF-ABBREVIATED] Basic Abbreviated Syntax// Resource Description Framework (RDF) Model and Syntax: W3C Recommendation 22 February 1999. – Red Ora Lassila och Ralph R. Swick.  
<http://www.w3.org/TR/REC-rdf-syntax#abbreviatedSyntax>

[RDF-SCHEMA] Resource Description Framework (RDF) Schema Specification: W3C Proposed Recommendation 03 March 1999. – Red Dan Brickley och R.V. Guha.  
<http://www.w3.org/TR/PR-rdf-schema/>

[RDF-SPEC] Resource Description Framework (RDF) Model and Syntax Specification: W3C Recommendation 22 February 1999. – Red Ora Lassila och Ralph R. Swick.  
<http://www.w3.org/TR/REC-rdf-syntax/>

[Reggie] Reggie: The Metadata Editor.  
<http://metadata.net/dstc/>

[RFC2396] Uniform Resource Identifiers (URI): Generic Syntax – Red. T. Berners-Lee, R. Fielding och L. Masinter. - RFC2396.  
<http://info.internet.isi.edu/in-notes/rfc/files/rfc2396.txt>

[SiRPAC] SiRPAC - Simple RDF Parser & Compiler.  
<http://www.w3.org/RDF/Implementations/SiRPAC/>

[vCard] Representing vCard v3.0 in RDF.  
<http://www.dstc.edu.au/RDU/RDF/draft-iannella-vcard-rdf-00.txt>

[W3C] W3C - World Wide Web Consortium.

<http://www.w3.org/>

[XHTML] XHTML 1.0: The Extensible HyperText Markup Language. A Reformulation of HTML 4 in XML 1.0. W3C Recommendation 26 January 2000.

<http://www.w3.org/TR/xhtml1/>

[XML] Extensible Markup Language (XML).

<http://www.w3.org/XML/>

[XML-NAME] Namespaces in XML: World Wide Web Consortium 14-January-1999. – Red. Tim Bray, Dave Hollander och Andrew Layman.

<http://www.w3.org/TR/REC-xml-names/>

[XML-SPEC] Extensible Markup Language (XML) 1.0: W3C Recommendation 10-February-1998. – Red. Tim Bray, Jean Paoli och C. M. Sperberg-McQueen.

<http://www.w3.org/TR/REC-xml>

([Åter](#) till början av artikeln)

---

© Ulf Kronman och John Parnefjord 1999

[Åter till Human IT 4/1999](#)