



XML-familjen

Vad är XML Formatmallar?





XML-familjen

Vad är XML Formatmallar?

Publikationen kan beställas från:

Statskontoret

Publikationsservice

Box 2280

103 17 Stockholm

Tfn: 08-454 46 43

Tfx: 08-454 46 45

E-post: publikations.service@statskontoret.se

Mer information om Statskontoret finns på Internet:

www.statskontoret.se

Kontaktpersoner Statskontoret:

Gunnar Hansson

gunnar.hansson@statskontoret.se

© STATSKONTORET

ISBN: 91-7220-430-3

Layout: Dreamforce Infomedia, Solna

Tryck: Novum Grafiska AB 2000

Innehållsförteckning

Inledning	5
Bakgrund	7
Historik	9
CSS	11
CSS uppbyggnad	11
Formateringsmöjligheter	11
Kontext och arv	13
Regler för företräde	14
XSLT	15
Varför transformera?	15
XSLT:s olika delar	17
Rotelement	17
Mallregler	17
Utformat	18
Övriga element	19
XSL Formatting Objects	21
Formatering	21
Sidhantering	21
Block	22
Case	23
Användningsområden	23
Standarderna i praktiken	23
Framtiden	29
Statskontorets publikationer för 1999 och 2000	31

Inledning

Denna rapport riktar sig i första hand till IT-verksamma inom svensk offentlig sektor. Den ingår i rapportserien XML-familjen och syftar till att belysa det faktum att XML är en av ett antal tekniska specifikationer från World Wide Web Consortium (W3C) som tillsammans skapar förutsättningar för betydligt förbättrad hantering av information. Rapporterna vill ge kortfattade lättlästa beskrivningar av de nya teknikerna.

Denna rapport behandlar XSL och CSS d v s de formatmallar som kan användas tillsammans med XML.

Rapporten är framtagen av Statskontoret inom ramen för arbetsområdet ärende- och dokumenthantering, projektledare Jan Lundh i samarbete med Daniel Björkman, Tero Ahola och Christoffer Andreasson på Improve.

I serien ingår:

Vad är XML?	Statskontoret 2000:30
Vad är XML Familjen?	Statskontoret 2000:31
Vad är XML Schema och XML DTD:er?	Statskontoret 2000:32
Vad är XML Formatmallar?	Statskontoret 2000:33
Vad är XML Länkar?	Statskontoret 2000:34
Vad är Document Object Model (DOM)?	Statskontoret 2000:35
Vad är XML Query Language (XQL)?	Statskontoret 2000:36

För aktuell information om de olika specifikationernas status se www.w3c.org. För svenska aktiviteter kring XML se www.xmlakademin.nu.

Stockholm december 2000

Jan Lundh

Bakgrund

Det sätt på vilket information presenteras för oss är viktigt. Detta gäller oavsett om presentationen sker via tal eller ett medium som Internet eller via papper. Är inte presentationen noga förberedd så får människor det svårare att ta till sig den information man vill förmedla.

För att andra skall kunna ta till sig information är det viktigt att presentationen är anpassad efter det medium den skall förmedlas på. Det gäller att planera layout och utseende efter detta. Det kan t. ex gälla:

Vilket teckensnitt skall man använda?

Är det ändamålsenligt att använda samma teckensnitt för både papper och Internet?

Hur skall indrag av text i olika sammanhang se ut?

Hur stort skall mellanrummet mellan rubrik och brödtext var på olika nivåer?

En annan sak att beakta är att den struktur som vi skapar information utifrån inte alltid är den struktur vi vill använda när vi presenterar informationen. Förutom att man skapar ett utseende för presentation av informationen bör man även skapa en struktur för presentationen. Presentationsstruktur innebär att man helt enkelt stuvor om i sin information för att göra den mer pedagogiskt riktig och lättillgänglig.

När man använder XML för att hantera sin information har man valt att separera utseende från innehåll. För att presentera XML behövs således särskilda standarder. CSS, XSLT och XSL Formatting Objects är formatmallar och transformeringsverktyg som används tillsammans med XML och som möjliggör skapande av presentationsstruktur och -utseende för XML-dokument. Det finns många fördelar med att separera presentation från struktur. Ett exempel:

Antag en myndighet med många organisatoriska delar som ger ut dokument som beskriver verksamhetsregler som styr olika beslut. Myndighetsdel A använder sig av Word 6.0, myndighetsdel B använder sig av HTML utan stylesheet, medan myndighetsdel C använder sig av XML. När de olika myndigheterna producerat ett 100-tal beskrivningar bestäms från centralt håll att beskrivningarna skall ges en myndighetsgemensam utformning. Myndighetsdel A och B kommer att få gå in i varje dokument som skapats och ändra detta för hand. Myndighetsdel C behöver bara ändra en gång i sin formatmall som används för att styra presentationen av samtliga beskrivningar som skapats.

Man kan skapa formatmallar för olika medier såsom papper, Internet och multimedia. I och med att man har en och samma informationskälla, avspeglar

sig eventuella förändringar i informationen direkt i de olika presentationsformat man valt att använda. Har man inte samma informationskälla måste informationen uppdateras på flera ställen, vilket ökar risken för felaktigheter. Med formatmallar når man ut till olika medier samtidigt som man uppdaterar på ett ställe, vilket underlättar återanvändning av informationen.

Denna skrift riktar sig mot de inom offentlig sektor som hanterar och skapar information. Skriften vill ge en inblick i varför man skall ha formatmallar och hur de är uppbyggda.

Historik

Tankarna kring formatmallar har funnits under en längre tid och har sin grund i behovet att separera information från presentation. En följd av en sådan separation är att informationen blir mediaoberoende. HTML är inte gjort för layout, ändå används många av de märkord som finns just för detta. I HTML-världen har man länge efterfrågat en bättre kontroll över layout av information. Det sätt på vilket man initialt uppnådde detta var genom att lägga till nya märkord. Dessa var ofta specifika för en browser och tankarna med att HTML skulle vara plattformsoberoende tappades bort. CSS-standarderna skulle komma att förändra detta.

Arbetet med en standard för formatmallar som skall användas på Internet påbörjades 1995. Året därpå lades det första förslaget för cascading stylesheets (CSS) fram av Hakon Lie, medlem av W3C. Detta blev sedermera en rekommendation, CSS level1 (CSS1). I och med denna rekommendationen tog man ett steg närmare plattformsoberoende och man minskade behovet av att missbruka HTML:s märkord för layout. Arbetet med CSS fortsatte och utmynnade i maj 1998 i CSS Level2 (CSS2) som innehåller fler möjligheter till layout och tankar om att skapa medieoberoende formatmallar. För tillfället pågår arbete med CSS Level3.

XML och HTML har sitt ursprung i Standard Generalized Markup Language (SGML). SGML har en standard för presentation knutet till sig som kallas Document Style Semantics and Specification Language (DSSSL). Denna standard kom 1990 då man såg behovet att ha en standard för formatering som var oberoende av olika plattformar. Tanken var att DSSSL skulle användas för att formatera SGML. Precis som SGML är DSSSL en väldigt omfattande standard som gör det svårt och dyrt att utveckla programvara som till fullo stöder den. Därför har man skapat XSL. Mycket av de tankar som fanns vid skapandet av DSSSL har förts över till XSL. Utöver detta så återfinns mycket av syntaxen från CSS i XSL.

När arbetet med XSL påbörjades var den tänkt som en rekommendation. Det var först en bit in i arbetet som man delade upp den i två delar. Detta skedde eftersom XSL behandlade två separata bitar, transformering och formatering. De resulterande delarna blev dels XSLT (transformering) som antogs som rekommendation 16 november 1999 och XSL FO (formatering), eller rätt och slätt XSL, som ännu inte blivit rekommendation.

CSS

CSS står för Cascading Style Sheets. Rekommendationen finns idag i två nivåer, CSS1 och CSS2. Den senare utökar den förra med ny funktionalitet. Syftet med CSS är att man skall kunna skapa ett utseende för sina dokument, där informationen om utseendet är separerat från informationen i källdokumentet. Både Netscape och Internet Explorer från version 4 och uppåt, stöder vissa delar av CSS. Troligtvis kommer man efterhand att utöka stödet av CSS rekommendationen i sina produkter.

CSS uppbyggnad

CSS är byggt på ett sådant sätt att det är skalbart, det vill säga möjligheten att bygga ut och bygga på befintliga lösningar är stor. Vi vill framför allt peka på tre saker som gör CSS nyttigt:

- CSS formateringsmöjligheter
- CSS regler bygger på arv och agerar efter kontext
- CSS har regler för företrädesrätt.

Formateringsmöjligheter

CSS erbjuder ett brett spektra av möjligheter för att skapa ett utseende på sin information. Sättet som man skapar dessa formateringsregler på är relativt enkla. För att man skall få CSS att fungera förutsätts att man har ett dokument med information som är uppmärkt på något sätt, då reglerna man skapar relaterar till denna uppmärkning. Sålunda förutsätts att man har ett HTML- eller XML-dokument som innehåller informationen som skall formateras.

Syntaxen för att skapa formateringsregler med hjälp av CSS ser ut så här:

```
element {regel}
```

Om man applicerar detta på exempelvis `<p>` i HTML med textstorleken 12 punkter och fetstil:

```
p {font-size:12pt;  
font-weight:bold;}
```

Skulle man istället välja att använda regeln direkt på ett XML-dokument, byter man helt enkelt ut 'p' mot ett elementnamn som man själv har definierat. Använder man exempelvis <namn>, så byter man ut 'p' mot 'namn'.

Om det är så att flera element i källdokumentet skall presenteras på samma sätt så kan samma formateringsregel användas till flera element.

```
namn, adress {display:block;}
```

När man gör på detta sätt säger man att man grupperar element.

Man kan dela upp de regler som finns i CSS i två grupper. Den ena hanterar hur olika textmassor förhåller sig till varandra, medan den andra delen hanterar hur ett textstycke skall se ut. Nedan visas exempel på några av de instruktioner som finns att tillgå:

Instruktioner för positionering av text		
Instruktion	Värde	Förklaring
display	block, inline list-item, none	Anger hur ett element skall visas. Är värdet 'block', kommer varje nytt element att börja på ny rad, 'inline' förhindrar ny radmatning. Om 'list-item' anges, är elementet tänkt att ingå i en lista, vid 'none' visas inte elementet.
margin-top margin-bottom margin-left margin-right	En giltig längd-enhet, exempelvis; cm, pt, px	Används för att relatera olika elements avstånd till varandra. Exempelvis en 'titel' kan ha 15pt mellanrum mellan sig och 'brödtext'.
text-indent	Som för 'margin' eller % av blockets längd	Styr indrag på den första raden text i ett element.

Instruktioner för formatering av text		
Instruktion	Värde	Förklaring
font-family	Ett teckensnitt som stöds av verktyget som används.	Anger teckensnittet som skall användas, exempelvis; Verdana.
font-style	oblique, italic, normal	Anger utseende på text, exempelvis kursiv stil.
color	Ett värde för en färg i text eller kod. Exempelvis; green eller #FF0088	Anger färgen på texten.

Kontext och arv

Med hjälp av CSS kan man även formge sin information utifrån vilken kontext man befinner sig i. Med kontext menas den position ett element har i ett XML-dokument i förhållande till andra element. Antag att man har följande XML-dokument:

```
<anstalld>
  <foretagsuppgifter>
    <namn>Chokladfabriken</namn>
  </foretagsuppgifter>
  <personuppgifter>
    <namn>Pelle</namn>
  </personuppgifter>
</anstalld>
```

Om <namn> representerar företaget vill vi att bakgrundsfärgen skall vara röd, annars skall den vara grön. Följande regler löser detta:

```
namn {background:green;}
foretagsuppgifter namn {background:red;}
```

Den första regeln säger att när ett element med namnet 'namn' hittas, så skall bakgrunden vara grön. Regel nummer två säger att om ett element 'namn' är barn till elementet 'foretagsuppgifter', så skall bakgrunden vara röd.

En annan egenskap hos CSS som är värdefull när det gäller underhåll av stora formatmallar är att de olika reglerna ärver av varandra. Tänkandet bakom denna idé är objektorienterat, det vill säga en regel utför dels sina egna instruktioner och dels de som finns definierade högre upp i hierarkin. Vi skapar en uppsättning regler utifrån ovanstående XML-dokument:

```
anstalld           {font-size:14pt;}
foretagsuppgifter {background:red;}
personuppgifter   {font-size:8pt;}
namn               {font-weight:bold;}
```

Elementet 'namn' kommer att ärva det utseende som är definierat högre upp i hierarkin. I fallet då 'foretagsuppgifter' är dess förälder blir textstorleken 14 punkter mot en röd bakgrund. I det andra fallet finner vi en krock. Textstorleken finns definierad på två ställen i hierarkin. Principen är då att den regel som ligger *närmast uppåt* i hierarkin gäller, det vill säga 8 pt.

Regler för företräde

Med CSS kan man även skapa formateringsregler på tre olika nivåer. Nivåerna kan närmast liknas vid att man har regler i ett externt stylesheet, man har regler på dokumentnivå och man har regler på elementnivå. Precis som i fallet med arv så träder de regler som ligger närmast ett enskilt element i kraft. Sålunda gäller reglerna i ordningen: Extern, dokument och slutligen element.

På detta sätt kan man skapa ett externt stylesheet med regler för en stor mängd dokument. Behöver man ett något annorlunda utseende på några av dem lägger man till regler för dessa på dokumentnivå. Skulle man behöva tränga ner på enskilda ord i dokumentet kan man slutligen även skapa regler på elementnivå. Skapar man sina formateringsregler på detta sätt, gör man det enklare för sig att underhålla sina dokument.

XSLT

XSLT står för Exstensible Stylesheet Language: Transformations. Denna rekommendation är till för att man skall kunna konvertera strukturen i ett XML-dokument till en presentationsstruktur eller till en annan informationsstruktur. Den struktur som blir resultatet av en transformering med hjälp av XSLT kan vara av formatet HTML, RTF eller varför inte ett annat XML dokument.

I de flesta fall skapar man även ett utseende på vägen med hjälp av exempelvis CSS, XSL Formatting Objects eller HTML.

Varför transformera?

Det finns huvudsakligen två skäl till varför man vill transformera, det vill säga omstrukturera sina dokument:

1. Från den befintliga informationsstrukturen vill man skapa en struktur som är mer lämpad för presentation.
2. Behovet av att konvertera information mellan två eller flera färdiga data-strukturer, exempelvis mellan två databaser.

Ett exempel på information som har behov av att omstruktureras för presentation är när man fyller i en tidrapport. Antag att vi har följande struktur:

```
<tidrapport>
  <dag datum="2000-06-07" veckodag="Måndag">
    <projekt>
      <namn>Transformeringsprojekt</namn>
      <timpris>1000</timpris>
      <antal_timmar>3</antal_timmar>
    </projekt>
    <utbildning>
      <namn>XSL-utbildning</namn>
      <antal_timmar>5</antal_timmar>
    </utbildning>
  </dag>
  <dag datum="2000-06-08" veckodag="Tisdag">
    <projekt>
      <namn>Transformeringsprojekt</namn>
      <timpris>1000</timpris>
      <antal_timmar>3</antal_timmar>
    </projekt>
    <utbildning>
      <namn>XSL-utbildning</namn>
      <antal_timmar>5</antal_timmar>
    </utbildning>
  </dag>
```

```

    <projekt>
      <namn>Annat projekt</namn>
      <timpris>1000</timpris>
      <antal_timmar>4</antal_timmar>
    </projekt>
  </dag>
</tidrapport>

```

Om vi presenterar informationen i den ordning den är skapad, får vi svårt att överblicka den och det är svårt att sammanställa den.

Veckodag	Benämning	Namn	Timmar
Måndag	Projekt:	Transformeringsprojekt	3
	Utbildning:	XSL-utbildning	5
Tisdag	Projekt:	Transformeringsprojekt	3
	Utbildning:	XSL-utbildning	5
	Projekt:	Annat projekt	4

Väljer man istället att skapa en presentationsstruktur kan vi få en mer lättöverskådlig bild.

	Namn	Måndag	Tisdag
Projekt:	Transformeringsprojekt	3	3
	Annat projekt		4
Utbildning:	XSL-utbildning	5	5

I det andra fallet handlar det om utbyte av information. Behovet kan uppstå om två skilda organisationer behöver utbyta data. Antag att företag A har följande struktur på sin data:

```
<livsmedel typ="godis">choklad</livsmedel>
```

Medan företag B har:

```
<godis>choklad</godis>
```

Om A vill utväxla data med B måste han konvertera till dennes format och vice versa. Detta måste ske för att respektive företags datorsystem skall kunna förstå hur data som skickas till dem skall hanteras. Datorsystemen bygger på den informationsstruktur man skapat, sedan må det vara i form av databaser eller XML.

XSLT:s olika delar

När man skapar en XSLT-formatmall skapar man ett XML-dokument. Följaktligen måste ett XSLT dokument följa alla de regler som gäller för XML. Exempelvis måste det finnas ett rotelement, element får ej överlappa varandra och så vidare.

Ett XSLT dokument består av ett antal delar, varav nedanstående kan ses som viktigast:

- dokumentets rotelement
- mallregler för elementen i XML-dokumentet
- element för att bestämma utformat.

Rotelement

Rotelementet i ett XSLT-dokument är alltid `<xsl:stylesheet>`. Vi tar en närmare titt på hur det ser ut:

```
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
```

Stylesheet är namnet på själva elementet och `xsl:` som föregår är den namnrymd (xmlns = xml name space) vars syntax man använder. Denna namnrymd, ”xmlns:xsl=”http....”, betyder att man följer de regler som är uppsatta för XSLT av W3C. När man sedan använder en term från dessa regler skriver man prefixet `xsl:` före så att den XSLT-parser man använder vet hur den skall tolka olika saker i formatmallen. Man behöver inte använda `xsl:` som prefix, utan man kan lika gärna skriva: `xmlns:hej="http://www.w3.org/1999/XSL/Transform"`. Gör man det skriver man sedan `<hej:stylesheet>` istället.

Det andra attributet säger bara vilken version av rekommendationen man använder. Idag finns det bara `version="1.0"`, men framtiden kommer troligen att ge oss fler.

Mallregler

Det som används när man skall mappa information ifrån sitt källdokument till en ny struktur är så kallade mallregler, som skrivs `<xsl:template>`. Det är en mallregel som är obligatorisk för varje XSLT-dokument:

```
<xsl:template match="/">
```

Denna regel mappas mot XML-dokumentets rotelement, det vill säga det element som är förälder till processinstruktionerna och dokumentelementet. Attributet 'match' anger vad man mappar mot i XML-dokumentet.

Det är inne i detta element som bearbetningen av källdokumentet börjar. Härifrån anropar man till exempel andra mallregler man skapat. Man kan likna det vid ett huvudprogram, från vilket man anropar procedurer och funktioner.

Övriga mallregler som man skapar relaterar till elementen i XML-dokumentet.

XML-dokument

```
<person>
  <namn>Pelle</namn>
  <adress>Lund</adress>
</person>
```

XSLT-dokument

```
<xsl:template match="person">
  <xsl:template match="namn">
  <xsl:template match="adress">
```



Vill man använda sig av de mallregler man skapat använder man sig av <xsl:apply-templates>. I attributet 'select' anger man vilken mallregel man vill tillämpa. Det skulle kunna se ut på följande sätt:

```
<xsl:template match="person">
  <xsl:apply-templates select="namn"/>
  <xsl:apply-templates select="adress"/>
</xsl:template>
```

Utformat

Det finns en möjlighet att bestämma vilket format man vill ha ut sin transformering i. De tre som stöds av rekommendationen är XML, HTML och text. Det märkord i XSLT som används för detta är <xsl:output>. Detta element måste finnas på toppnivån i XSLT dokumentet, dvs. dess enda förälder kan vara <xsl:stylesheet>. Vilket utformat man vill ha bestäms med hjälp av attributet 'method', <xsl:output method="html">.

Övriga element

Utöver de element som redan nämnts finns det ett antal till, varav några behandlas nedan:

Elementnamn	Attribut	Förklaring
<xsl:value-of>	select disable-output-escaping	Detta element används för att lägga texten i ett element till resultatträdet. Med 'select' väljer man det element vars text man vill åt och 'disable-output-escaping' styr över hur vissa special-tecken presenteras i resultatet.
<xsl:for-each>	select	Detta element används för att traversera en lista av element. Med 'select' väljer man ut en lista av element.
<xsl:if>	test	Detta element medger styrning över resultatet med hjälp av olika villkor. Attributet 'test' tolkar om resultatet av ett villkor är sant eller falskt.
<xsl:sort>	select data-type order	Detta element används för att sortera element. Med 'select' väljer man vilket värde man sorterar efter. Attributet 'data-type' anger om värdet skall vara exempelvis text eller nummer. Med 'order' bestäms om man sorterar i stigande eller fallande ordning.

XSL Formatting Objects

Uppbyggnaden följer den som är satt för ett XSLT-dokument. Man kan säga att de olika instruktioner som XSL FO erbjuder kommer att infogas i ett XSLT-dokument och därigenom formge informationen. Vad man behöver göra är att definiera en namnrymd även för XSL FO. Den namnrymd man vanligtvis ser är 'fo'. En deklaration blir sålunda:

```
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fo="http://www.w3.org/1999/XSL/Format"
  version="1.0">
```

Vid författandet av denna skrift var XSL FO fortfarande inte en antagen rekommendation hos W3C, vilket gör att det fortfarande kan ske förändringar med den.

Formatering

De formateringsinstruktioner som kommer att finnas för XSL FO är i mångt och mycket lika de som finns för CSS. Det beror på att idéerna som ligger till grund för XSL FO bygger på CSS och DSSSL. Syntaxen är sålunda i många fall rätt lik rekommendationerna emellan.

```
<xsl:template match="FORETAGSUPPGIFTER">
  <fo:block background-color="red">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>
```

Exemplet ovan är samma regel som användes i avsnittet om CSS men här uttryckt i XSL FO. Regler som skapas med XSL FO kommer bland annat att förekomma i de mallregler som presenterades i avsnittet om XSLT och så är fallet även ovan.

Sidhantering

En viktig del av XSL FO är möjligheten att hantera sekvenser av sidor, något som man har behov av om mediet som används för presentation är papper.

Exempelvis kan man skapa regler för hur man skapar och hanterar sidor. Detta kan innebära att man skapar sidhuvuden och sidfötter, att man specificer-

rar antal kolumner samt placering och utseende på. Detta används sedan som mall när man skall placera in information.

Block

Ett block kan sägas vara en grupp av objekt, såsom stycken av text eller bilder. Typiskt kan man säga att block är olika delar av en sida. Det finns exempelvis en instruktion som heter `<fo:block>` som används för att formatera paragrafer, titlar och dylikt. Med denna instruktion bestämmer man färg, indrag med mera.

Case

Användningsområden

Traditionellt sett har den typ av strukturerad informationshantering som XML står för använts till att författa teknisk och strategisk information. Då i formen av SGML, som använder bland annat DSSSL för att formge datan. XML kan också användas inom detta område och teknikerna vi presenterat här är fullgoda för att skapa ett utseende på denna typ av information.

Vad gäller publicering på papper, så kommer antagligen de båda XSL standarderna att vara förhärskande. Anledningen till detta är att CSS ännu inte har de egenskaper som krävs för att på ett tillräckligt bra sätt presentera information på papper. CSS är skapat för att användas på Internet, medan XSL FO är skapat för att vara medieoberoende.

Förutom enbart publicering, kan man använda XSLT för att konvertera till andra format. Det kan vara till olika presentationsformat, såsom HTML eller RTF, eller till en annan XML-struktur för utbyte av data.

Ser man på de olika presentationsformat som finns, så är HTML idag ännu ett av de vanligaste format som används när man publicerar sin XML data. En vanlig kombination är att man skapar en formatmall som transformerar informationen till HTML, som använder sig av CSS för att få utökade möjligheter att styra presentation och layout. WML (Wireless Markup Language) är ett annat alltmer förekommande alternativ när det gäller att publicera data. Det är WML som används för att märka upp data när man använder WAP och även här kan XSLT användas för konvertering från XML.

XSLT ger också möjligheter att på ett enkelt sätt utbyta data mellan olika system. Dess roll blir att konvertera data så att den blir förståelig för det system som skall ta emot information. Detta är något som är av stor vikt då organisationer i allt större utsträckning kommunicerar på elektronisk väg. Använder man XML som mellanlagring innan informationen kommer in i systemen, kan man med hjälp av XSLT skapa förutsättningar för informationsutbyte. Detta är ett sätt att komma runt att olika organisationer har olika plattformar, men även ett sätt att konvertera mellan olika informationsstrukturer. I det här fallet har man ingen användning för varje sig CSS eller XSL FO, då data i det här läget ej skall presenteras utan bara förmedlas.

Standarderna i praktiken

Nedan kommer ett komplett exempel på hur man kan presentera data med hjälp av XSLT och CSS. I exemplet har vi en XML-fil med data som konverteras till

HTML och där CSS används för att skapa utseendet. I exemplet ges det även prov på hur XPath (se skriften om länkning) används tillsammans med XSLT för att ställa frågor till ett XML-träd.

XML-filen som används ser ut så här:

```
<?xml version="1.0"?>
<?xml:stylesheet href="bilregister.xml" type="text/xsl"?>
  <bilregister>
    <bil farg="gul">Volvo</bil>
    <bil farg="vit">Ford</bil>
    <bil farg="gul">Mercedes</bil>
    .
    .
  </bilregister>
```

Formatmallen skall presentera data på två sätt. I det första fallet presenteras de olika bilarna som en sorterad och filtrerad lista med hjälp av `<xsl:for-each>`, dels i den sekvens de är skrivna i med hjälp av `<xsl:template>` och `<xsl:apply-template>`.

Det första som måste göras är att skapa de processinstruktioner som talar om för programvaran hur den skall tolka filen och den obligatoriska mallregeln som klargör var skapandet av själva resultatet, i det här fallet en HTML-fil. Vi anger också vilket utformat vi vill ha.

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output type="html" indent="yes"/>
  <xsl:template match="/">
    <HTML>
      <body>
      </body>
    </HTML>
  </xsl:template>
</xsl:stylesheet>
```

Nästa steg blir att skapa de mallregler som skall svara mot elementen i källdokumentet. Vi skapar också HTML kod för att lägga de olika bilarna i en lista.

```
<xsl:template match="bilregister">
  <ul><xsl:apply-templates select="bil"/></ul>
</xsl:template>
```

Man skulle kunna skriva bara `<xsl:apply-templates/>` om man vill. Vad som händer då är att XSLT letar efter om det finns mallregler för det aktuella elementets barn och använder dessa.

```
<xsl:template match="bil">
  <li><xsl:value-of select="."/></li>
</xsl:template>
```

Punkten som förekommer i uttrycket betyder att det värde som finns för den aktuella noden skall användas i resultatet. I vårt exempel kommer mallregeln för bil att användas tre gånger då vi har tre bilar i vårt källdokument. För varje gång hämtas det aktuella värdet.

Nu vill vi formge texten med hjälp av CSS och detta gör man genom att skapa CSS regler i sin formatmall. Vad man faktiskt gör är att man bygger på resultatet, HTML dokumentet, med denna information. Vi måste också lägga till ett anrop till den första mallregel vi vill använda, det vill säga för bilregis-
ter.

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output type="html" indent="yes"/>
  <xsl:template match="/">
    <HTML>
      <style>
        .bilregister {font-family:verdana;
                      font-size:16pt;}
        .bil         {color:red;}
      </style>
      <body>
        <xsl:apply-templates select="bilregister"/>
      </body>
    </HTML>
  </xsl:template>

  <xsl:template match="bilregister">
    <ul><span class="bilregister"><xsl:apply-templates select="bil"/></span></ul>
  </xsl:template>

  <xsl:template match="bil">
    <li><span class="bil"><xsl:value-of select="."/></span></li>
  </xsl:template>
</xsl:stylesheet>
```

Till skillnad från om vi hade använt CSS direkt på ett XML-dokument behöver man här inte döpa sina regler efter de elementnamn man definierat i källdokumentet. Det är viktigt att veta att när vi använder CSS i vårt exempel så är det resultatet, det vill säga HTML, inte källdokumentet vi formger. Punkten som föregår de olika CSS reglerna innebär att man skapar en klass i HTML. Denna klass kan man sedan ropa på från i sitt html-dokument.

Nu återstår att skapa en lista där vi sorterar bilarna i stigande ordning och att vi utelämnar de bilar som är vita.

```
<xsl:for-each select="bilregister/bil[not(@farg='vit')]">
  <xsl:sort order="ascending"/>
  <xsl:apply-templates select="."/>
</xsl:for-each>
```

Vi börjar med att ange att vi vill traversera listan med bilar. I select-satsen har vi även angett ett villkor för vilka bilar som skall skrivas ut, nämligen de som inte har ett attribut 'farg' med värdet 'vit'. Det är detta uttryck som är XPath och det är frekvent förekommande i XSLT-dokument. Vidare säger vi att de olika bilarna skall sorteras i stigande ordning, det är <xsl:sort> som gör det och slutligen anger vi att vi vill använda mallregeln för det aktuella elementet. I det här fallet är det 'bil' som är aktuellt. Slutligen ser vårt XSLT-dokument som följer:

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output type="html" indent="yes"/>
  <xsl:template match="/">
    <HTML>
      <style>
        .bilregister {font-family:verdana;
                      font-size:16pt;}
        .bil         {color:red;}
      </style>
      <body>
        <xsl:apply-templates select="bilregister"/>
        <xsl:for-each select="bilregister/bil[not(@farg='vit')]">
          <xsl:sort order="ascending"/>
          <xsl:apply-templates select="."/>
        </xsl:for-each>
      </body>
    </HTML>
  </xsl:template>

  <xsl:template match="bilregister">
```

```
<ul><span class="bilregister"><xsl:apply-templates select="bil"/></span></ul>
</xsl:template>

<xsl:template match="bil">
  <li><span class="bil"><xsl:value-of select="."/></span></li>
</xsl:template>
</xsl:stylesheet>
```

Detta exempel är skapat enligt rekommendation för XSLT 1.0 och kan användas med XSLT processorer som följer denna. Exempelvis kan Apaches 'Xalan' eller Microsofts 'MSXML3' parsers användas.

Framtiden

Framtiden för användandet av formatmallar ser ljus ut. Medvetenheten om fördelarna med formatmallar sprids allt mer och mer och kommer troligtvis att bli en självklarhet vid presentation av större informationsmängder.

För närvarande håller W3C på att arbeta fram en ny rekommendation för CSS, CSS level 3. Man håller på att modularisera CSS, vilket kommer att innebära att det kommer att finnas små mini-rekommendationer. Dessa kommer att stödja ett väl avgränsat ämne såsom 'färger och bakgrund'. Ett av skälen att göra på detta sätt är att utvecklare lättare skall kunna välja vilka delar av CSS deras verktyg skall stödja. Man kan också inrikta sig på att implementera en modul åt gången.

Precis som skett med många andra rekommendationer kommer vi troligtvis att få se uppdateringar av XSLT. Det dyker hela tiden upp nya krav och idéer hur man kan använda sig av XSLT och detta bäddar för utökning av rekommendationen till förmån för dessa tankar.

XSL FO är sedan 21 november 2000 Candidate Recommendation. Vi bör ha rekommendationen här inom en snar framtid. Precis som med CSS och XSLT kommer XSL FO att bli föremål för förändringar. Vilka dessa blir återstår att se.

Det ser positivt ut när det gäller verktygsstöd för formatmallar. Det finns redan ett tiotal verktyg som stödjer XSLT, däribland MSXML3 som används tillsammans med Internet Explorer 5. CSS har ett visst stöd i våra vanligaste webbläsare och förhoppningsvis blir detta stöd utökat. För XSL FO finns det något enstaka verktyg som stödjer ett tidigt utkast av den tilltänkta rekommendationen. Att det är så få beror troligtvis på att mjukvaruföretagen vill vänta med sin utveckling till dess att XSL FO anses stabilt.

Statskontorets publikationer för 1999 och 2000

1999

- 99:1 Statskontorets publikationer utgivna under åren 1996-1998. GRATIS
- 99:2 Skiftet till år 2000 – läget i myndigheter. Lägesrapport 7. Slut finns i PDF.
- 99:3 E-post i förvaltningen – En rättslig översikt. 75:-
- 99:4 Samordning och styrning av smittskyddsverksamheten. GRATIS
- 99:5 Informationskampanjen Starta eget tillsammans. GRATIS
- 99:6 Utvecklingsgarantin för ungdomar – det första året. GRATIS
- 99:7 Regelförenkling i EU och Sverige. GRATIS
- 99:8 Användning av otraditionella medel i Skåne åren 1998 och 1999. GRATIS
- 99:9 Myndigheternas förutsättningar för deltagande i ett nytt offentligt rättsinformationssystem. GRATIS
- 99:10 Skiftet till år 2000 – Läget i myndigheter. Slut finns i PDF.
- 99:11 Läget i viktiga samhällsfunktioner. GRATIS
- 99:12 Tillstånd för näringsverksamhet. GRATIS
- 99:13 Intensifierat nordiskt samarbete – analys av det svenska styrsystemet. GRATIS
- 99:14 E-post i skolan. 150:-
- 99:15 Staten i omvandling 1999. 175:-
- 99:15A The Swedish Central Government in Transition 1999. 50:-
- 99:16 XML-verktyg. Ej utk.
- 99:17 Strukturer för hantering av certifikat och kryptonycklar i Sverige – Statskontorets förslag till vidare arbete. GRATIS
- 99:18 Hur mycket kostar Kunskapslyftet? – en analys av kommunernas kostnader och bidrag. GRATIS
- 99:19 Punktskatteorganisationen inom skatteförvaltningen. GRATIS
- 99:20 En samlad administration av lönegarantin. 100:-
- 99:21 Det viktiga valet av verkschef. En jämförande studie av verkschefsutnämningar. GRATIS

- 99:22 Skiftet till år 2000 läget i myndigheter. Lägesrapport 10.
Slut finns i PDF.
- 99:23 Enkla råd och tips vid E-posthantering. 75:-
- 99:24 Gamla län blir nya regioner? Slutrapport. GRATIS
- 99:25 Vad kostar ett tillstånd?
– förslag till enklare tillståndsgivning. GRATIS
- 99:26 SESAM Öppnade museisamlingarna? GRATIS
- 99:27 Läget i vissa samhällsfunktioner. Lägesrapport 11. GRATIS
- 99:28 Miljökrav vid offentlig upphandling
– samhällsekonomiska konsekvenser. GRATIS
- 99:29 Morgondagens kommission – en svensk angelägenhet. 75:-
- 99:29A Tomorrow´s Commission – a Swedish Concern. GRATIS
- 99:30 Säkerhet med elektronisk identifiering. 100:-
- 99:31 Tjänster för elektronisk identifiering i offentlig förvaltning. GRATIS
- 99:32 Vem ser efter arbetslöshetskassorna? GRATIS
- 99:33 Förslag till uppföljning av de transportpolitiska målen. GRATIS
- 99:34 Konvergens mellan tele och data – en orientering. 100:-.
- 99:35 Skiftet till år 2000 – Läget i myndigheterna. Lägesrapport 12. GRATIS
- 99:36 Läget i viktiga samhällsfunktioner – Skiftet till år 2000.
Lägesrapport 13. GRATIS
- 99:37 IT-kostnader. Förstudie. 100:-
- 99:38 Organisations-, styr- och verksamhetsformer i kommuner och lands-
ting. GRATIS
- 99:39 Elektronisk upphandling under tröskelvärdena. GRATIS
- 99:40 Informationstjänster i fokus. 85:-
- 99:41 Från underutnyttjande av användarboom Online/webbtjänster,
CD-rom och betalsamtal i Sverige 1997 och 1998. 100 kr
- 99:42 Interaktiva medier – framtidsbransch med intelligens? 100 kr
- 99:43 Investeringar i digital information. GRATIS
- 99:44 Efterfrågan på tjänster via Internet. GRATIS
- 99:45 Förstudie – IP-telefoni. 100:-
- 99:46 Förstudie – Centrexlösningar. 100:-
- 99:47 Mobiltelefoni – en förstudie. 100:-
- 99:48 UTGÅTT

- 99:49 Läget i myndigheter och vissa samhällsfunktioner – skiftet till år 2000. GRATIS
- 99:50 Framtidens servicefunktioner – en förstudie. 100:-
- 99:51 Att konsultera medborgarna. GRATIS
- 99:52 Hur kan Sveriges möjligheter att påverka EU:s samarbete förbättras? 130:-
- 99:53 Mittutvärderingar av strukturfonderna. GRATIS
- 99:53A Mittutvärderingar av strukturfonderna. Bilagedel. GRATIS

2000

- 00:1 Effektivare kustbevakning. 100:-
- 00:2 Compazisons of Character Sets – Utgåva 2. GRATIS
- 00:3 Öppenhet och insyn i myndigheternas EU-arbete. 75:-
- 00:4 Postrummet – ett sätt att öka säkerheten i e-post. 100:-
- 00:5 Intelligent tjänster och elektroniska blanketter. 150:-
- 00:6 Sverige i världen – en utvärdering av svenskt deltagande i några internationella mellanstatliga organisationer. 150:-
- 00:7 Infrastruktur för säker elektronisk överföring till, från och inom statsförvaltningen. 100:-
- 00:8 Svenska myndigheter och euron – gemensamma frågor. GRATIS
- 00:9 Styrelser med fullt ansvar. 100:-
- 00:10 Intrångsdetekteringssystem
- 00:11 Statliga myndigheters rutiner för hantering av programvarulicenser. GRATIS
- 00:12 Telefonitrafik – vägledning och råd. GRATIS
- 00:13 SHS – en del av infrastrukturen för samhällets elektroniska tjänster. 150:-
- 00:14 Efter skiftet till år 2000, CD-rom. GRATIS
- 00:15 Staten i omvandling 2000.
- 00:16 Staten som kommersiell aktör. Del I Huvudrapport. 150:-
- 00:16A Staten som kommersiell aktör. Del II Fallstudier. GRATIS
- 00:17 Utvärdering och politik. GRATIS
- 00:18 Uppföljning och utvärdering av det regionala utvecklingsarbetet En rapport till den parlamentariska regionkommittén (PARK). GRATIS

- 00:19 Resultatredovisning och resultatbedömning i budgetpropositionen år 2000. GRATIS
- 00:20 Fem år i EU – en utvärdering om statsförvaltningens medverkan i EU-samarbetet. Huvudrapport. 150:-.
- 00:20A EU-arbetet i Sverige, Danmark och Finland. Appendix I. GRATIS
- 00:20B Fallstudier av tre EU-intensiva plitikområden. Appendix II. GRATIS
- 00:20C Genomförandekommittéer. En expendstudie om svenska departements- och myndigheters ansvar från antagande av gemenskapslagstiftning. Appendix III. GRATIS
- 00:21 24-timmarsmyndighet. Förslag till kriterier för statlig elektronisk förvaltning i medborgarnas tjänst. 100:-
- 00:22 Policymätning ett viktigt stöd för en bättre telefoni. 100:-
- 00:23 Administrationskostnader för regionala strukturfondsprogram. GRATIS
- 00:24 Slutredovisning – myndigheternas arbete med införandet av rättsinformationssystemet. GRATIS
- 00:25 En översyn av avgiftssystemet för inspektionen för strategiska produkter. GRATIS
- 00:26 Värmyndighet för nämndmyndigheter – utredning av förutsättningarna för utlokalisering av kanslistöd. GRATIS
- 00:27 Svenskundervisning för invandrare (sfi) – egen skolform eller del av komvux? GRATIS
- 00:28 Vinstandelsstiftelser. GRATIS
- 00:29 Att spara växel pengar. 100:-
- 00:30 Vad är XML? 125:-
- 00:31 Vad är XML Familjen? 125:-
- 00:32 Vad är XML Schema? 125:-
- 00:33 Vad är XML Formatmallar? 125:-
- 00:34 Vad är XML Länkar? 125:-
- 00:35 Vad är Document Object Model (DOM)? 125:-
- 00:36 Vad är XML Query Language? 125:-
- 00:37 Nummerinformation och katalogtjänster. 100:-
- 00:38 Vägledning – ramavtal för programvaror och tjänster. 100:-
(*för stat, kommun och landsting) *GRATIS
- 00:39 Administration inom Regeringskansliet. Förstudie av Förvaltningsavdelningen. GRATIS

- 00:40 Elektroniska signaturer och elektronisk identifiering för myndigheters e-tjänst. 100:-
- 00:41 The 24/7 Agency – Criteria for 24/7 Agencies in the Networked Public Administration. GRATIS
- 00:42 Utlokalisering av sprängämnesinspektionen – Sammanslagning med Räddningsverket? 100:-
- 00:43 Förvaltningens utveckling – förvaltningspolitikens genomslag. GRATIS
- 00:44 Vägledning om ramavtal för dokumenthanteringssystem (DHS). 100:- (GRATIS för offentlig förvaltning)
- 00:45 Kan rättsväsendets information samordnas? – En utvärdering. GRATIS
- 00:46 Med världen i sikte. En studie av Sveriges internationella försvarsmaterielsamarbete. 100:-
- 00:47 Översyn av verksförordningen. En förstudie. GRATIS
- 00:48 Att verka i EU. En studie av hur medlemsländerna deltar i EG:s utvecklingssamarbete. GRATIS
- 00:49 Modernitet, effektivitet och förvaltningsförnyelse – en kartläggning och analys av Europeiska kommissionens vitbok om interna reformer. GRATIS



STATSKONTORET

Box 2280, 103 17 Stockholm.
Norra Riddarholmshamnen 1.
Telefon 08-454 46 00. Fax 08-791 89 72.
www.statskontoret.se