

XML — Revolution eller hybris? Tillämpningar av XML inom det digitala biblioteksväsendet

Sigfrid Lundberg, sigfrid.lundberg@lub.lu.se
LuCEP/Biblioteksdirektionen
Lunds universitet
PO Box 134
S-221 00 Lund

SAMMANFATTNING

Denna text syftar till att ge en överblick över vissa områden av det digitala biblioteksområdet vilka är stadda i snabb utveckling. En utveckling som i hög grad katalyseras av utvecklingen av olika tillämpningar av eXtensible Markup Language, XML, som i år fyller fem år.

Utöver användningen av XML inom det digitala biblioteksväsendet ges en överblick över modernt tänkande när det gäller metadata, hur XML kan tänkas påverka hur vi tar del av, och hanterar elektroniska texter både i form av digitalt bevarande och publicering av t ex forskningsrapporter.

1. Inledning

XML fyller fem år i år. Födelsedagsbarnet är ett *metaspråk*. Det är inte ett språk man skriver texter i, utan ett språk man använder för att definiera andra språk.[XML] Språket har snabbt vunnit popularitet i vida kretsar till den grad att många har börjat lida av en slags "angle-bracket fatigue". Drabbas du av en oförklarlig matthet när du ser tecknen < och > bör du lägga undan denna text. Du är i riskzonen för att drabbas av denna form av depression.

Jag tänker mig att denna text skall kunna tjäna som en inledning för den tekniskt mindre bevandrade till landvinningar som gjorts inom det digitala biblioteksväsendet fram till XMLs femårsdag, och även en del teknologier som fått uppleva en renässans i XML-boomens kölvatten.

Innan jag går över till det egentliga innehållet vill jag säga några få ord om mig själv. Jag har arbetat med informationssökning på Internet sedan 1995. Hela tiden har markerad text varit min specialitet. Då började jag med robotar och automatisk indexering av webbsidor i tjänster som t ex Nordiskt Webbindex. Sedermera lade jag till metadata — katalogiseringsinformation — som bäddats in i webbsidorna i så kallade metataggar i tjänster som Svenska Miljönätet, Safari och Studera.NU. Jag har fortsatt i samma spår; att försöka hjälpa mina medmänniskor att hitta texter som de kan ha nytta och glädje av. Under mitt arbete har miljön jag arbetat inom blivit mer och mer kontrollerad. Nu är det strikta XML-baserade tjänster, som täcker alltfrån sökning bland digitaliserade medeltidshandskrifter till elektronisk publicering av vetenskapliga uppsatser.

Min text har tre delar. Först tar jag upp en rad "mjuka" ämnen, trender inom utvecklingen på metadataområdet. Både äldre standarder som utvecklats vidare och nya som aspirerar på att bli implementerade. Frågan är med vilka affärsmodeller de är förenliga och vilka olika funktionella krav de uppfyller. Standarderna

har uppkommit i olika ekonomiska eller politiska sammanhang som det inte går att bortse ifrån.

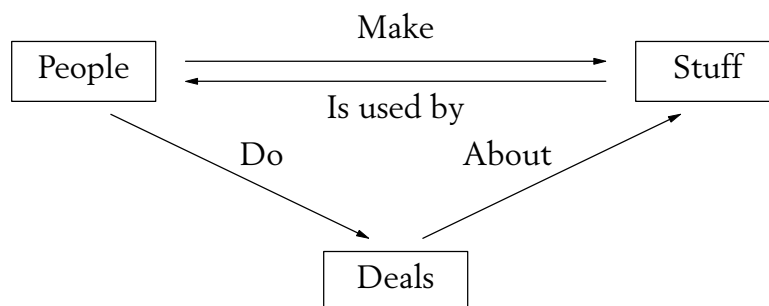
Därefter övergår jag till hur XML kan används för att koda data och dokument, och även metoder som används för att transportera XML över ett nätverk. In emellan gör jag helt personliga reflektioner om hur standardarder används och exempel på projekt och sammanhang där de förekommer inom Sverige.

2. Metadata, "Business models" och "data models"

För att någon skall bygga en digital informationstjänst, så måste den fylla något behov. Någon måste behöva den, och någon måste vilja betala den. Metoden för att översätta ett behov till betalningsvilja brukar kallas tjänstens "*business model*" eller dess *affärsidé*.

Vilken roll, om någon, spelar en affärsidé för hur en databas byggs, och vilka metadata som används? Metadata är big business. Tro alltså inte att metadata skapas av idealistiska bibliografer som vill möjliggöra högkvalitativa sökverktyg, drömmande bibliotekarier i fotriktiga skor. Eller att metadatatjänster enbart skapas av orakade datornördar[SHOES]

Metadata är en av grundvalarna i mediaindustrin. Denna lever på att äga, köpa och sälja nyttjanderätten till copyrightskyddat material av olika slag (se Figur 1). För att inse att detta är fallet, fundera en stund över Svenska Tonsättarens Internationella Musikbyrås [STIM] verksamhet.



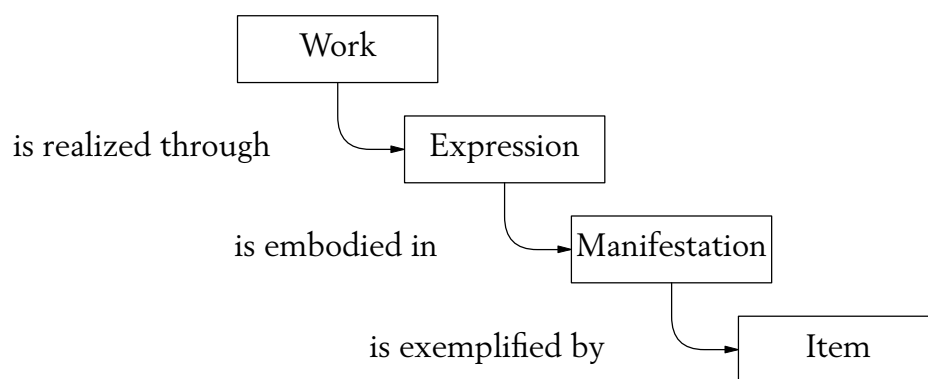
Figur 1. INDECs businessmodell, som ligger till grund för mycket tänkande för mediaindustrin när det gäller metadata. Efter Bearman et al. [BRMN]

Hur skall man få reda på alla rättsinnehavare involverade när man spelar låt oss säga Louis Armstrongs *Mack the knife*, alltså Broadwayversionen av Bert Brechts och Kurt Weills *Mackie Messer* på svensk radio? Brecht skrev librettot till *Die Dreigroschenoper*, *Tolvskillingsoperan* och Weill komponerade musiken, någon översatte till engelska, ytterligare någon skrev det jazzigare arrangemanget, slutligen har vi orkester, dirigent och Louis Armstrong själv. Alla personer eller organisationer som bidragit till verket kallar jag i fortsättningen för *agenter*. För att kunna fördela STIM-pengar till alla som bidragit krävs detaljerad information som tar hänsyn till alla uppgörelser som har gjorts kring det intellektuella innehållet i text och melodi, och de övriga "transformationer" verket genomgått innan det

hamnade i Sveriges Radios skivsamling.

Alla krav man kan ställa för att det skall vara meningsfullt att skapa sådana metadata är uppfyllda. Vi har en affärsidé; svenska tonsättare blir medlemmar och STIM tar till vara deras intressen. Liknande organisationer runt om gör motsvarande jobb, och tillsammans bevakar de systerorganisationernas medlemmars intressen internationellt.

Utan att egentligen bry oss om hur det går till i verkligheten, är det en lärorik övning att fundera över vilken information om *Die Dreigroschenoper* STIM och koalitioner inom mediasektorn behöver för att rätta konton skall berikas när Louis raspiga röst hörs i radio. Ett databasschema som klarar av detta måste vara baserad på en struktur som liknar den som visas i Figur 2. Denna struktur är hämtad från en publikation som heter *The functional requirements for bibliographic records*, FRBR. Om du inte visste det, så uttalas det förbör. Du måste nästan svälja r:en för att verka initierad.



Figur 2. De grundläggande entitetsrelationerna enligt IFLAs beskrivning av de funktionella kraven för en bibliografisk post.[FRBR]

Man kan framställa FRBRs fyra nivåer så här:

BERT BRECHT [*stolt*]: Jag kom med med idén, det abstrakta verket!

BRECHT & WEILL [*i kör*]: Vi skapade operan, idéns konstnärliga uttryck.

BRECHT [*leende*]: Jag iscensatte operan. Jag regisserade och så att verket fick en scenisk manifestation.

KVINNAN I PUBLIKEN [*Med en lycklig suck*]: Och jag var på premiären! Den första i en serie underbara föreställningar. Vart och ett exempel på Berts storhet.

JOHN GAY [*Överseende teatervisning*]: Jag gav ut *The Beggar's Opera* redan 1712.

Om vi för tillfället bortser från Gays kommentar, ser vi här ett exempel på hur en abstrakt modell av en kreativ process kan påverka hur man modellerar en bibliografisk post. Steget här är långt från den traditionella MARC-postens 999 fält, subfält från a-z och indikatorer. Fälten betecknas som bekant med sina nummer. Det ligger trettio års utveckling inom databasområdet mellan MARC och FRBR. En utveckling som stimulerats av idéer inom fält som objektorienterad programmering.

När det gäller STIM och dess broder- och systerorganisationer internationellt räcker inte relationerna som visas i Figur 2. En del av det som behövs finns faktiskt i Dublin Core Metadata Initiatives (DCMI) vokabulärer. Innan jag går in på dem är det några kraftfulla generella metadatakoncept jag vill diskutera.

2.1. I händelsernas centrum

Alla som arbetat i projektform vet att det finansierar och projektledare fäster störst vikt vid är tidsplanen, att den följs och att projektet levererar de produkter som specificerats. Det blir lättare att hålla reda på alla entreprenörer då.

Tänk på *Die Dreigroschenoper* som ett projekt, som genom en serie aktiviteter eller händelser leder oss från Berts första tomma vita pappersark i tjugotalets Berlin till en teaterscen i Amerika. Varje milstolpe i projektet är en *händelse* (i vardagslag kallar jag detta — på moderna halvsvenska för ett "event"). Den eller de som arbetar med händelsen kommer till stånd kallar jag agenter. En latinist jag känner säger att agent är presens particip av verbet "agere"— handla, göra, utföra, alltså görande, handlande, utförande. Så nu vet du det.

Det existerar två sätt att beskriva resurser — de må vara inspelade sånger eller böcker. Antingen beskriver vi händelserna som ledde fram till objektet vi håller i handen, eller så beskriver vi objektet. Detta kan, men behöver inte, leda fram till exakt samma beskrivning, men den kan se lite olika ut. Dessa två beskrivningssätt leder oss till två olika typer av metadatasystem:

- Händelsebaserad (event based)
- Resursbaserad (resource based)

I en händelsebaserad beskrivning orsakas varje händelse av en agent. I vårt exempel var den förste agenten Bert. En agents relation till verket beskrivs genom hans eller hennes roll. Berts relation till verket är författarens. Nästa händelse är att verket tonsattes. Och så vidare.

Det bör inte förvåna dig att de flesta av mediaindustrins metadatasystem (t ex INDECS och MPEG-7) är händelsebaserad. Principiellt är det ingen större skillnad mellan att beskriva logistik, entreprenörers verksamhet i ett husbygge och vägen mellan ett oskrivet ark i Berlin och en musical på Broadway.

Översättare och murare, liksom illustratörer och målare vill ha lön för sina mödor. De vill sätta mjölk på bordet åt sina barn. Händelsebaserade metadata visar på vad var och en har gjort, och underlättar STIMs arbete. Ännu så länge har dock ingen kommit på idén att ta betalt av folk som tittar på hus. Man har försökt i Bomässor och annat, fast det var inte särskilt kul.

2.2. Rum för namn

Tänk på metadataelementet titel. Tänk på vilka värden det kan ta. Exempelvis:

Titel: *Ett drömspel*

eller

Titel: Fröken Harriet Bosse

Vi kan lätt tänka ut fler intressanta titlar, som t ex *En dåres försvarstal*, "His majesty the king", *Krig och fred*, "His high velocity the air marschal" och, som sagt, "fröken". Vill ni ha ett litet barn med mig, fröken Bosse? lär August Strindberg ha sagt medan han arbetade med *Ett drömspel* i Danmark.

Jag förmodar att du hajar galoppen här. Såväl svenskans "titel" som engelskans "title" är tvetydiga och kan beteckna såväl namnet på ett verk som den sortens titlar som på engelska brukar kallas honorific. Den uppmärksamme läsaren ställer naturligtvis genast frågan varför jag vill blanda boktitlar och hederstitlar i samma burk. Mitt svar på den frågan är att jag inte nödvändigtvis vill det, men titulaturproblem är bara ett av de mest uppenbara exemplen på en hel klass av problem.

För att ta ett relaterat problem, de standardiserade sökattributen i Z39.50. Jag gillar dem, skarpt. Det är en lysande idé att det skall vara möjligt att skriva ett sökprogram och veta att sökattributet för "personal name" är 1 och att det för "title" alltid är 4. Myntet har en baksida. Ett som jag ständigt stött på i mitt arbete när jag implementerat Z39.50 är klassifikationssystemen. Jag har följande, vilka ISO tyckte borde räcka: "Dewey", "UDC", "Bliss", "LCC", "NLM", "NAL", "MOS" och så naturligtvis "Local classification". Som en fattig kusin från landsorten har jag alltid valt just detta sökattribut, "local classification". Smaka på det igen. *Local classification*. Hur är det nu, egentligen. Hur många kusiner är från landsorten, och hur många är från stan? Jag bara undrar.

Tänk om alla kusiner reser sig som en man. Ropet skallar: Landsortskusiner, förenen eder! Vi skall i dag solidariskt kunna utföra sökningar i varandras kataloger. Ack, söndrade vi falla. Vi vet ju både du och jag att alla kusiner från landet har var sitt klassifikationssystem, och alla är de local classification.

Idén om namnrymder är gammal, och överfördes till markeringspråkens värld tidigt i utvecklingen av XML. Det fanns plötsligt en metod att låta "fröken" samleva med "Ett drömspel" och låta båda vara titlar. Varje titel kan få ett eget rum, de kan leva sina liv i separata namnrymder. En namnrymd för element som beskriver verk och en annan för element som beskriver personer. Varför är nu detta viktigt och revolutionerande?

Namnrymder är en viktig uppfinning av flera skäl. För det första, som programmerare är jag inte smart nog att skriva datorprogram som kan skilja godtyckliga titlar på verk från godtyckliga titlar på personer i ett godtyckligt sammanhang. För det andra, genom namnrymder kan olika projekt eller organisationer komma med individuella bidrag till en gemensam samling med metadatavokabulärer. Nya tjänster och metadatasystem kan återanvända det som var lyckat och bra i tidigare versioner.

2.3. DCMI — samling kring en samling av element

Informationen i Tabell 1 skall in någonstans i dokumentet, i anslutning till detta ännu icke skrivna avsnitt. Men jag tror nog att det skall ha en annan form. Jag gillar inte min egen tabell. Den är ganska missledande.

3. Dokument, data och "Content models"

Vad är ett *dokument* för dig? För mig brukade till för några år sedan dokument vara ett högtidligt ord. Ett dokument var handskrivet, kanske med bläck och på fint papper. Kanske det var skrivet på pergament. Ett dokument var alltid gammalt, och alltid värdefullt.

Dokument var texter som dokumenterade eller belyste något; ett skeende i historien till exempel. Gamla gulnade brev från mormors mor till mors morfar, sådana kändes för mig som dokument. Sen kom informationsteknologin och med den dokumentinflationen. På min dator har jag en massa dokument, och du har väl det också. Ofta heter de något i stil med *.doc. Oavsett om de är brev, artiklar eller kanske till och med någon bok. När vi ser ett sådant dokument utskrivet, vet vi trots allt vad det är. Det gör vi eftersom ett brev typiskt har:

- ett brevhuvud
- en datering.
- en adressat

Tabell 1. Några av de vanligaste element i DCMIs vokabulärer. Dessa fördelar sig över två namnrymder, vilka i tabellen på XMLmanér anges med prefixen dc: respektive dcterms:

Content (Innehållsrelaterade)	Intellectual property (Upphovsrelaterade)	manifestation (Versionsrelaterade)
dc:title	dc:creator	dc:date, dcterms:created, dcterms:issued, dcterms:valid, dcterms:modified, dcterms:available
dc:subject	dc:publisher	dc:type
dc:description, dcterms:abstract, dcterms:tableofcontents	dc:contributor	dc:format, dcterms:extent, dcterms:medium
dc:source	dc:rights	dc:identifier
dc:language		
		dc:relation, dcterms:IsVersionOf, dcterms:HasVersion, dcterms:IsReplacedBy, dcterms:Replaces, dcterms:IsRequiredBy, dcterms:Requires, dcterms:IsPartOf, dcterms:HasPart, dcterms:IsReferenceBy, dcterms:References, dcterms:IsFormatof, dcterms:HasFormat
dc:coverage, dcterms:temporal, dcterms:spatial		

Exempel 1. Två sätt att skriva en adress i HTML, som i de flesta bläddrare ser exakt likadana ut.

```
<I>Sigfrid Lundberg
LuCEP/Biblioteksdirektionen
Lunds universitet
PO Box 134
S-221 00 Lund</I>
```

```
<ADDRESS>Sigfrid Lundberg
LuCEP/Biblioteksdirektionen
Lunds universitet
PO Box 134
S-221 00 Lund</ADDRESS>
```

- en inledande hälsningsfras.
- ett meddelande.
- en avslutande hälsningsfras.

□ en underskrift.

Det finns ingen generell metod som gör det möjligt att öppna ett brev skrivet med någon populär ordbehandlare, och konstatera: *Detta är ett brev*. I princip är enda metoden att öppna dokumentet och läsa det. Rent visuellt finns då säkert de element jag nämner ovan. Finns elementen där i någon djupare mening? Vem bryr sig? Ganska få. Bortsett från registratören vid ämbetsverket och arkivarien och vissa andra som kanske har en lite högtidlig syn på begreppet dokument. De som tänker som så att dagens *.doc skall bli framtidens gulnade och värdefulla urkunder.

XML, och dessförinnan SGML, ger oss möjlighet att definiera typer av dokument, i en sådan formell mening. En sådan *dokumenttyp* kan vara just *brev* och *kontrakt*. Andra dokumenttyper skulle kunna vara *artikel* eller *bok*. För varje sådan typ kan man ställa upp en formell definition som måste vara uppfylld för att ett dokument skall vara exempelvis ett brev.

När väl en sådan definition är gjord, kan man med datorprogram kontrollera om definitionen följs. Programmet kan naturligtvis inte avgöra om jag menar allvar när jag avslutar ett brev med "Din för evigt, Sigfrid". Däremot kan det kontrollera är om ett dokument innehåller en sådan avslutningsfras. För att programmet skall kunna göra detta måste frasen också markeras på exakt det sätt definitionen föreskriver.

XML gör alltså en distinktion mellan innehåll och form, och koncentrerar sig på innehållet. Ett mycket enkelt exempel på skillnaden mellan innehåll och form i ett markeringsspråk är taggen <ADDRESS>...</ADDRESS> i vanlig HTML. Den är mycket ovanlig numera. Den skapades en gång för att användas på slutet av webbsidor för att berätta vem det var som var ansvarig för sidan, och hur man skulle kunna ta kontakt med sidansvarig. Den enda visuella effekten taggen normalt har är att dess innehåll kursiveras. De som inte gillar kursivering där skippar den taggen av den anledningen, och de som vill ha kursivering sätter förmodligen in ett <I>...</I> i stället. Det ser ju likadant ut! (se Exempel 1).

Som den typiske gnällspik jag är kan jag konstatera att folk ansvariga för HTML-slöjd helt enkelt har struntat i den distinktion som en gång fanns mellan form och innehåll även i HTML. Därför bryr sig naturligtvis inte heller utvecklare av sökmaskiner eller liknande produkter.

Om vi nu bryr oss, och vill att ett brev skall vara ett brev i djupaste möjliga mening, så skulle ett sådant kunna markeras i XML som i Exempel 2. Detta är ett ganska realistiskt exempel. Namn, datum och platser är markerade. En samling brev av det här slaget skulle kunna lagras i en databas och om man gjorde det, skulle det också vara möjligt att söka samlingen med precision. Till en tryckt upplaga skulle det vara lätt att med automatik generera alfabetiska index av brevskrivare, adressater, personer som nämns i brödtexten, platser. I hypertext kan sådana element markeras och länkas till såväl textkritiska noter som till sökningar i databasen. Vilka brev skrev Strindberg medan han bodde i Lund, vem skrev han till i mars månad 1897? Ett elektroniskt dokument som det i Exempel 2 det håller. XML är elektroniskt pergament av högsta kvalitet. Det skall gå att läsa i framtiden, om formatet för markeringen är välkänd och väl dokumenterad.

För varje dokumenttyp finns det vissa givna typer av innehållsliga element man kan vänta sig. Listan med innehållsliga element för ett brev ovan stämmer väl överens med såväl intuition som det jag använt i Exempel 2.

Det kan vara en intressant övning att fundera över vad andra dokumenttyper skulle kräva för markeringar. Jag tar upp några. En vetenskaplig artikel ser

Exempel 2. Delar av ett brev markerat i XML (enligt Text Encoding Initiative[TEI]). Brevet är från August Strindberg till sin dotter med Frida Uhl, Kersin Strindberg. (August Strindberg. Brev till min dotter Kerstin. Albert Bonniers Förlag. Stockholm 1981. Det kommer en riktig referenslista när min XSLT fungerar igen)

```
<body>
  <opener>
    <address>Till
      <name
        reg="Strindberg, Kerstin"
        type="person">Kerstin Strindberg</name>
    </address>
    <dateline>
      <name type="place">Lund</name>,
      <date value="1897-03-29">29 Mars 1897</date>
    </dateline>
    <salute>Kära stackars Lilltösen,</salute>
  </opener>
  <p>Du har varit sjuk, har haft det svårt och jag
trodde du var ond. Förlåt! Men jag anade det,
kände det. Den natten du sjuknade drömde jag,
att jag såg dig falla ned från en trädtopp,
du föll från gren till gren; men när jag full
av ängslan kom springande, satt du helt lugnt på
nedersta grenen. Alltså tur i sista ögonblicket.</p>
  <p>Vad säger
    <name type="person"
      reg="Uhl, Frida">Frida</name>-mamma om din sjukdom?
  </p>
  <p><name type="person"
    reg="Uhl, Marie">Mormor</name> har rätt:
    livet förgår och det bästa få vi
    ej njuta. Vi leva var och en i sin ödemark,
    främmande för varandra, åtskils, ansatta av
    samvetsagg. Så kan våra öden länkas.</p>
  <p>Här händer ingenting. Har tråkigt,
    då jag icke kan arbeta; väntar bara på
    ett slut, vilket som helst, i eländet.</p>
  <p>Våren har gått sin väg, luften är svart
    som i november; regn som i oktober.
    Kallt som i december.</p>
  <closer>
    <salute>Lev väl!</salute>
    <signed>Pappa
      <name type="person"
        reg="Strindberg, August">August</name>.
    </signed>
  </closer>
</body>
```

annorlunda ut:

- titel
- lista av författare
- ett antal sektioner, typiskt med namn som "Introduction", "Materials and methods" och "Discussion"

- litteraturförteckning
- figurer
- tabeller

Exempel 3. Strindbergsbrevet i Exempel 2 i klartext

Till Kerstin Strindberg

Lund 29 Mars 1897

Kära stackars Lilltösen,

Du har varit sjuk, har haft det svårt och jag trodde du var ond. Förlåt! Men jag anade det, kände det. Den natten du sjuknade drömde jag, att jag såg dig falla ned från en trädtopp, du föll från gren till gren; men när jag full av ängslan kom springande, satt du helt lugnt på nedersta grenen. Alltså tur i sista ögonblicket.

Vad säger Frida-mamma om din sjukdom?

Mormor har rätt: livet förgår och det bästa få vi ej njuta. Vi leva var och en i sin ödemark, främmande för varandra, åtskils, ansatta av samvetsagg. Så kan våra öden länkas.

Här händer ingenting. Har tråkigt, då jag icke kan arbeta; väntar bara på ett slut, vilket som helst, i eländet.

Våren har gått sin väg, luften är svart som i november; regn som i oktober. Kallt som i december.

Lev väl!

Pappa August.

En typisk roman skulle sannolikt ha en enklare struktur:

- kapitel
- och inuti varje kapitel en massa stycken
- och inuti varje stycke dialog

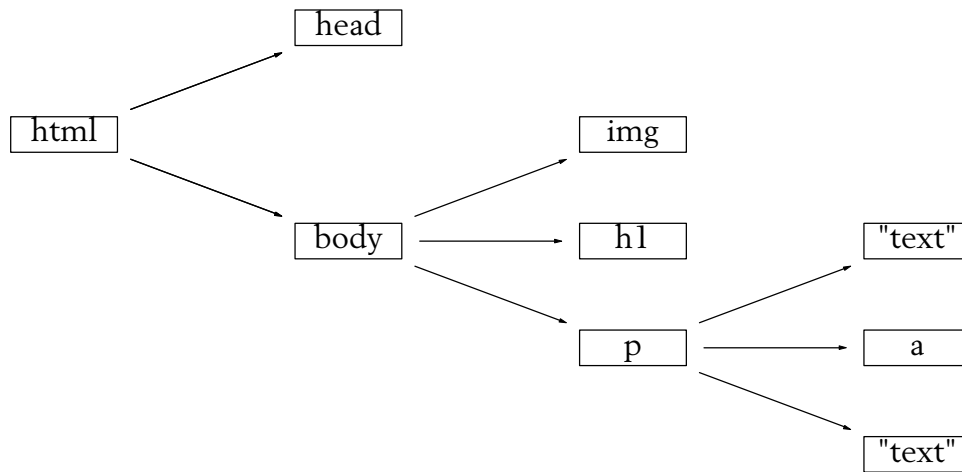
De innehållsliga strukturerna i olika dokumenttyper leder fram till olika former av markeringar. När vi talade om databaser och metadata var begreppet "datamodell"; i samband med XML och markeringspråk har vi ett analogt begrepp, "innehållsmodell" (content model).

4. Träd, stigar och stilar

5. Diskussion

XML har blivit mjukvaruindustrins gullgosse. Det har, tyvärr, spridits en uppfattning att applikationer baserade på < och > är bättre än andra, på grund av någon inneboende kvalitetsskillnad. Så är det naturligtvis inte. Trots detta kan man med fog säga att femåringen har öppnat en rad nya fält som hade varit omöjliga för fem år sedan. Jag tror inte att XML är den verkande orsaken till detta, utan att femåringen verkar som en katalysator.

Det mesta av det vi idag gör med XML hade varit möjliga att göra med Standard Generalized Markup Language (SGML), Z39.50 och andra äldre standarder.



Figur 3. En trädrepresentation av ett enkelt html dokument.

Det är bara det att få gjorde det. XML inbjuder till byggandet av strukturer för utbyte av data. Det är tillräckligt många som har kastat sig över de nya teknologierna för att det skall bli en kritisk massa. Vad som en gång var svårt att göra har blivit enklare, och *roligare*, att göra.

6. Osorterade länkar. Metadata i XML och annat.

Vilka också visar vad resten delvis kommer att handla om.

- Metadata Principles and Practicalities
<http://www.dlib.org/dlib/april02/weibel/04weibel.html>
- Expressing Simple Dublin Core in RDF/XML
<http://www.dublincore.org/documents/2002/07/31/dcmes-xml/>
- The Open Archives Initiative Protocol for Metadata Harvesting
<http://www.openarchives.org/OAI/openarchivesprotocol.html>
- Metadata Object Description Schema
<http://www.loc.gov/standards/mods/>
- Reference Manual for the MASTER Document Type Definition
<http://www.tei-c.org/Master/Reference/>
- Encoded Archival Description (EAD)
<http://lcweb.loc.gov/ead/>

7. Textmarkering i XML. Olika språk

- Daisy Consortium
<http://www.daisy.org/>
- TEI Consortium
<http://www.tei-c.org/>
- TEI projects
<http://www.tei-c.org/Applications/index-lang.html>
- DocBook
<http://www.oasis-open.org/committees/docbook/>

- Open Office
<http://www.oasis-open.org/committees/office/>

Referenser

[XML] <http://www.w3.org/TR/REC-xml>
[SHOES] http://www5conf.inria.fr/fich_html/slides/invited/IS1/all.htm
[STIM] <http://www.stim.se/>
[BRMN] <http://www.dlib.org/dlib/january99/bearman/01bearman.html>
[FRBR] <http://www.ifla.org/VII/s13/frbr/frbr.pdf>
[TEI] <http://www.tei-c.org/>