Markup, XML och namnrymder

BIBLIOTEKSHÖGSKOLAN
HÖGSKOLAN I BORÅS

Mikael Gunnarsson

Oktober, 2017

BHS/GU — Borås/Göteborg

HÖGSKOLAN I BORÅS

Agenda – till vilken avbrott för hands-on kommer

1. Vad karaktäriserar XML
2. Markup
3. XML notation och generell struktur
4. Tillämpningar – dokumentmodeller
5. ”XML is not the end”

Vad karaktäriserar XML

Det som är XML

- förverkligar någon form av **markup** (uppmärkning)
- implicerar en viss bestämd generell **data struktur**
- implicerar en viss **notation**
- förverkligar en viss **tillämpning** genom en abstraktion i termer av en eller flera konkreta datastrukturer, där strukturens element måste användas på ett visst förutbestämt sätt – i de flesta fall knutet till en s k **namnrymd**

Vissa tillämpningar/namnrymder förverkligas som XML, men kan också förverkligas med andra formalismer. T ex HTML, RDF och MARC21 ...
Vissa tillämpningar är bara XML, som XSL, och kan inte uttryckas på andra vis.

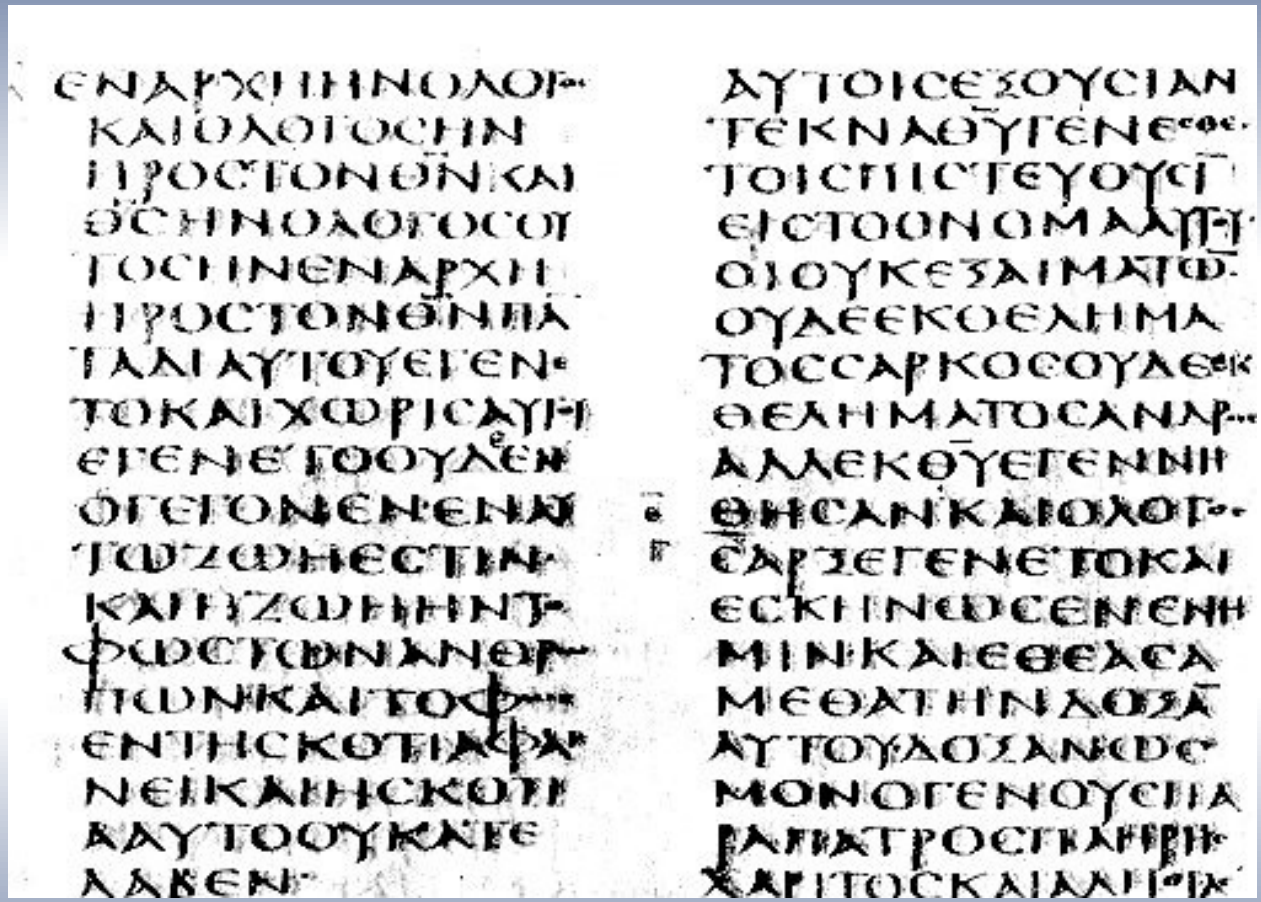
Markup - typologi

- Interpunktion
- Presentationsorienterad
- Procedurell
- Beskrivande

  \+ Referentiell markup; Metamarkup

Coombs, J. H., Renear, A. H., & DeRose, S. J. (1987). Markup systems and the future of scholarly text processing. *Communications of the ACM*, 30(11), 933–947.

Scriptio continua



Figur: Scriptio continua från Codex Sinaiticus, 300 AD

Interpunktion och “white space”

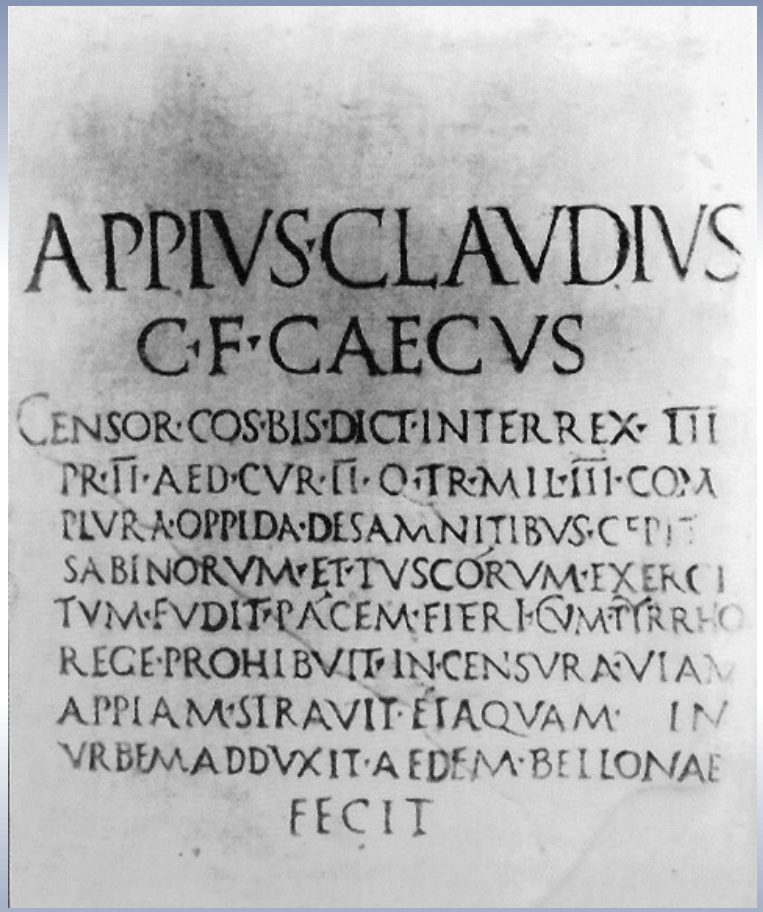
“Jag heter Mikael”	$\Longleftarrow$	«Je m’appelle Mikael»
Vad heter du?	$\Longleftarrow$	¿cómo te llamas?

Interpunktion och “white space”

“Jag heter Mikael”	$\Longleftarrow$	«Je m’appelle Mikael»
Vad heter du?	$\Longleftarrow$	¿cómo te llamas?

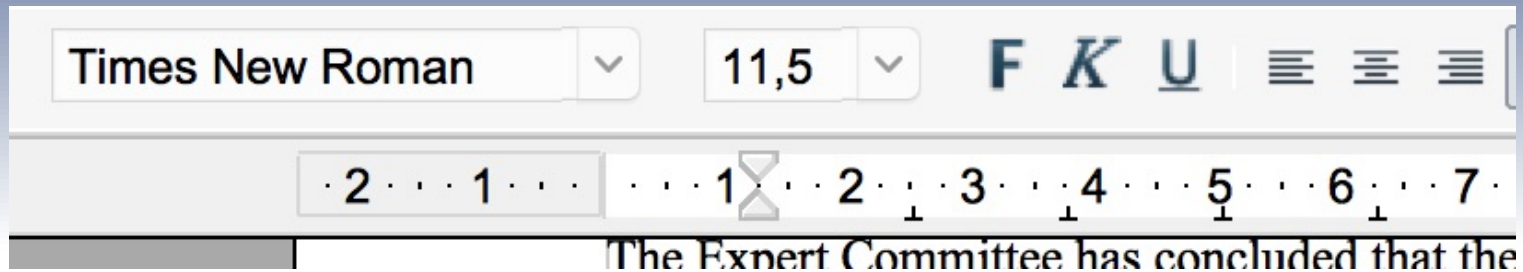
En sjuk syster	$\neq$	En sjuksyster
A web site	$\equiv$	A website

Tidiga tecken på interpunktion



Figur: Romersk inskription
notera punkterna och apex (den akuta accenten) för att markera lång vokal

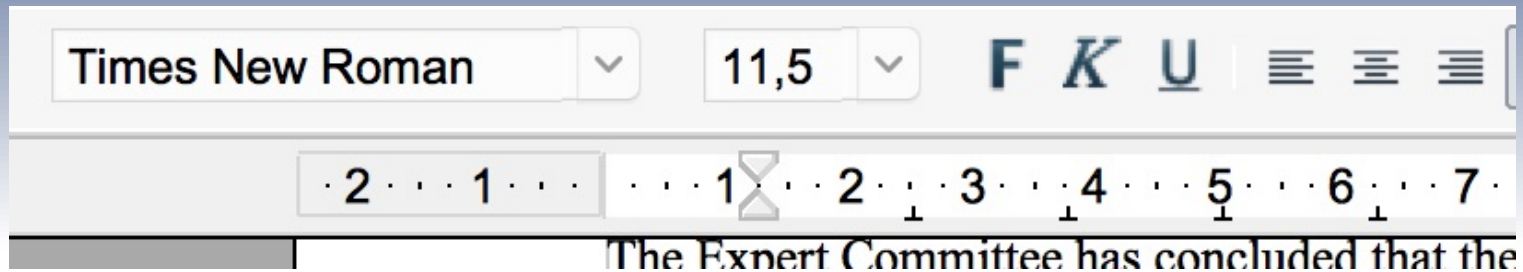
Presentationsorienterad markup



Figur: Funktionalitet för visuellt orienterad formatering i ordbehandlingsbehandling

Om typsnittet behöver ändras från Times New Roman till Verdana, storleksförhållandet mellan brödtext och rubriker behöver justeras, citationstecknens form behöver förändras . . . ???

Presentationsorienterad markup



Figur: Funktionalitet för visuellt orienterad formatering i ordbehandlingsbehandling

Om typsnittet behöver ändras från Times New Roman till Verdana, storleksförhållandet mellan brödtext och rubriker behöver justeras, citationstecknens form behöver förändras . . . ???

Visuell form är sällan otvetydig. Vi vet inte alltid **varför** ett ord, en fras eller ett stycke är kursiverat — har givits en viss form?

Procedurell markup

Distinktionen mellan presentationsorienterad och procedurell markup är inte skarp, men den senare kan definieras som **instruktioner** till typsättaren (mänsklig eller icke-mänsklig aktör) för hur text ska bearbetas.

Procedurell markup

Distinktionen mellan presentationsorienterad och procedurell markup är inte skarp, men den senare kan definieras som **instruktioner** till typsättaren (mänsklig eller icke-mänsklig aktör) för hur text ska bearbetas.

\begin{center} Markup, XML och namnrymder \\
Workshop för Digital Humaniora
\end{center}

Deskriptiv markup

```
\section{Markup, XML och
namnrymder}
\label{sec:greetings}
\par Lorem ipsum dolor sit amet,
consectetuer ...
    \par Nam dui ligula, fringilla a,
        euismod sodales ...
```

Figur: LaTeX markup

```
<h1>Markup, XML och
namrymder</h1>
<p>
Lorem ipsum dolor sit amet,
consectetuer ...
</p> <p> Nam dui ligula, fringilla a,
    euismod sodales ...</p>
```

Figur: HTML markup

Deskriptiv markup, forts.

- "Ren" beskrivande markup:
  Skribenten beskriver titel, kapitel, stycken, titelsidor, **rad- och sidbrytningar** mm i en existerande fysisk utgåva (ex: TEI)

Deskriptiv markup, forts.

- "Ren" beskrivande markup:
  Skribenten beskriver titel, kapitel, stycken, titelsidor, **rad- och sidbrytningar** mm i en existerande fysisk utgåva (ex: TEI)
- Performativ markup:
  Skribenten deklarerar titel, avsnitt, stycken, titelsidor, citat, ekvationer mm i en text i tillblivelse (ex: HTML)

I det första fallet är det meningsfullt att fråga om uppmärkningen är korrekt eller inte, i det andra fallet inte.

Sammanfattning

- Markup (oavsett typ) tjänar syftet att strukturera en mängd data genom att ge dess olika komponenter “etiketter” som indikerar vilken typ av data vi har att göra med — som avslöjar dess semantik

Sammanfattning

- Markup (oavsett typ) tjänar syftet att strukturera en mängd data genom att ge dess olika komponenter “etiketter” som indikerar vilken typ av data vi har att göra med — som avslöjar dess semantik
- Men sekventiell text är inte den enda typen av data vi har att göra med som behöver struktur . . .

HÖGSKOLAN I BORÅS

Komponenter i en xml-instans

Element

Alla element tillsammans utgör det samlade innehållet i ett dokument ...

```
<titel>Introduktion till XML  </titel>
<bild URI="logo.jpg"/>
     /.../
```

Komponenter i en xml-instans

Element

Alla element tillsammans utgör det samlade innehållet i ett dokument ...

```
<titel>Introduktion till XML  </titel>
<bild URI="logo.jpg"/>
     /.../
```

Element kan innehålla andra element och text (s k parsed character data": PCDATA)

HÖGSKOLAN I BORÅS

Komponenter i en xml-instans

Attribut

... tillsammans med attributens innehåll ...

```
<titel type="huvudtitel">Introduktion till XML</titel>
<bild URI="logo.jpg"/>
```

HÖGSKOLAN I BORÅS

Komponenter i en xml-instans

Attribut

... tillsammans med attributens innehåll ...

```
<titel type="huvudtitel">Introduktion till XML</titel>
<bild URI="logo.jpg"/>
```

Terminologi:

<titel>, </titel> och <bild/> kallas **märke** eller **tag(g)**. Märken tillsammans med vad som ges mellan par av märken utgör **element**.

Komponenter i en xml-instans

Entiteter

... och eventuella entitetsreferenser

```
&amp;   &gt;    &lt;
```

Komponenter i en xml-instans

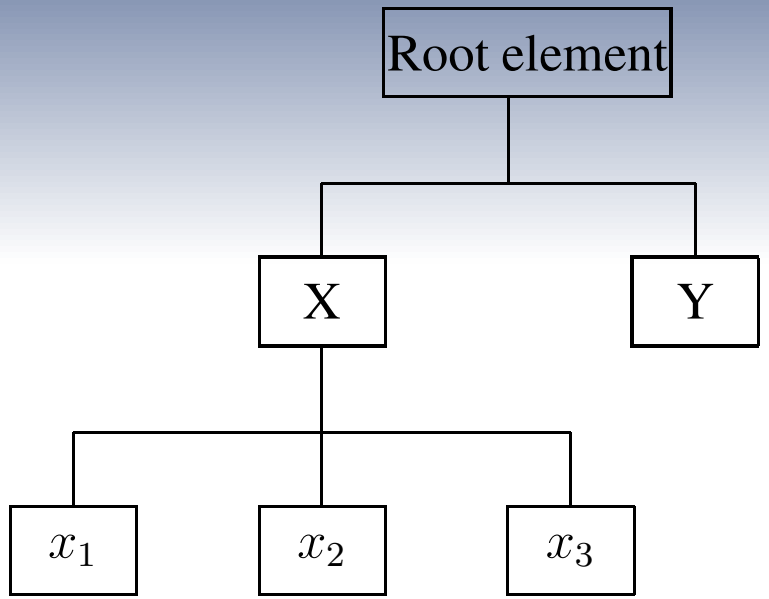
Entiteter

... och eventuella entitetsreferenser

```
&amp;   &gt;    &lt;
```

&, < och > har speciell innebörd och måste ges som entiteter

En resulterande struktur



1) x_1, x_2, x_3 is a (*child*) of X

2) X is a (*parent*) of x_1, x_2, x_3

3. x_1, x_2, x_3 are (*siblings*)

4) x_1 is a (*preceding-sibling*) of x_2

5) The root element is (*ancestor*) of x_1, x_2, x_3

6) x_1, x_2, x_3 are (*descendants*) of the root element and of X

Exempel - inköpslista

```
<?xml version="1.0" encoding="utf-8"?>
<groceryList>
 <product><quantity>1 l</quantity>
          <name>Milk</name></product>
 <product><name>Coffee</name></product>
 <product><name>Bread</name></product>
 <product><quantity>3 dl</quantity>
          <name>Cream</name></product>
 <product><quantity>1 package</quantity>
          <name>Yeast</name></product>
 <product><quantity>2 kg</quantity>
          <name>Flour</name></product>
</groceryList>
```

Naturliga vs artificella språk

- Ett **språk** kan definieras som en ändlig mängd meningsbärande **ord** (en vokabulär) som kan kombineras utifrån vissa bestämda **regler** (dess syntaktiska regler) för att forma meningsfulla satser som kan användas för påståenden eller för att åstadkomma effekt.

Naturliga vs artificella språk

- Ett **språk** kan definieras som en ändlig mängd meningsbärande **ord** (en vokabulär) som kan kombineras utifrån vissa bestämda **regler** (dess syntaktiska regler) för att forma meningsfulla satser som kan användas för påståenden eller för att åstadkomma effekt.
- Vi skiljer mellan **naturliga språk** och **artificiella språk**. Naturliga språk kan **beskrivas** i en lexiko-grammatik, medan artificiella språk vanligen **definieras** med hjälp av ett metaspråk. (programspråk är artificiella språk)

Naturliga vs artificella språk

- Ett **språk** kan definieras som en ändlig mängd meningsbärande **ord** (en vokabulär) som kan kombineras utifrån vissa bestämda **regler** (dess syntaktiska regler) för att forma meningsfulla satser som kan användas för påståenden eller för att åstadkomma effekt.
- Vi skiljer mellan **naturliga språk** och **artificiella språk**. Naturliga språk kan **beskrivas** i en lexiko-grammatik, medan artificiella språk vanligen **definieras** med hjälp av ett metaspråk. (programspråk är artificiella språk)
- HTML är ett artificiellt språk för publicering av text (naturligt språk) på webben, ursprungligen definierat med hjälp av SGML (ca 1989). TEI är ett artificiellt språk för bl a beskrivning av kulturhistoriskt värdefullt material i analog from, numera definierat med hjälp av XML. Såväl HTML som TEI förutsätter att varje instans följer en viss bestämd grammatik — en dokumentdefinition.

Metaspråk

- XML är ett **metaspråk** i den meningen att det är ett artificiellt språk konstruerat för att definiera nya (artificiella) språk — som då kallas för **XML-tillämpningar**. XML är **inte** ett programspråk.

Metaspråk

- XML är ett **metaspråk** i den meningen att det är ett artificiellt språk konstruerat för att definiera nya (artificiella) språk — som då kallas för **XML-tillämpningar**. XML är **inte** ett programspråk.
- En tillämpning definieras genom att fr a dess **element** och **attribut** definieras, och hur de tillåts förekomma. På så vis kan man också säga att en viss precis datastruktur, eller datatyp, definieras.

Metaspråk

- XML är ett **metaspråk** i den meningen att det är ett artificiellt språk konstruerat för att definiera nya (artificiella) språk — som då kallas för **XML-tillämpningar**. XML är **inte** ett programspråk.
- En tillämpning definieras genom att fr a dess **element** och **attribut** definieras, och hur de tillåts förekomma. På så vis kan man också säga att en viss precis datastruktur, eller datatyp, definieras.
- En tillämpning kan definieras med flera ”formalismer” (se nästa slide). Därmed skapas **regler** som är **tvingande**.

Formalismer för XML-kontroll

- Document Type Definition (DTD)
- XML Schema
- Relax NG (https://en.wikipedia.org/wiki/RELAX_NG)
- Schematron

Dokumenttypsdefinitioner

- En dokumenttypsdefinition kan definieras som en preskriptiv grammatik för hur en XML-instans tillåts vara strukturerad — hur dess delar skall forma en trädstruktur med förutsägbarhet för varje gren. (Jmf. den relativt förutsägbara strukturen i en vetenskaplig tidskriftsartikel)

Dokumenttypsdefinitioner

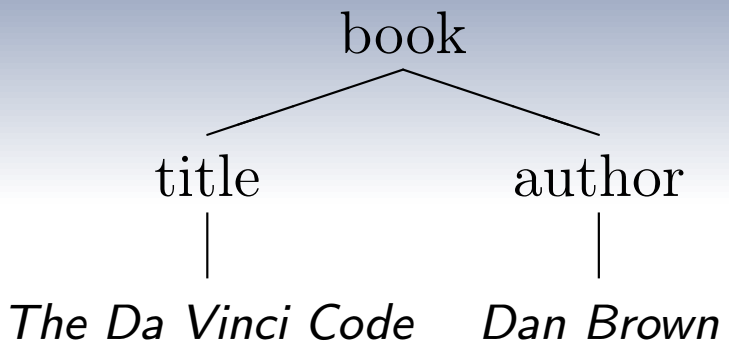
- En dokumenttypsdefinition kan definieras som en preskriptiv grammatik för hur en XML-instans tillåts vara strukturerad — hur dess delar skall forma en trädstruktur med förutsägbarhet för varje gren. (Jmf. den relativt förutsägbara strukturen i en vetenskaplig tidskriftsartikel)
- Vissa dokumenttyper är mer strikta än andra. (En privat hemsida uppvisar sannolikt minimal förutsägbarhet m a p strukturen. En inkomstdeklaration måste tvärtom vara extremt förutsägbar.)

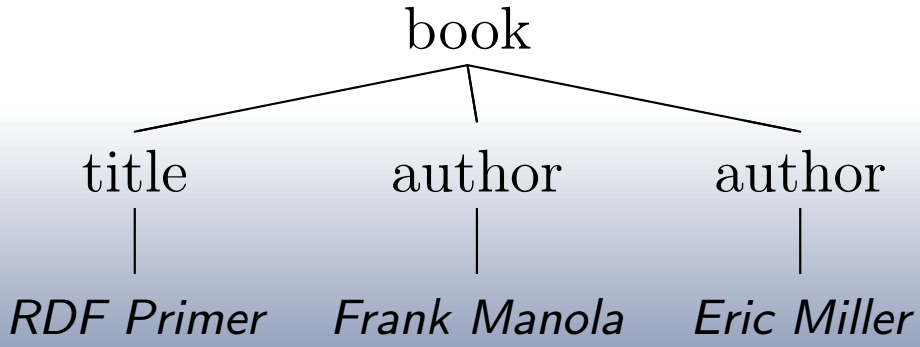
Dokumenttypsdefinitioner

- En dokumenttypsdefinition kan definieras som en preskriptiv grammatik för hur en XML-instans tillåts vara strukturerad — hur dess delar skall forma en trädstruktur med förutsägbarhet för varje gren. (Jmf. den relativt förutsägbara strukturen i en vetenskaplig tidskriftsartikel)
- Vissa dokumenttyper är mer strikta än andra. (En privat hemsida uppvisar sannolikt minimal förutsägbarhet m a p strukturen. En inkomstdeklaration måste tvärtom vara extremt förutsägbar.)
- Varje datastruktur (givet att den är meningsfull att generalisera) kan formuleras som en dokumenttyp.
- Kreatören måste bl a bestämma
  - vilka element som är tillåtna,
  - hur de skall benämnas,
  - och i vilken ordning de får förekomma

DTDer

```
<!ELEMENT book (title,author+)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
```





Några olika innehållsmodeller

Ordningen mellan element är viktig

```
<!ELEMENT book (author,year,title)>
```

Ordningen mellan element är valfri

```
<!ELEMENT book (author|year|title)*>
```

Något element kan saknas

```
<!ELEMENT book (author,year?,title)>
```

Något element kan förekomma flera gånger

```
<!ELEMENT book (author+,year,title)>
```

Några olika innehållsmodeller

Något element kan saknas eller förekomma flera gånger

```
<!ELEMENT book (author*,year,title)>
```

Elementet som definieras kan ha ett ”blandat innehåll”

```
<!ELEMENT book (author|year|title|#PCDATA)*>
```

Attribut:

Elementet `author` kan ta ett attribut `role`

```
<!ATTLIST author role CDATA #IMPLIED>
```

Några olika innehållsmodeller

Elementet `author` måste ges ett attribut `role`

```
<!ATTLIST author role CDATA #REQUIRED>
```

Attributet `role` kan endast ta ett av flera möjliga värden, varav `auth.` är default

```
<!ATTLIST author role (red.|auth.|övers.) "auth.">
```

Namnrymder

- Om flera ”språk” skall användas samtidigt måste det finnas otvetydigaindikationer på vilket språk som avses vid en viss bestämd punkt
- En namnrymd är enkelt uttryckt en särskild XML-tillämpning med ett särskilt namn och identifierare som anropas med ett särskilt **namnrymdsprefix**.

```
<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://libris.kb.se/resource/bib/3095664">
    <dc:creator
       rdf:resource="http://libris.kb.se/resource/auth/94541"/>
</rdf:Description>
      /.../
```

XML is not the end

- Förutom XHTML bör alla tillämpningar av XML för statisk data ses som antingen arkivkälla eller utbytesformat. Vanligen omvandlas källorna för presentation eller annan bearbetning.
- **XSL** (**Extensible Stylesheet Language**): XPath, XSL Transformations, XSL Formatting Objects
- XQuery ...