

XML i 10 punkter

XML, XLink, Namespace, DTD, Schema, CSS, XHTML ... Om du inte har arbetat med XML, så kan det vara svårt att veta var man skall börja. Denna sammanfattning i tio punkter försöker att beskriva tillräckligt mycket av de grundläggande begreppen för att ge en nybörjare en möjlighet att se skogen och inte bara alla träden. Och om du skall presentera XML, varför inte börja med dessa tio punkter?

1. XML används för att strukturera data



Strukturerad data innefattar sådant som kalkylblad, adresslistor, konfigureringsparametrar, finansiella transaktioner och tekniska ritningar. XML är en uppsättning regler (du kan också tänka på dem som riktlinjer eller överenskommelser) för hur textuella format skall konstrueras, format som låter dig strukturera dina data. XML är inte ett programmeringsspråk, och du behöver inte vara en programmerare för att använda det eller lära dig det. XML gör det lätt för en dator att generera data, läsa data och garantera att datastrukturer är entydiga. XML undviker vanliga fällor i språkkonstruktion: det är utvidgningsbart, plattformsoberoende och det ger möjlighet att uttrycka sig såväl oberoende som specifikt för plats, kultur och språk.

2. XML ser nästan ut som HTML



Liksom HTML använder XML *taggar* (märkord avgränsade av '<' och '>') och *attribut* (av formen `namn="värde"`). HTML definierar dock vad varje tagg och attribut betyder, medan XML använder taggar bara för att avgränsa delar av data från varandra, och överlåter tolkningen av data till den tillämpning som läser dessa data. Med andra ord, om du ser "<p>" i ett XML-dokument så ska du inte förutsätta att det avgränsar ett textstycke. Beroende på sammanhanget kan det betyda att efterföljande data är ett pris, en parameter, en person, ett p... (och vem har sagt att det måste vara ett ord som börjar med ett "p"?).

3. XML är text, men är inte avsett att läsas



Program som genererar kalkylblad, adresslistor och andra strukturerade data, lagrar ofta dessa data på skivminne, i binär eller textuell form. En fördel med textuell form är att det gör det möjligt för människor att läsa data utan att använda det program som genererade dessa data. I värsta fall kan du alltid använda en vanlig ordbehandlare. Textuell representation gör det också lättare för programutvecklare att avlusa sina program. Liksom HTML är XML-filer textuella filer som människor inte borde behöva läsa direkt, men det kan finnas situationer där det är nödvändigt. Till skillnad från HTML så är XML:s strukturregler entydiga. Om man glömmer en tagg, eller definierar ett attributvärde utan citationstecken så är XML-filen oanvändbar, medan sådana felaktigheter ofta tolereras och ibland är uttryckligen tillåtna i HTML. Den officiella XML-specifikationen förbjuder tillämpningsprogram att försöka gissa vad en felaktig XML-fil betyder; om filen innehåller fel så ska tillämpningen avslutas och felet rapporteras.

4. XML är avsiktligt ordrikt



Eftersom XML är ett textformat och använder taggar för att avgränsa data, så är XML-filer oftast större än motsvarande binära filer. Detta är en effekt av ett medvetet val av de som specificerade XML. Fördelarna med ett textuellt format är uppenbara (se punkt 3 ovan), och nackdelarna kan ofta kompenseras för på andra nivåer. Skivminne är nu billigare än tidigare, och komprimeringsprogram som zip och [gzip](#) kan effektivt komprimera filer. Dessutom kan kommunikationsprotokoll som modem-protokoll och [HTTP/1.1](#), det viktigaste protokollet för webben, komprimera data under körning, och därmed utnyttja bandbredd lika effektivt som för binära filer.

5. XML är en familj av teknologier



[XML 1.0](#) är specifikationen som definierar vad "taggar" och "attribut" är. Utöver XML 1.0 så är "XML-familjen" en växande uppsättning av moduler som erbjuder användbara tjänster för att täcka viktiga och ofta efterfrågade behov. [Xlink](#) beskriver ett standardiserat sätt att lägga till hyperlänkar till en XML-fil. [XPointer](#) och [XFragments](#) är språk under utveckling, avsedda att för att kunna peka ut delar av XML-filer. En XPointer är ungefär som en URL, men istället för att utpeka dokument

på webben pekar den på data i en XML-fil. [CSS](#), språket man uttrycker formatmallar i, kan tillämpas på XML på samma sätt som på HTML. [XSL](#) är det [avancerade språket](#) för att uttrycka formatmallar ("stylesheets"). Det baseras på [XSLT](#), ett transformeringsspråk som används för att stuva om, lägga till och ta bort taggar och attribut. [DOM](#) är en standardiserad uppsättning funktionsanrop som utnyttjas av program för att bearbeta XML (och HTML) filer. [XML Schemas 1](#) och [2](#) hjälper utvecklare att fullständigt definiera strukturen på egna XML-baserade format. Det finns många fler moduler och verktyg tillgängliga eller under utveckling. Bevaka gärna [W3C:s sida med tekniska rapporter](#).

6. XML är nytt, men inte så nytt



Utvecklingen av XML började 1996 och blev en "W3C Recommendation" i februari 1998. Detta kan kanske få en att tro att det är en omogen teknologi. Men i själva verket är teknologin inte ny. Innan XML fanns SGML, utvecklat under första delen av 1980-talet, standardiserat av ISO 1986, och använt inom stora dokumentationsprojekt. Utvecklingen av HTML började 1990. XML:s utvecklare valde, baserat på erfarenheter från HTML, helt enkelt de bästa delarna av SGML, och resultatet blev lika kraftfullt som SGML, och mycket mer välstrukturerat och enkelt att använda. Det är ibland svårt att skilja utveckling från omvälvning... Och det bör betonas att medan SGML mestadels används mest för teknisk dokumentation och i mindre utsträckning för andra typer av data, så gäller det helt omvända förhållandet för XML.

7. XML leder till HTML och till XHTML



En viktig tillämpning av XML är dokumentformatet för W3C:s XHTML, efterföljaren till HTML. XHTML och HTML har en stor uppsättning gemensamma element. Grammatiken har i viss utsträckning ändrats för att anpassats till XML:s grammatikregler. Ett dokument som är "XML-baserat" ärver sin grammatik från XML och begränsar det på olika sätt (t.ex. tillåter XHTML "<p>", men inte "<r>"); den ger också en betydelse till grammatiken (XHTML säger att "<p>" betyder "paragraph", och inte "pris", "person", eller annat).

8. XML är modulärt



XML ger dig en möjlighet att definiera nya dokumentformat genom att kombinera och återanvända andra format. Eftersom två format utvecklade oberoende av varandra kan ha element eller attribut med samma namn, så måste man vara uppmärksam (betyder "<p>" "stycke" från det ena formatet eller "person" från det andra?). För att eliminera namnkollisioner då man kombinerar format erbjuder XML en [namnrymdsmekanism](#). XSL och [RDF](#) är bra exempel på XML-baserade format som använder namnrymder. [XML Schema](#) har konstruerats för att denna modulariseringsprincip skall kunna användas då man definierar ett nytt dokumentformat som en kombination av två andra dokumentformat.

9. XML utgör grunden för RDF och "den semantiska webben"



XML erbjuder en entydig grammatik för W3C:s RDF, språket för att uttrycka metadata (i själva verket godtycklig kunskap). RDF är ungefär som hypertext lyft till en högre nivå. Där hypertextlänkar knyter samman bitar av text och är tyst vad gäller hur dessa textbitar hänger samman, kan RDF länka samman vad som helst och namnger den relations som knyter samman dem: "A är priset för B" kan vara en relation mellan ett objekt och en penningssumma, "A är tyngre än B" kan vara relationen mellan två sumo-brottare, "A orsakar B" kan vara relationen mellan en dusch och det faktum att du blir våt. För att kommunicera kunskap, i XML/RDF eller i vanlig svenska, måste både människor och maskiner komma överens om vilka ord som skall användas. En exakt definierad uppsättning ord som beskriver en aspekt på verkligheten (alltifrån "shopping" till "matematisk logik") kallas en "ontologi". RDF, ontologier, och kunskapsrepresentation som gör det möjligt för datorer att hjälpa människor att göra sitt jobb, är alla del i verksamheten i [W3C:s Semantic Web Activity](#).

10. XML är utan licens, plattformsoberoende och har brett stöd



Genom att välja XML som grund för ett projekt så kan du få tillgång till en stor och växande uppsättning verktyg (det kanske redan finns ett som gör det du behöver) och yrkesmänniskor med erfarenhet av teknologin. Att välja XML är ungefär som att välja SQL för databaser: du måste i alla fall bygga din egen databas och dina egna program och procedurer som hanterar den, och det finns många verktyg

tillgängliga och många yrkesmänniskor som kan hjälpa dig. Och eftersom XML inte kräver licens så kan du låta din egen mjukvara utnyttja XML utan att behöva betala någon för dess användning. Det växande stödet för XML betyder att du inte behöver binda dig till en viss leverantör. *XML är inte alltid den bästa lösningen, men betrakta den som ett värdefullt alternativ.*

Bert Bos

Revised 13 Nov. 2001 (last update: \$Date: 2001/11/21 19:45:05 \$)

Created 27 Mar 1999 ([Previous version](#))

Översatt av [Olle Olsson](#)

[Copyright](#) © 1999-2000 [W3C](#)® ([MIT](#), [INRIA](#), [Keio](#)), All Rights Reserved.