

GÖTEBORGS UNIVERSITET

Institutionen för informatik

## **TIDS- OCH PLATTFORMSOBEROENDE DOKUMENTLAGRING**

Shirley Saavedra Carlström  
Lars-Olof R Norling  
Uppsats 10 p VT 1998  
Handledare: Kjell Engberg

Det finns ett behov idag i företag/organisationer att kunna hantera sina dokument plattformsoberoende. De vill kunna använda sina dokument på ett effektivt sätt med hjälp av IT-tekniken. Bland de språk som erbjuder denna möjligheten är SGML, Standard Generalized Markup Language, och XML, eXtensible Markup Language, som är beskrivningsspråk för att strukturera sina dokument.

Uppsatsen beskriver kort SGML/XML. Vidare tar uppsatsen upp några av de konsekvenser ett införande av SGML/XML medför för ett företag/en organisation.

Tillvägagångssättet har varit att studera SGML/XML samt att genomföra intervjuer med några utvalda företag/organisationer som arbetar med SGML. Insamlat material har sammanställts och bearbetats.

De slutsatser som dragits är att ett införande av SGML/XML i dagsläget är en lång process som är förenad med höga initialkostnader. Erfarenheterna är dock mest positiva hos de intervjuade företagen.

En viktig faktor för företag/organisationer att införa SGML/XML är deras krav på att kunna presentera sin dokumentation på olika typer av media. SGML/XML är väl lämpat för detta ändamål med den skillnaden att XML är bättre lämpad för web-publicering än SGML

# INNEHÅLLSFÖRTECKNING

<b>1 INTRODUKTION.....</b>	<b>2</b>
1.1 BAKGRUND .....	2
1.2 PROBLEMMRÅDE .....	2
1.3 SYFTE .....	2
1.4 FRÅGESTÄLLNING.....	2
1.5 METOD .....	3
1.6 AVGRÄNSNINGAR .....	3
<b>2 SGML .....</b>	<b>4</b>
2.1 BAKGRUND .....	4
2.1.1 SGMLs utveckling.....	4
2.2 VARFÖR UTVECKLADES SGML? .....	5
2.3 INTRODUKTION AV SGML .....	6
2.3.1 Ett beskrivningsspråk.....	6
2.3.2 Ett SGML-dokuments struktur.....	7
2.3.3 SGMLs uppbyggnad .....	8
2.3.4 SGML Declaration .....	9
2.3.5 Document Type Definition .....	9
2.3.6 Document Instance .....	10
2.3.7 SGML Parser .....	11
2.4 INFORMATIONSPUTBYTET.....	11
2.5 SGML I PRAKTIKEN.....	11
<b>3 XML .....</b>	<b>13</b>
3.1 VARFÖR UTVECKLADES XML?.....	13
3.2 INTRODUKTION AV XML .....	14
3.2.1 eXtensible Linking Language .....	15
3.2.2 eXtensible Style .....	16
3.2.3 XMLs mål.....	18
3.2.4 XMLs egenskaper.....	18
3.3 XMLS ANVÄNDNINGSSOMRÅDEN .....	19
3.3.1 XML och EDI .....	20
<b>4 SKILLNADER MELLAN SGML/XML.....</b>	<b>21</b>
<b>5 SGML/XML I FÖRETAGET .....</b>	<b>22</b>
5.1 VARFÖR HAR FÖRETAGEN VALT SGML/XML? .....	22
5.2 INFÖRANDE, KOSTNADER OCH VERKTYG.....	23
<b>6 SLUTSATSER.....</b>	<b>25</b>
<b>7 KÄLLFÖRTECKNING .....</b>	<b>27</b>

# **1 Introduktion**

## **1.1 Bakgrund**

Det har idag visat sig vara ett problem med att dokumentation inte är plattformsoberoende. Vi anser därför att det är intressant att undersöka hur företagen idag hanterar sin lagring av dokumentation/information.

Vår uppsats handlar om två plattformsoberoende beskrivningsspråk för dokumentstruktur, SGML, Standard Generalized Markup Language, och XML, eXtensible Markup Language. Vi kommer även att ta upp varför ett företag/en organisation satsar på att införa SGML/XML och vad det innebär.

Detta är ett område som det har varit en låg fokusering kring ända fram till de senaste två åren då XML började utvecklas och man började inse vilka möjligheter användandet av XML erbjuder. En av XMLs styrkor är att den är väl anpassad till den snabba utveckling på webben.

## **1.2 Problemområde**

Den snabba tekniska utvecklingen av ”apparater” kräver att dokumentationen över dessa är mer omfattande. Företag/organisationer vill på ett mer effektivt sätt skapa och använda sina dokument. Företag/organisationer vill även kunna presentera sina dokument på olika typer av media. I dag innebär det att dokumenten måste skapas i flera varianter.

Organisationer får en ökad arbetsbörda samt en ökad kostnad för att hantera sin dokumentation. Det är angeläget för dem att kunna effektivisera dokumentationshanteringen, för att få ned kostnaderna. Vår uppsats är fokuserad kring ett förslag till lösning av detta problem.

## **1.3 Syfte**

Syftet med uppsatsen är att kort beskriva de båda beskrivningsspråk SGML och XML samt att beskriva vilka effektivitetsmässiga och kostnadsmässiga konsekvenser, positiva och negativa, det medför för en organisation att införa ett beskrivningsspråk.

## **1.4 Frågeställning**

Vår frågeställning är:

Vad får det för konsekvenser för företag/organisationer att välja att införa ett beskrivningsspråk för dokumentstrukturer ur ett effektivitets- och kostnadsperspektiv?

## **1.5 Metod**

Vi har i vårt uppsatsarbete använt oss av en kvalitativ metodmodell. Vi har försökt att gå på djupet och skaffa oss riklig information om några få företag/organisationer. Vi har gjort några djuplodande intervjuer där vi har utgått från en frågemall, men även känt oss fria att gå utanför den och kunnat låta intervjun styra frågorna.

Vi har intervjuat representanter från företag som har implementerat SGML, samt konsulter som arbetar med att bistå företag/organisationer med expertkunskap. Vi har även intervjuat en representant för ett företag som inte har infört något beskrivningsspråk. Vi har försökt att skapa oss en helhetsbild av problemområdet.

Intervjuerna har varat från en till två timmar och de genomfördes under perioden april och maj 1998.

Vi har samlat material från böcker, artiklar och Internet, för att kunna skapa oss en bild av de båda beskrivningsspråken, deras styrkor och svagheter.

## **1.6 Avgränsningar**

I vår uppsats tar vi endast upp beskrivningsspråken, SGML och XML vilka är metaspråk samt HTML som även är ett statiskt språk.. Vi har inte undersökt om det finns andra beskrivningsspråk för dokumentstruktur.

## **2 SGML**

### **2.1 Bakgrund**

SGML, Standard Generalized Markup Language, härstammar från idéer från början av 1960-talet då man började att fundera kring tekniker för att märka upp och lagra informationen i ett dokument. Dokumentationen skulle bli oberoende av framtida ord- och textbehandlares krav. (Möller s 12)

1967 presenterade William Turncliffe, dåvarande ordförande för GCAs, Graphics Communications Association, Composition Committee, ett nytt tankesätt som handlade om separationen av ett dokument information från dess format som ett koncept för generell märkning. Detta nya tankesätt har legat till grund för utveckling av språk för att märka upp dokument strukturellt. (Charles F Goldfarb, 1996)

På IBM skapades ett par år senare GML, General Markup Language, av Charles F Goldfarb tillsammans med Edward Mosher och Raymond Lorie. Syftet var att få ett dokument att fungera i subsystem för texteditering, formattering och informationsåtervinning. GML står även för Goldfarb, Mosher och Loire. GML lanserades därefter av IBM som en produkt inom DCF, Dokument Composition Facility. (SGML User's Group)

ANSI, American National Standard Institute, startade 1978 ett projekt, lett av Goldfarb, för att skapa ett textbeskrivningsspråk som skulle vara baserat på GML. GCA bidrog med resurser och utvecklingen av SGML startade. (SGML User's Group, 1990)

#### **2.1.1 SGMLs utveckling**

Utvecklingsarbetet av SGML startade i USA och när ISO, den internationella standardiseringsorganisationen, började intressera sig för utvecklingen av SGML började också fler och fler synpunkter komma från Europa och Kanada. Från Europa kom kraven som tydligt påpekade nödvändigheten av att ta hänsyn till egna nationella tecken som t. ex. å, ä och ö i internationella sammanhang. Dessa krav gjorde att det inte gick att använda den amerikanska standardiserade teckenuppsättningen ASCII, American Standard Code for Information Interchange då den inte hade plats för nationella tecken. (Möller s 13)

Utvecklingen av SGML fortsatte och 1984 hade en rad omarbetningar gjorts som bland annat tog hänsyn till de europeiska kraven. SGML godkändes som en ISO-standard efter att först varit ute som ett ISO-Draft International Standard, DIS. Godkännandet publicerades den 15 oktober 1986 som ISO 8879-1986. När ISO publicerade standarden användes SGML. Sedermera har ett tillägg till standarden publicerats med beteckningen ISO 8879:1986/A1:1988 och 1991 var det dags för en femårsöversyn av standarden. Den befanns då vara en stabil standard. (Ibid)

Huvudsyftet med SGML är att kunna få dokument flyttbara mellan olika datorer och plattformar utan att behöva konvertera dokumenten. SGML skall göra informationsutbytet lättare, underlätta återanvändning av information, vara en mall för att

få struktur på informationen samt skapa förutsättningar för mer intelligenta informationslösningar. Framförallt handlar det om komplex information, information som skall leva länge samt vara flyttbar. (Vad är XML?)

## **2.2 Varför utvecklades SGML?**

Anders Möller beskriver i sin bok SGML – en introduktion till Standard Generalized Markup Language, hur ett dokument kan betraktas som en struktur av varierande element. En bok är till exempel organiserad i kapitel som innehåller avsnitt som innehåller stycken osv. En tidning är organiserad i artiklar som innehåller avsnitt, stycken mm. (Möller s 23)

Olika ordbehandlingsprogram behandlar dessa element på olika sätt. De använder sig av olika typsnitt, radavstånd mm. Styrkoderna för allt detta sparas tillsammans med dokumentet och när man sedan vill försöka läsa in dokumentet i en annan ordbehandlare än den som det skapades med, blir resultatet sällan bra. Ibland går det inte ens att läsa in ett dokument med en annan ordbehandlare. Dessa styrkoder är en kvarleva från den tid då dokument skickades till tryckerier för tryckning och man fick skicka med information om vad det skulle vara för typsnitt och storlek mm. (Ibid)

Trots försök från tillverkare av ord- och textbehandlingsprogram att tillhandahålla lösningar för olika datorfabrikat och operativsystem är det fortfarande ofta förknippat med problem att flytta ett dokument från t. ex. en IBM-kompatibel PC till en Sun arbetsstation eller en Macintosh eller vice versa. (Ibid)

I takt med den snabba teknologiska utvecklingen av allt mer avancerade produkter vi ser i dagens samhälle så kan vi också se att ett ökat behov av dokumentation samt att kunna presentera dokument via olika typer av media som CD, web-sidor mm. Ett behov av att kunna kombinera ljud, bilder osv. också har växt fram. Genom den utvecklingen ställer vi också högre krav på att dokument skall vara lättflyttad och plattformsoberoende. Vi vill kunna lagra dokument på en plats och i en plattform men ha möjligheten att lätt kunna flytta den till en annan plats och presentera den med hjälp av en annan plattform. (Ibid s 52)

Möller beskriver SGML som ett försök att lösa ovan beskrivna problem. Genom att enas om en internationell standard har man kommit en god bit på väg. En internationell standard innebär att den kan användas oberoende av vilket språk eller alfabet man använder. Inga regionala varianter finns utan den är lika över hela världen. För att detta skall fungera så måste standarden vara genomarbetad och omfattande. Alla dokument som följer standarden kontrolleras mot ett regelverk som gäller för just den dokumenttypen. SGML kontrollerar att dokumentet innehåller rätt element i rätt ordning. Olika regelverk måste alltså skapas för olika dokumenttyper eftersom de innehåller olika informationselement. (Ibid s 24)

## 2.3 Introduktion av SGML

### 2.3.1 Ett beskrivningsspråk

SGML är ett generellt beskrivningsspråk för dokumentstruktur. SGML saknar semantik och innehåller bara syntax. SGML kallas därför för ett metaspråk, en beskrivning av ett språk. För att kunna fungera som en standard så måste språket vara mycket noggrant specificerat. SGML är, som tidigare nämnts, baserat på regler som kan göra det möjligt att kontrollera förekomsten eller frånvaron av märkord<sup>1</sup> för de olika informationselementen i ett dokument. Möller s 26)

SGML använder generella koder (beskrivande märkning) för att beskriva strukturen i ett dokument i stället för process-specifika koder (procedurmärkning). Ett element som t. ex. stycke kan då kodas som <STYCKE> i stället för med en typografisk processkod som talar om att stycket skall vara skrivet med 12 punkter kursiv stil. Genom att använda generella koder så frigör man sig från speciella datorsystem och SGML kan därför klara av att ta hand om informationen även långt in i framtiden. (Ibid s 25) För att på ett enkelt sätt åskådliggöra skillnaderna mellan beskrivande märkning och procedurmärkning visas nedan en bild på beskrivande märkning samt en bild på procedurmärkning.

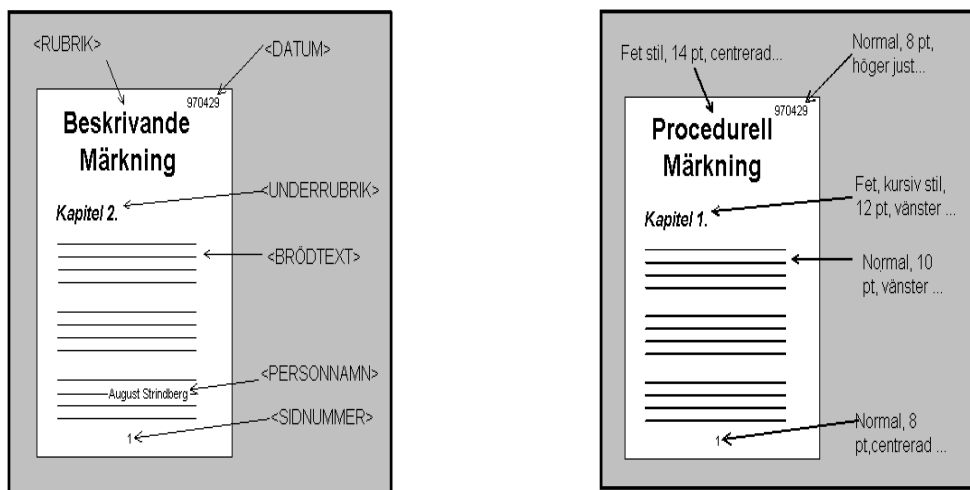


Fig. 1. Beskrivande märkning och procedurmärkning

Källa: Broady & Juliusson

Förutom att beskriva strukturen i ett dokument kan SGML användas för att beskriva ovanliga tecken som inte är normalt förekommande. Vidare kan det beskriva olika teckenuppsättningar och specialtecken för att det skall kunna fungera för olika språk. (Möller s 27)

<sup>1</sup> Markup Language. Märkord är benämningen för den tilläggsinformation som är inbakad i den text som ett dokument består av. Märkorden har två syften:

- att separera de logiska elementen i ett dokument
- att tala om vilken bearbetning som skall utföras på respektive element, som t. ex. att lägga på fetstil eller kursivering.

(Ibid s 28)

SGML använder sig av s k entiteter<sup>2</sup> för att beskriva symboler eller tecken som inte finns representerade. En textsträng av standardiserade tecken ur ASCII-kodtabellen utnyttjas för detta (Ibid s 27).

Anders Möller tar upp de två grundprinciper som behövs för att ett språk skall kunna kallas för ett generellt märkordspråk.

- Märkorden skall vara styrande och helt skilda från processinstruktioner för presentation av t ex layout och typografi. Märkorden innehåller både generella identifierare för dokumentets logiska informationselement och attribut som kan användas för att ge informationselementen ytterligare egenskaper. Alla processinstruktioner oavsett språk eller datorsystem ligger normalt utanför det egentliga dokumentet. (Ibid s 27)
- Märkorden skall vara formellt definierade för varje typ av dokument. I ett generellt märkordspråk definieras märkorden i en s. k. document type definition, DTD. Denna definition kan ses som en formell grammatik och innehåller en specifikation över de element<sup>3</sup> och attribut<sup>4</sup> som får förekomma i dokumentet samt i vilken ordning de får uppträda. Denna information gör det möjligt att avgöra om märkorden i ett dokument är korrekta och det är också möjligt att lägga till märkord som saknas. (Ibid s. 27)

Genom att använda generella märkord uppnår man betydande vinster jämfört med om man skulle använda sig av märkord innehållande processinstruktioner. Det finns ändå vissa brister hos de generella märkorden. De räcker inte till i alla situationer. Ibland är dokument mycket komplexa och har egenskaper som märkorden inte klarar av att beskriva, t. ex. när en figur med ett unikt namn, en unik identitet, vill läggas till dokumentet. I SGML löser man detta genom att utöka märkordet med ett identitetsmärke som pekar på det unika element som skall ingå i dokumentet. Detta förfaringsätt kan även användas till att peka på och länka in andra dokument med unika namn. (Ibid s 33)

### 2.3.2 Ett SGML-dokuments struktur

Det finns två olika strukturer kopplade till informationen om ett SGML-dokument. Dels är det vad som kallas för *logisk struktur*, som talar om informationens syfte och dels är det vad som kallas för *typografisk struktur*, som talar om hur informationen presenteras. Båda dessa strukturer är kopplade till samma informationsinnehåll. Ibland kallas den logiska strukturen även för innehållsstruktur. Den logiska strukturen är lätt att se. Informationselementen i den logiska strukturen kan t. ex. för en bok vara ett omslag med

---

<sup>2</sup> Från engelskans entity, kan i detta sammanhang närmast översättas med enhet eller saker. Det finns två typer av entiteter: generella- och parameterentiteter. En generell entitet kan t. ex. vara en förkortning av en längre fras, ex. `<!ENTITY MVH "Med vänlig hälsning">`. Istället för att skriva ut hela frasen räcker det att skriva MVH som sedan ersätts automatiskt med hela frasen. (Möller s 52) En parameterentitet används för att gruppera element eller attribut (Donald Broady och John Juliusson) Et exempel på en parameterentitet: `<!ENTITY % arbvecka "måndag, tisdag, onsdag,, torsdag, fredag">`

`<!ENTITY % helg "lördag, söndag">`  
`<!ELEMENT dubbelt -- (%arbvecka;, %helg;, %arbveacka;, %helg;)>` (Möller s 53)

<sup>3</sup> Element är en definition av ett objekt eller informationselement som skall ingå i den logiska strukturen i ett dokument. (Möller s 47)

<sup>4</sup> Ett attribut ger speciella egenskaper till det element som tilldelats attributet. (Ibid s 32)



titeln på boken, förlagsnamn, författare, copyright, förlagets logotyp, en sammanfattning, innehållsförteckning mm. (Ibid s 34)

Den typografiska strukturen talar om hur varje del av dokumentet skall presenteras, vilket typsnitt, teckenstorlek, radavstånd osv. Genom att man skiljer mellan dessa två olika strukturer i SGML så kan man använda flera typografiska strukturer till samma information, strukturerad enligt den logiska strukturen. Detta gör att man kan byta typografiskt utseende och layout när man vill utan att behöva arbeta om själva innehållet. SGML hanterar bara den information som är lagrad enligt den logiska strukturen. Hanteringen av typografin överläts åt det publiceringssystem som används för att presentera dokumentet. (Ibid s 34)

### 2.3.3 SGMLs uppbyggnad

SGML består av fyra skilda komponenter (se fig. 2) som gör det möjligt att lagra och representera ett dokument i SGML-format. Komponenterna som ingår är SGML Declaration, Document Type Definition, Document Instance och SGML Parser.

Ett SGML-dokument kan bestå av alla förekommande typer av information som kan lagras elektroniskt såsom grafik, ljud, bilder, video mm. Begränsningarna finns endast i de program eller hos de system som används för hantering av SGML-dokument. Det går att specificera hur systemet skall hantera främmande informationstyper. Detta görs i SGML-deklarationen och i DTDen. Det är helt möjligt att låta systemet starta upp en annan programvara som kan hantera den främmande informationstypen.

SGML klarar även av att referera till information som ligger utanför det egentliga dokumentet och till och med utanför det egna systemet. Detta beror på att SGML inte hanterar fysiska lagringsplatser utan lagringen sker virtuellt. Implementationen av de fysiska lagringsplatserna får skötas av respektive programvaruproducent. (Ibid s 37)

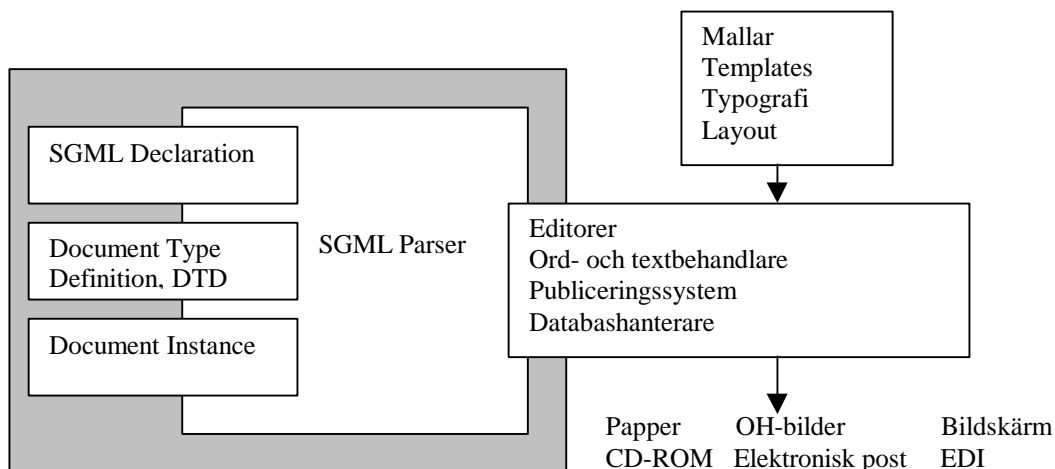


Fig. 2 De fyra komponenterna i SGML  
Källa: Anders Möller, SGML en introduktion, s 42.

### 2.3.4 SGML Declaration

SGML Declaration, innehåller information om t. ex. vilka teckenuppsättningar som är tillåtna, vilket ”concrete syntax” som får användas samt vilka värden som gäller för längden av ett elementnamn mm.

”Concrete syntax” beskriver vilka tecken som skall användas som avgränsare, hur många tecken ett märkord får bestå av. Genom att en applikation inte blir knuten till vissa tecken eller värden blir SGML applikationsoberoende. Varje system som hanterar SGML har också en SGML-deklaration kopplad till sig. Det skulle inte vara möjligt att hantera ett SGML-dokument utan denna deklaration. Fig. 3 visar ett exempel på en SGML-deklaration. (Ibid s 43)

```
<!SGML "ISO 8879-1986"
  CHAARSET
BASESET "ISO 646-1983//CHARSET
  International Reference Version (IRV)//ESC 2/8 4/0"
DESCSET 0
  9      UNUSED
  9      2      9
  11     2      UNUSED
  13     1      13
  14     18     UNUSED
  32     95     32
  127    1      UNUSED

CAPACITY PUBLIC "ISO 8879-1986//CAPACITY reference//EN"
SCOPE DOCUMENT
SYNTAX PUBLIC "ISO 8879-1986//SYNTAX Reference//EN"

FEATURES
MINIMIZE DATATAG NO OMITTAG YES RANK NO SHORTTAG NO
LINK SIMPLE NO IMPLICIT NO EXPLICIT NO
OTHER CONCUR NO SUBDOC NO FORMAL YES

APPINFO NONE>
```

Fig. 3 Exempel på en SGML-deklaration

Källa: Anders Möller, SGML – en introduktion

### 2.3.5 Document Type Definition

Document Type Definition, DTD är en viktig del i SGML. Där talar man om vilka element som får förekomma, vad de skall ha för märkord, i vilken ordning de får uppträda och vilka inbördes relationer som elementen skall ha. Här ligger även deklarationer som talar om vilka eventuella attribut som får förekomma för elementen samt deklarationer över alla olika typer av entiteter man vill ha. (Ibid s 36) Det är viktigt att man noga tänker igenom vilka märkord man vill ha. Det går att lägga till nya märkord men det går inte att ta bort gamla (Jesper Gunnarsson, Inzide).

DTDen innehåller alla regler som skall gälla för en viss typ av dokument och eventuella länkar till andra dokument. Man delar in dokument i olika dokumentationsklasser. En roman t. ex. tillhör en dokumentationsklass där regler över strukturen är definierade, hur titel, författarnamn, publiceringsdatum, kapitelrubrik brödtext etc får förekomma. I DTDen finns namnen för alla ingående informationselementen, deras ev. attribut och deras inbördes förhållanden. Dessutom finns här referenser till entiteter och andra typer av innehåll som kan förekomma i dokumentet. Fig. 4 visar ett strukturdiagram över en DTD och ett exempel på en DTD. (Broady & Juliusson)

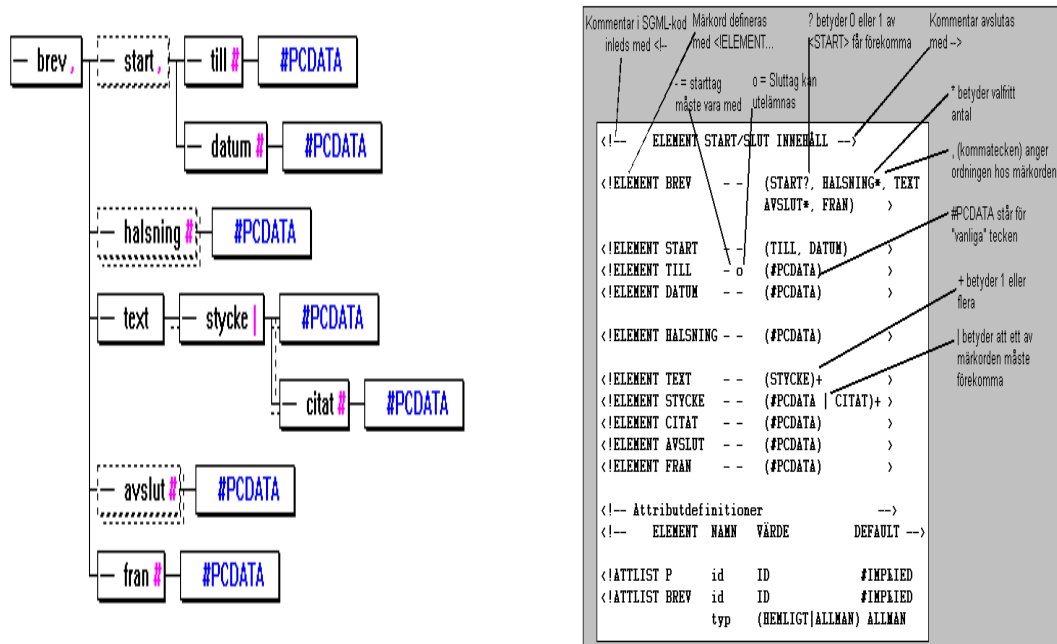


Fig. 4. Strukturdiagram över en DTD samt exempel på en DTD enligt strukturdiagrammet.  
Källa: Broady & Juliusson

### 2.3.6 Document Instance

Document Instance är själva SGML-dokumentet. Här återfinns själva texten tillsammans med märkorden med ASCII-tecken som går att flytta till vilken datorplattform som helst. Nedan visas ett exempel på ett SGML-dokument. (Broady & Juliusson)

```

<!DOCTYPE BREV SYSTEM "brev.dtd">
<BREV typ = "ALLMAN">
  <START>
    <TILL>TILL BENGT LIDFORSS</TILL>
    <DATUM>24 Febr. 1894.</DATUM>
  </START>

  <HALSNING>Käre Lidfors.</HALSNING>
  <TEXT>
    <STYCKE>Bitte öfverlemlna bref och titel till B.B. Men har Du tid
    kopiera brevet?</STYCKE>
    <STYCKE>D0 Stryka hugget i 1a Brevet på Atomvigten. Det andra
    hugget i 4 kan stå. Icke polemiser i noterna.</STYCKE>
    <STYCKE>Inför not i Paralipomena. <CITAT> Det senaste försöket göra
    diamanter i Paris, af kol och jern utföll i analogi med
    bordiamanter i Alumin, kolet transmuterades till Kisel och
    kemisten erhöU äfven kiselkristaller. Hvarifrån?</CITAT></STYCKE>
    <STYCKE>Du får inte strypa mina sättningar. Ordning eller icke
    ordning rör. Kisel, Al. Kol: det är frågan. Jag voterar för
    oordning.</STYCKE>
    <STYCKE>Öfvertala Steinschneider beskedligt ingå på bisakerna!
    </STYCKE>
  </TEXT>

  <AVSLUT>Jag dör af trötthet</AVSLUT>
  <AVSLUT>Vännern</AVSLUT>

  <FRAN>Aug Sg.</FRAN>
</BREV>

```

Fig. 5 Ett exempel på ett SGML-dokument.  
Källa:, Broady & Juliusson.

### **2.3.7 SGML Parser**

Den sista komponenten är Parsern. Det är ett program som kontrollerar om ett dokument stämmer med de strukturregler som finns i DTDen. Vidare kontrolleras att DTDen stämmer med SGML-deklarationen. (Möller s 45)

## **2.4 Informationsutbyte**

Ett argument för SGML är att ett SGML-dokument är flyttbart och plattformsoberoende. Detta är dock en sanning med modifikation. Vill man flytta ett SGML-dokument från en dator till en annan dator så är det, enligt Möller, inte tillräckligt att bara flytta över dokumentet. För att den mottagande datorn skall kunna tolka dokumentet måste man flytta med SGML-deklarationen, document-instancen och DTDen. Parsern behöver inte flyttas med eftersom den oftast är inbyggd i SGML-programmet eller också är den så hårt knuten till datorn att den inte kan flyttas med. Förmodligen har också det mottagande systemet en egen parser. Externa entiteter och eventuella bilder måste även de flyttas med eftersom de endast är länkade till dokumentet. (Ibid s 57)

För att vara säker på att dokumentet får samma layout på den mottagande datorn som i den dator där det konstruerat finns det ytterligare en komponent som bör flyttas med till den mottagande datorn. Det är en s. k. FOSI, Format Output Specification Instance. En FOSI är den komponent som bestämmer den layout och den typografi som ett dokument skall ha när det presenteras. FOSIn läser de ”taggar” som dokumentet har märkts upp med via DTDen. Används en annan FOSI får dokumentet en annan typografi och layout än vad som var tänkt från början. Själva textinnehållet är dock detsamma. Detta gör att ett SGML-dokument fortfarande kan läsas om hundra år även om layouten ser annorlunda ut. (Jesper Gunnarsson, Inzide)

## **2.5 SGML i praktiken**

Anders Möller skriver också om den tekniska utvecklingen och därmed det ökade behovet av dokumentation. Den tekniska utvecklingen ställer högre krav på dokumentation. Jämfört med bara för ca 25 år sedan är behovet ett helt annat. Då kunde det ofta räcka med en sprängskiss över en maskin för att en duktig mekaniker skulle kunna plocka isär och reparera den. Dagens tekniska produkter innehåller allt mer avancerad elektronik och smådatorer vilket innebär att drift- och underhållsmanualer, bruksanvisningar etc. måste innehålla mycket mer information än förr<sup>5</sup>. (Möller s 40)

Den tekniska dokumentationsmängden bedöms öka med ca en miljon (1.000.000) sidor per dygn om man räknar ihop hela världens produktion. Räknar man om dygnsproduktionen till årsbasis så blir det 364 miljoner sidor per år. Pappersåtgången för all denna information motsvarar ett kalhygge på ca 4 km<sup>2</sup> per dygn. Till detta tillkommer tillväxten inom den offentliga sektorn som är minst lika stor. Genom att använda SGML kan man minska behovet av pappersdokumentation. (Möller s 60)

---

<sup>5</sup>Som ett exempel på den enorma mängd dokumentation en produkt kan kräva kan vi se på en AXE-telefonväxel. En sådan levereras idag med 420 A4-pärmar med kunddokumentation. Till detta kommer all produktdokumentation inklusive kravspecifikation, ritningar beredningar etc. Den totala dokumentationsmängden för en AXE-växel omfattar då ca 4.200 pärmar. (Möller s 61)

Med SGML går det att producera, lagra, återsöka och presentera informationen på ett mer rationellt sätt.

Ett exempel är Svensk Handel och Tjänsteföretagen (SHT) som har som en av sina viktigaste uppgifter att informera sina medlemmar om lagar och avtal inom sin del av arbetsmarknaden. Tidigare gick informationen ut till medlemmarna som pärmar och utskick i form av papper. Nu har de börjat publicera sina pärmar på internet. De hade som krav en teknisk lösning som kunde administrera all information på ett ställe för både publicering på papper och web. Medlemsregistret skulle även kunna användas både för elektroniskt abonnemang och för pappersutskick. (Konsten att hålla 120 pärmar aktuella)

WM-data presenterade en lösning av problemet byggd på en databasapplikation för SGML, Information Manager från Texcel. I databasen lagras tre olika dokumenttyper; Avtal, Cirkulär och Pärmbroschyr. Varje element lagras för sig i databasen. Utifrån den grundinformation som finns lagrad i databasen kan man sedan skapa unika pärmar. När en pärm är färdig eller uppdaterad publiceras den på webben. SGML-filerna och länkarna konverteras till HTML-format. (Ibid)

Varje medlemsföretag har ett eget konto på web-servern och efter att ha loggat in får de tillgång till de pärmar de abonnerar på.

SHT har 20.000 medlemsföretag och från medlemsregistret hämtas regelbundet information om vilka pärmar företagen abonnerar på. Man räknar även med att i framtiden kunna skicka ut riktad e-mail om ändringar i avtal och lagar till de företag som valt att få sin information elektroniskt. Det tryckeri som trycker informationen kan hämta sina plocklistor för pärmar och original från samma databas via en ISDN-lina. (Ibid)

SGML Ger bl. a. följande fördelar: (Goossens, 1995)

- källdokumentets kvalitet förbättras
- dokument kan användas mer rationellt vilket resulterar i en förbättrad livscykel
- publiceringskostnaderna reduceras
- information kan lätt återanvändas vilket tillfogar dokumentet ett ökat värde

## 3 XML

### 3.1 Varför utvecklades XML?

XML, eXtensible Markup Language, utvecklades av en XML-grupp kallad XML Working Group<sup>6</sup> som bildades 1996. Gruppen leds av Jon Bosak från Sun Microsystems och deltagarna är medlemmar från XML gruppen organiserad av W3C. (W3C, 1997)

Den 10 februari 1998 frisläppte W3C XML 1.0 som en rekommendation. Redan innan XML frisläpptes hade många leverantörer utvecklat editorer för XML. (Gennusa, 1998)

Enligt en artikel av Neil Randall insåg man ganska tidigt att SGMLs struktur inte passade för web-publicering, eftersom SGML inte är snabbt och inte lätt att använda. Man behövde ett enklare system som man snabbt kunde lära sig. (Randall, 1997)

Då gjorde HTML, HyperText Markup Language, sitt intåg. HTML är lite mer än ett specificerat SGML-dokument. HTML är en SGML applikation som är lätt att lära sig och lätt att implementera och eftersom tidiga navigationsverktyg, s.k. web-läsare, stödde HTML blev det grunden för webben. Faktum är det att om SGML hade varit beskrivningsspråket på webben hade antagligen webben inte varit så populär. (Ibid)

Problemet med HTML är att språket är för enkelt och inte ”extensible” (sträckbart). HTML var bra i början när man skrev www-sidor som presenterade ett hypermediadokument, ett dokument som kan innehålla flera olika former av information (text, bilder, ljud, film...), och dessutom länkar mellan olika punkter i informationen. Men ganska snart upptäckte web-masters att HTML inte var tillräcklig när det gällde multimedia och design av web-sidor. Bilder med inbäddade hyperlänkar skapade nya problem som behövde lösas. Sedan dök det upp blinkande texter, tabeller, inramningar (frames på engelska) och dynamisk HTML. Varje gång något nytt dök upp stötte man på nya problem och varje gång uppkom ett behov att lägga till något mer i HTML. Man upptäckte då att språket ej var kompatibelt med olika web-läsare och att det fanns ett behov av att ha en standard (Ibid).

I många år har Microsoft lagt till egna ”tags” (märken) som bara är kompatibla med Internet Explorer och Netscape har gjort likadant, men som web-master har man inte haft den möjligheten att lägga till egna ”tags”. Utan tvivel, fortsätter Randall, har man kanske själv upplevt den frustrationen över HTMLs begränsningar och varje gång har man mottagit de nya ”tags” eller element som har blivit introducerade med glädje. Men för att designa bra web-sidor behöver man följaktligen fler metoder såsom Java, JavaScript, ASP<sup>7</sup> (Active Server Pages), och all den fortsatta utvecklingen av olika metoder ger HTML fler möjligheter att designa bra web-sidor. (Ibid)

---

<sup>6</sup> XML Working Group känd som SGML Editorial Review Board.

<sup>7</sup> ASP, är en serveranpassad scriptmiljö som är avsedd för byggandet av interaktiva web-applikationer.

Senaste utvecklingen är CSS<sup>8</sup>, Cascading Style Sheets, och dynamisk HTML<sup>9</sup> som ger en web-master ytterligare möjligheter till en mera komplett web-design, men dessa kompletteringar belyser bara det växande kravet och behovet av ett mera kraftfullt språk för design av web-sidor. (Ibid)

Randall ställer sig frågan om hur man skall kunna utnyttja SGMLs ”extensibility” (flexibilitet) utan att behålla den komplexitet som omgärdar SGMLs användning och som ingen vill ha samt hur man skall överbrygga gapet mellan SGML och HTML . Svaret är, skriver Randall, *XML*. (Ibid)

Enligt Alexandra Heymowska är den stora skillnaden mellan SGML, XML och HTML att HTML är ett statiskt språk med en fix uppsättning taggar medan SGML och XML är så kallade metaspråk. Detta innebär att man inte är begränsad till en fördefinierad uppsättning fasta taggar utan man har möjlighet att göra i princip vad som helst. (Heymowska,1998)

Erik Geijer skriver att XML inte längre är en lovande framtida teknik utan en officiell rekommendation. Förslaget som presenterades i december 1997 från bland andra Netscape, Microsoft, Sun och Adobe har antagits av ”World Wide Web Consortium” som en officiell rekommendation. XML är en mer avancerad teknik för att presentera strukturerade data. (Geijer, 1998)

### **3.2 Introduktion av XML**

XML är en delmängd av SGML, definierad som en extremt enkel SGML-dialekt. Därför kan XML-dokument också behandlas som ”rena” SGML-dokument. Med XML kan man alltså definiera egna element, attribut och hierarkiska konstruktioner för olika typer av dokument. Man har även fastslagit vilket teckenset XML skall använda. Teckensetet är en ISO-standard ISO 10646 (Unicode), en internationell teckenstandard som använder 31 binära siffror. Med denna kan man representera alla tecken inklusive exempelvis kinesiska. XML-dokument skall kunna visas i web-läsare direkt utan att först behöva konverteras till HTML. (W3C, 1998)

I XML finns någonting som heter ett *välformat dokument* (eng. well-formed). Kravet att XML-dokumenterna skall vara välformade har specificerats för att applikationer skall ha lätt att läsa XML-dokument och för att det skall vara lätt att distribuera XML-dokument i nätverk. (Ibid)

XML-familjen består av tre standardförslag:

- XML, eXtensible Markup Language.
- XLL, eXtensible Linking Language.
- XSL, eXtensible Style Language.

---

<sup>8</sup> CSS är ytterligare en utökning av HTML. Den ger webbdesignern möjlighet att bestämma profilen för sidans typografiska utseende.

<sup>9</sup> DHTML, ger möjlighet att ändra en web-sidas utseende och innehåll efter att den har laddats hem till användarens dator utan att ta kontakt med webservern.

Målet med dessa är att möjliggöra publicering, mottagning och hantering av SGML-dokument via Word Wide Web på samma sätt som HTML möjliggör. XML har utvecklats för att det skall vara lätt att implementera och vara kompatibel med både SGML och HTML. (Ibid)

Nedanstående är ett exempel på hur man kan märka upp sina dokument i XML.

```
<?XML VERSION="1.0" ENCODING="UTF-8" RMD="NONE"?>
<!DOCTYPE FAQ SYSTEM "FAQ.DTD">
<FAQ>
  <INFO>
    <SUBJECT> XML Uppsats </SUBJECT>
    <AUTHOR> Shirley Saavedra Carlström </AUTHOR>
    <EMAIL> a96shir@student.informatik.gu.se </EMAIL>
    <VERSION> 1.0 </VERSION>
    <DATE> 25.maj.1998 </DATE>
  </INFO>

  <PART NO="1">
    <Q NO="1">
      <QTEXT>Introduktion av XML?</QTEXT>
      <A>SGML light.</A>
    </Q>

    <Q NO="2">
      <QTEXT>XMLs mål</QTEXT>
      <A>Anything</A>
    </Q>

  </PART>
</FAQ>
```

### 3.2.1 eXtensible Linking Language

XLL, handlar om hur man kan konstruera länkar mellan olika objekt, både envägs-, tvåvägs- och flervägslänkar.

- Envägslänkar, dvs. länkar med en speciell roll eller speciellt uppförande.
- Tvåvägslänkar, dvs. länkar som kan följas med samma lätthet i båda riktningarna.
- Flervägslänkar, dvs. länkar som inte är så uppenbara och som går till två eller flera mål.

XLL baserar sig på en till SGML relaterad standard, HyTime<sup>10</sup>, Hypermedia/Time-based structuring language. (Westerling, 1997)

XLL är baserade på hyperlänkar och utarbetade efter många års erfarenhet av hyperlänkar. XLLs möjligheter är kraftfulla och har dessutom inbyggda möjligheter, vilket är tillräckligt kraftfullt för att klara de flesta web-baserade applikationer. (Ibid)

---

<sup>10</sup> HyTime-standarden definierar ett sätt att länka vad som helst oberoende av plats och tid. (Möller s 107)



Speciellt är det tre standarder som har påverkat designen av XLL. Dessa är följande:

- HTML, som tillhandahåller en länkningsmekanism som redan är känd för alla som har kodat en webb-sida.
- HyTime, som är en Internationell standard som bl. a. innehåller vissa användbara koncept för länkningsmekanismer.
- Text Encoding Initiative (TEI) Guidelines, vilket är en SGML-applikation, som innehåller en koncis syntax för hur man kan specificera komplexa länkar. (Statskontoret, 1998, s 25)

Enligt en publikation från Statskontoret, har tillvägagångssättet vid utvecklingen av XLL varit att börja med länkningsmekanismen i HTML och därefter tillfoga mer funktionalitet. Det är naturligt eftersom XML är tänkt att användas på WWW. Därför måste det vara möjligt att länka mellan HTML- och XML-dokument. (Ibid s 25)

XML- och HTML-länkar är kompatibla genom att båda har HREF, vilket är namnet på ett speciellt attribut och är den viktigaste delen i länken. Det innehåller länkens mål uttryckt som en URL, Uniform Resource Locator. Detta för att kunna ha samma länkningsmekanism som i HTML. (Ibid s 26)

XLL tillåter också att man kan deklarerar ett eller flera XML-dokument som en grupp länkar. Detta gör att man på ett enkelt sätt hanterar en uppsättning dokument som innehåller ett nätverk av länkar. (Light, 1997, s 30 f )

### 3.2.2 eXtensible Style

Richard Light skriver i sin bok att Jon Bosak i maj 1997 presenterade det första utkastet till en specifikation för att hantera layouten för XML-dokument. Denna specifikation kallas för XML-style mer känd som XS. (Light, 1997, s 311)

XS skiljer layouten från innehållet i ett XML-dokument. Informationen om layouten ligger i stället lagrad i en egen fil, i en så kallad layoutmall ”style sheet”. (Light, 1997, s 32 f)

XS är grundad på DSSSL<sup>11</sup>, Document Style and Semantics Specification Language, som stödjer omvandlingen och utseendet av SGML dokument. XS är designat för att hantera utseendet av ett XML-dokument på en bildskärm, en skrivare eller någon annan tvådimensionell enhet. XS är en väsentlig ”data-driven (engelska)” layoutmekanism. När man startar en XS ges man möjligheten att skapa nya layoutmallar eller att öppna redan existerande sådana. En eller flera XS layoutmallar anropas när ett XML dokument skall visas. Dessa kan redan vara specificerade i dokumentet, men det finns också möjlighet för användaren att välja den layoutmall han/hon vill. (Ibid s 32 f)

XSs syfte är att låta web-masters enkelt skapa XSL layoutmallar som formar presentationen av XML-dokument. XSL, är ett förslag för att kontrollera layouten i det

---

<sup>11</sup> DSSSL, Document Style and Semantics Specification Language, standard ISO/IEC 10179:1996 (Light s33)

media där XML-dokument publiceras. Ett enkelt sätt lära sig XS och också XSL är helt enkelt att studera de testmallar som finns för dessa verktyg.(Bremser, 1998)

Enligt publikationen från Statskontoret utvecklas XSL för att det skall vara kompatibelt med en annan SGML-relaterad standard, nämligen DSSSL. XSL specificerar syntax för hur man får bygga upp en layoutmall, vilket innebär att även de blir plattform- och verktygsberoende. XSL är en tillämpning av XML eftersom layoutreglerna i XSL skrivs enligt XML-syntax. En layoutmall blir ett eget XML-dokument. En stor fördel med att särskilja informationen och presentationen av densamma är att samma information kan presenteras på ett flertal olika sätt utan att man behöver ändra i XML-dokumentet. Exempel:

- En layoutmall för bildskärmspresentation.
- En för pappersutskrift.
- Mer än en layoutmall, där användaren kan välja mellan olika layouter och välja den som presenterar informationen på bästa sättet. (Statskontoret, 1998, s 24 f)

Enligt Richard Light är CSS1, Cascading Style Sheets Level 1 standarden som finns idag för att hantera utseendet på ett XML-dokument på en bildskärm. CSS1 bestämmer reglerna i termer av elementnamn. CSS1 var specifikt designat kring HTML och vissa av dess karaktärsdrag bygger på HTMLs sätt att tillämpas. (Light, 1997, s 32 f)

En schematisk bild att presentera dokument idag.

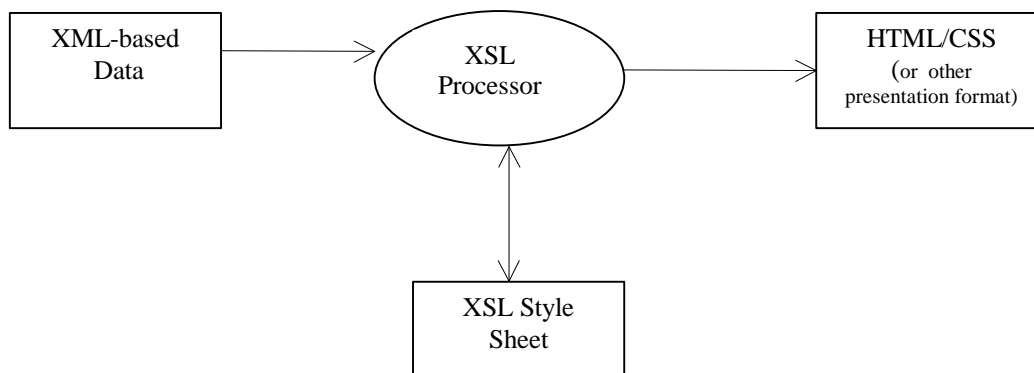


Fig. 6 Presentation av XML-dokument  
Källa: Microsoft.

### 3.2.3 XMLs mål

Målen som XML-gruppen har satt upp för utvecklingen av XML är följande:

1. XML skall gå att använda direkt över Internet.
2. XML skall stödja en mängd applikationer.
3. XML skall vara kompatibelt med SGML.
4. Det skall vara enkelt att skriva program som kan hantera XML-dokument.
5. Antalet valfria egenskaper i XML skall hållas på ett absolut minimum, helst noll.
6. XML-dokument skall vara läsbara och tydliga för användaren.
7. XML skall utvecklas snabbt.
8. XML skall vara formellt och precist definierat.
9. Det skall vara enkelt att producera XML-dokument.
10. XMLs kompakthet är av ringa betydelse. (W3C, 1997)

### 3.2.4 XMLs egenskaper

XML har följande egenskaper:

- kraftfullt
- intelligent
- anpassningsbart
- kraftfulla länkmöjligheter
- enkelt
- plattformsoberoende

XML är intelligent oberoende av informationens komplexitet. Märkord kan inkapslas inom märkord för generell uppmärkning av dokument. Ett exempel på hur man kan inkapsla `<HUND>Lassie</HUND>` är följande, `<HUND> <HANKÖN> <ANLÄNDE> <SCOTTISHURSPRUNG> <9GENERATION> Lassie </9GENERATION> </SCOTTISHURSPRUNG> </ANLÄNDE> </HANKÖN> </HUND>`. Det innebär att man gör en mer specifik uppmärkning. Möjligheten finns att märka upp informationen i oändliga former. Informationen är redan känd i sig själv. (Rees, 1997)

XML kan anpassas till vilka applikationer som helst eftersom det är ett metaspråk. Skraddarsydd uppmärkning av innehållet i ett dokument kan skapas enligt önskemål. Ingen mer begränsning av en statisk uppmärkning. (Ibid)

XML är lätt att underhålla. XML består bara av struktur (thought) och uppmärkning. Layouten presenteras med en separat layoutmall likaså länkar i dokumenten. Kontroller och ändringar kan lätt göras. Det behövs inte längre att kämpa sig igenom uppmärkningen av dokumenten för att ändra något. (Ibid)

XML är enkelt. SGML har en specifikation på 300 sidor. XML 33 sidor. (Statskontoret, 1998)

XML är stabilt, skälet för dess existens är att det är kraftfullt och portabelt . SGML är kraftfullt. HTML är portabelt. XML har båda delarna. XML kan exekveras med eller utan DTD, vilket gör att man kan ladda ned dokumentet mycket snabbare. Allt som

webbläsaren behöver för att läsa ett XML-dokument är det textbaserade dokumentet och en layoutmall som kontrollerar layouten. Om strikt validation behövs kan man skicka med DTD som har med sig exakta detaljer om dokumentens uppmärkning. (Light, 1997)

Richard Light skriver i sin bok att det finns många fördelar med att använda XML eftersom dokumenten har kunskap om sin egen struktur och semantik. Det gör att dokumentet är mer användbart. XML kommer att ha ett stort inflytande i webbens värld. Det är ett skäl till att redan nu börja bygga upp kunskap kring tekniken och hur den kan användas. (Light, 1997, s38)

Exempel på applikationer som framförallt kommer att driva XMLs fortsatta utveckling är sannolikt de som inte kan åstadkommas på grund av HTMLs begränsningar:

- Applikationer som kräver att en webb-klient skall kunna interagera mellan flera databaser med sinsemellan olika struktur. XML möjliggör strukturerad överföring av information.
- Applikationer där man försöker flytta över en stor del av bearbetningen till webbläsaren för att minska belastningen på webb-servern. T. ex. resebokningssystem.
- Applikationer där man vill att samma informationsmängd skall kunna presenteras på olika sätt för användaren. Ett på det är att tabeller och listor skall kunna sorteras på det sätt som läsaren önskar. För att lyckas skapa denna typen av applikationer måste man skilja på innehåll och layout, det gör man i XML.
- Applikationer i form av "intelligenta webb-agenter" som skräddarsyr information till användaren. Idag bygger i princip all sökning på WWW på fritextsökning, vilket innebär att man kan få väldigt många träffar om väldigt många olika saker som man inte är intresserad av. Genom att strukturera information och förse den med information om information kan man göra betydligt effektivare sökningar än genom enkel fritext. (Statskontoret, 1998 s 30)

### **3.3 XMLs användningsområden**

Enligt Richard Light är XMLs roll mest intressant för människor och organisationer som har information som en viktig resurs och där HTMLs form och resurser inte passar när man vill presentera informationen på webben. Följande områden är några exempel.

- Böcker
- Finansiella transaktioner (EDI)
- Tekniska manualer
- Kemiska formler
- Medicinska journaler
- Uppslagsverk

XMLs roll maximeras i situationer där informationen:

- har en lång livslängd
- har inbördes komplexa relationer

- skall återanvändas

När dessa förhållanden finns är företagen villiga att lägga extra arbete för att sätta upp en XML applikation eftersom informationen då blir generellt uppmärkt. (Light, 1997, s 22 - 23)

### **3.3.1 XML och EDI**

Statskontoret, Electronic Data Interchange (EDI) innebär att man definierar en samling standardiserade meddelanden så att företag och organisationer kan utbyta information via elektroniska meddelande. EDI förknippas med elektronisk handel som innebär att två eller flera organisationer kommer överens om att utbyta affärsrelaterad information på ett standardiserat sätt. Affärstransaktioner som t. ex. beställning, orderbekräftelse, betalning osv, kan överföras direkt mellan olika organisationers ekonomisystem utan någon manuell hantering. (Statskontoret, 1998, s 32)

Ett problem med EDI har varit att definiera standarder för hur informationen skall struktureras. En sådan standard är EDIFACT som specificerar ett antal bestämda fält som måste förekomma i en viss följd. Idag måste ofta de företag som vill etablera en EDI förbindelse kontakta varandra i förväg samt göra mer eller mindre komplexa installationer av speciell programvara för att konvertera de elektroniska dokument och översätta dem till ett format som kan tolkas av de interna systemen. Detta är ett kostsamt och statitisk arbetssätt. XML är väl lämpat för att användas som format för EDI. Formatet kommer sannolikt att stödjas av de flesta webb-läsare i framtiden. Man måste kunna utnyttja standardprodukter för att kunna dra nytta av andras utveckling och dela på kostnader. (Ibid s 32)

## 4 Skillnader mellan SGML/XML

<i>Skillnader</i>	<i>SGML</i>	<i>XML</i>
Lämplig för webben	Nej	Ja
Komplext	Ja	Nej
Kräver DTD	Ja	Nej
Funktionalitet	Hög	Låg
Teckenuppsättning	ASCII	Unicode
Lätt att lära sig	Nej	Ja

De flesta skillnaderna mellan SGML och XML är mer på detaljnivå, dvs. hur man får skriva kommentarer, deklarerar element, attribut osv. De flesta av dessa skillnader är till XMLs nackdel, man tillåts göra mer i SGML än i XML. (Light, 1997, s 357 ff)

Det finns dock även detaljskillnader som är till XMLs fördel. XML är mer förlåtande vid deklARATION av attribut. Deklarationer som skulle ge felmeddelanden i SGML tillåts i XML. (Ibid)

## 5 SGML/XML i företaget

Grunden för intervjuens utförande är att se hur SGML/XML används på företagen och vilka konsekvenser införandet av dessa har medfört för företagen. Syftet med urvalet av företag har varit att vi ville titta på olika verksamhetsområden. De flesta personer vi har intervjuat är experter på SGML/XML och har varit med vid införandet av dessa och som följer den fortsatta utvecklingen på området.

Nedan följer en presentation av de företag som ingår i vår undersökning:

Ericsson Utvecklings AB, ansvarar för utvecklingen och produktionen av AXE-systemet, som är världens mest installerade telefonisystem.

Ericsson införde SGML 1991 och följer utvecklingen av XML. Ericsson kommer att gå över till XML så fort det har kommit en editor som de vill ha och en av anledningarna till det är att XML ger ökad tillgänglighet på Internet.

Statskontoret, en statlig myndighet. Statskontoret gör generella upphandlingar för den offentliga sektorn, bland annat upphandlingar av IT, tekniska plattformar, mm.

Statskontoret har gjort en utredning om XMLs möjligheter och dess tillämpningar. Statskontoret anser att XML kommer att få avgörande betydelse för Internets fortsatta utveckling. Statskontoret kommer att rekommendera den offentliga sektorn att använda XML.

Göteborgs-Posten, Västsveriges morgontidning samt också Skandinavians näst största. GP använder sig i dag av SGML i sin redaktionella arbete.

Inzide kreativ teknik AB, ett konsultföretag inom teknisk information. De bistår olika företag med expert kunskap inom SGML. De har t. ex. varit med att införa SGML på Volvo.

WM-data, ett av nordens ledande datatjänstföretag. WM-data har börjat att ge kurser om XML och erbjuder även helhetslösningar med SGML/XML. Roxana Öberg på WM-data säger att XML kommer att *"...sprida sig som ringar i vattnet..."*. Hon menar att om ett företag börjar använda XML kommer andra företag att följa efter.

SKF, en internationell industrigrupp inom tillverkning av högteknologiska industrikomponenter. SKF är intresserade av att eventuellt införa SGML/XML.

### 5.1 Varför har företagen valt SGML/XML?

Orsaken till att företagen intresserat sig för SGML/XML är att de vill lagra sina dokument plattformsoberoende, ha dem lätt sökbara, kunna återanvända dem samt kunna publicera dem på olika media. Det kan även finnas krav från andra organisationer att företagen skall använda sig av t. ex. SGML.

Enligt en expert på SGML/XML kommer t. ex. den offentliga sektorn att satsa mycket på att införa XML. Det skapas redan nu ett system för spridning/hämtning av information. Systemets syfte är att man skall kunna utbyta information. All metainformation om dokument skall vara i XML-format.

Riksdagen har t. ex. beslutat att författningssamlingar, lagar och förordningar från regering och riksdag skall vara i SGML-format men troligen kommer formatet att bli XML.

De effektivitetsmässiga konsekvenser som finns för de företag vi har intervjuat är positiva. Helt klart är det att företagen på sikt kommer att göra vinster och även redan har gjort vinster med att införa SGML. Ericsson t. ex. producerar 5000 cd per år och för varje cd de producerar så sparar de 2 veckor i arbetstid.

Företagen vill uppnå en högre funktionalitet och en effektivare användning av information. En effektivare användning av information är återanvändning. En datakonsult nämnde som ett exempel ett företag som har en dokumentation över en motor och om företaget tillverkar en ny motor, som till 70 % består av nya komponenter och till 30 % av komponenter från den gamla motorn behöver bara 70 % av dokumentationen förnyas. Resterande 30 % kan återanvändas vilket spar tid och pengar.

Den offentliga sektor vill t. ex. använda sig av digitala signaturer och det är viktigt för dem att ha ett format som inte förstör dessa. XML är ett format som väl lämpar för deras behov. Genom en ökad användning av elektroniska dokument kommer användningen av pappersdokument att minska och det har en positiv effekt på miljön.

Det som har varit gemensamt för företagen varför de valde att satsa på SGML/XML är att verktygen är leverantörsberoende och neutrala på alla plattformar.

## **5.2 Införande, kostnader och verktyg**

För de företag som vi har intervjuat har implementationstiden av SGML varit mellan ett och två år från det att beslut fattats tills det att första dokumentet har kunnat presenterats. En intervjuat datakonsult menar att företagen få vara beredda på en lång startprocess.

Det som är gemensamt för företagen är att dokumentanalysen har varit den mest tidskrävande delen i implementationen. Dokumentanalysen är en viktig del för att kunna skapa bra dokumentstrukturer.

Kostnader för att införa SGML/XML är höga enligt de företagen som vi har intervjuat. Om ett företag väljer att införa SGML kan kostnaderna bli mycket högre än att välja att införa XML.

Editorer för att hantera SGML är dyrare än editorer för att hantera XML. Dessutom är XML editorerna lättare att använda. Kostnader för XML kan alltså bli lägre. Införandet av XML kommer ändå att innebära stora investeringar inledningsvis men att på sikt kommer det att bli effektivitetsvinster för företagen. Exempel på dessa är att företagen



kan göra intelligenta sökningar, sorteringar, sammanställningar och ändringar samt får en enklare hantering av dokument med lång livslängd. Detta medför även en minskad materialförbrukning.

Verktygen som företagen behöver för att införa SGML/XML kan vara följande. Databas för att samla företagets dokumentation, underhåll av databasen kanske ny hård- samt mjukvara. Enbart att samla all dokumentation i en databas är ett stort kostsamt projekt.

Ett av de intervjuade företag har varit delaktiga i utvecklingen av det SGML-system de använder. Systemet är anpassat till deras verksamhet, hittills har de satsat c:a 2 miljoner kronor. Systemet har även sålts till andra. Detta gör att företaget räknar med intäkter på sikt. Spinn-off effekter som de redan har märkt är att företaget lätt kan överföra sina dokument elektroniskt mellan olika delar inom företaget. Detta har medfört att personalstyrkan har minskats.

En annan viktig del som framgick av intervjuerna, är svårigheten i början för användarna att tänka på ett nytt sätt d.v.s. att man inte bara skriver ett dokument utan att man skall se informationen på ett strukturellt sätt. En expert betonade följande *"...utbildningen av personalen är viktig och med facit i hand så borde vi ha satsat mer på utbildning..."*. Förutom det nya sättet att tänka på information har det, enligt experten, inte varit några större problem för användarna att lära sig SGML.

Det är svårt för användarna är att inte tänka på dokument layoutmässigt, t. ex. när man "taggar" ett dokument i SGML är det svårt för användarna att föreställa sig det färdiga resultatet. En datakonsult säger *"...SGML är inte WYSIWYG, what you see is what you get. /.../ Det kräver lång tid innan man får den kunskapen att man erfarenhetsmässigt kan föreställa sig hur det färdiga dokumentet kommer att se ut..."*

Layouten i SGML/XML sköts separat med layoutmallar. I SGML t. ex. använder man sig av en FOSI för att kunna presentera dokumenten. För närvarande krävs en lång formatteringsprocess för att se hur ett SGML-dokument kommer att se ut. Dokument bör därför vara korta och bilder bör vara generella.

Det som är gemensamt för företagen är att de har haft svårt att hitta användarvänliga SGML-editorer. Ett av företagen har därför utvecklat en egen lösning med ett användargränssnitt som passar deras verksamhet.

För XML utvecklas mer användarvänliga editorer och till hösten 1998 kommer man i office paketet att kunna spara i XML-format. Detta kommer att underlätta användningen av XML.

## 6 Slutsatser

Vi har studerat beskrivningsspråken SGML och XML samt vad det innebär för ett företag/en organisation att införa ett beskrivningsspråk. Vi har tittat på både positiva och negativa konsekvenser. De konsekvenser vi undersökt är följande:

- effektivitetsmässiga
- kostnadsmissiga

Vi börjar med att konstatera att SGML är ett komplext och omfattande beskrivningsspråk men har en stor potential att underlätta dokumentationshanteringen inom företag/organisationer. Det är främst lämpat för företag/organisationer som har dokument med lång livslängd.

Genom intervjuerna har det framkommit att XML är så nytt att företag/organisationer inte har hunnit implementera det. Men bland företagen verkar XML vara det mest intressanta alternativet vid valet av beskrivningsspråk, framförallt för att XML är lätt att använda och att det är ett beskrivningsspråk som de i framtiden kommer att satsa på.

Vi har funnit att kostnader att införa SGML är höga men troligtvis kommer dessa att löna sig på sikt. Det har visat sig att det är en lång process att införa SGML. Organisationens hela dokumentation måste analyseras. Regler måste författas för de olika dokumentens strukturer. Layoutmallar måste skapas. Editeringsverktyg måste införskaffas och personal måste utbildas.

Vi anser dock att XML blir billigare att implementera på grund av att SGML-editorer är dyrare än XML-editorer. Det utvecklas fler editorer för XMLs användning bland annat har Microsoft annonserat att office 98 kommer att ha XML-format

Det ställs höga krav på att man väljer rätt typ av databas som passar ens verksamhet och de mål man har vid införandet av ett beskrivningsspråk.

På sikt så finns det en lönsamhetspotential i att implementera ett beskrivningsspråk. Genom att införa SGML/XML kan man återanvända dokument, göra intelligenta sökningar, sortera och sammanställa dokument på ett enkelt sätt. Detta bör medföra en kostnadsminskning för företaget/organisationen.

Användning av SGML/XML gör att man kan presentera dokument på olika media utan att några förändringar av dokumenten behöver göras. SGML/XML effektiviserar personal utnyttjandet.

SGML är svårt att använda på grund av att de editorer som finns idag för SGML inte ger ett bra användarstöd. XML kommer att bli lättare att använda eftersom de editorer som utvecklas för XML kommer att ge ett bättre användarstöd.

Gemensamt för de båda beskrivningsspråken är att det krävs "ett nytt tankesätt" när man skall skapa ett dokument. Det nya tankesättet är att tänka strukturellt och inte

layoutmässigt som är det vanliga idag. En kritisk framgångsfaktor är att berörd personal ges tillräcklig utbildning.

Ett företag/en organisation kan räkna med att få en mer effektiv hantering av sin dokumentation om man satsar på SGML/XML genom att man får ett bättre grepp om sin dokumentation.

Vi anser avslutningsvis att implementering av SGML eller XML får övervägande positiva konsekvenser för ett företag/en organisation med ”rätt dokumentationstyp” det vill säga dokument med lång livslängd och/eller som måste presenteras med hjälp av olika presentationsmedia. Det är viktigt att genomföra en ordentlig dokumentationsanalys innan ett slutligt beslut om ett införande fattas.

XMLs fortsatta utveckling kan leda till att ett införande av XML kan ge positiva effekter för alla företag som vill strukturera upp sin dokumentationshantering.

Införandet av SGML/XML ger en möjlighet till en ökad användning av elektroniska dokument. På sikt kan detta minska pappersförbrukningen vilken vi ser som en positiv faktor med tanke på miljön.

## 7 Källförteckning

- Bergström, P och T Ericsson, "HTML, XML och SGML – en interaktiv jämförelse <<http://www.sorman.se/xml/>>, 22 maj 1998.
- Bremser, W. "First Baby Steps on The Journey to Working XML", 30 april 1998. <<http://www.internetworld.com/print/1998/03/30/iwlab/19980330-baby.htm>>, 22 maj 1998.
- Broady, D & J Juliusson. "Introduktion till SGML" 2 augusti 1997. <<http://www.nada.kth.se/~broady/digit/sgml-intro/sgmlintr.htm>>, 7maj 1998.
- Clark, J. "Comparison of SGML and XML, World Wide Web Consortium", 15 december 1997. <<http://www.w3.org/TR/NOTE-sgml-xml.html>>, 24 april 1998.
- Geijer, E. "XML blir officiell standard", Webbteknik nr 12, 1998. <<http://domino.idg.se/cs/artikel.nsf/a73b44e923fd585fc125650f0031bd38/3e5d330a65e70324c12565b00034c16a?OpenDocument>>, 1 maj 1998.
- Gennusa, P. "XML – why, when and how?", Database Publishing Systems, 1998. <<http://info.admin.kth.se/SGML/Konferenser/sgml98/DATA/pam.htm>>, 24 april 1998.
- Goldfarb, C F. "The Roots of SGML – A Personal Recollection" 1996. <<http://www.sgmlsource.com/history/roots.htm>>, 16 maj 1998.
- Goossens M & J Saarela. "A Practical Introduction to SGML" 20 juni 1995. <<http://www.irb.hr:80/~cern/WWW/publications/sgmlen/sgmlen.htm>>, 16 maj 1998.
- Goossens M, "Why SGML?" 8 mars 1995. <<http://faust.irb.hr/~cern/WWW/htmltools/node1.html>>, 16 maj 1998.
- Heymowska, A. "Webbens nya språk är XML", Nätverk & Kommunikation, 12 februari 1998. <<http://domino.idg.se/natkom/nokart.nsf/24da562fb5b11fe7c12563a000657853/a9b3744696ad0f2a412565a70061c62a?OpenDocument>>, 4 april 1998.
- International Standard ISO 8879, Information processing – Text and office systems – Standard Generalized Markup Language (SGML), First edition, 15 oktober 1986.
- Light, R "Presenting XML" Sams. Net Publishing, 1997, First edition.
- Microsoft "XML:Enabling Next-Generation Web Applications", 3 april 1998. <<http://www.microsoft.com/xml/articles/xmlwp2.htm>>, 24 april 1998.

Möller, A. "SGML – en introduktion till Standard Generalized Markup Language", 1994, Studentlitteratur, Lund.

Randall, N. "XML: A Second Chance for Web Markup" PC Magazine ONLINE, 4 november 1997.

<<http://www.zdnet.com/pcmag/issues/1619/pcmg0139.htm>> , 24 april 1998.

Rees, L.C. "a brief introduction to the mysteries of the eXtensible markUp language" 1997.

<<http://www.geocities.com/SiliconValley/Peaks5975/wxml.htm>>, 1 maj 1998.

SGML Users' Group. "A Brief History of the Development of SGML". 11 juni 1990.

<<http://www.sgmlsource.com/history/sgmlhist.htm>>, 16 maj 1998.

Statskontoret, "Vad är XML?", Publikation 1998:6

Starker-Laurberg, O, M Taavo & L-O Lindgren. "Konsten att hålla 120 pärmar aktuella".

<<http://info.admin.kth.se/SGML/Konferenser/sgml198/DATA/html/sht.htm>>, 24 april 1998.

Toppledarforum. "Toppledarforums arbetsprogram".

<<http://toppled.nutek.se/sid4n.html>>, 16 maj 1998.

W3C "Extensible Markup Language (XML): Part I. Syntax W3C Working Draft 31-Mar-97" 31 mars 1997.

<<http://www.w3.org/TR/WD-xml-lang-970331.html>>, 4 april 1998.

W3C "Extensible Markup Language (XML) 1.0 W3C Recommendation 10-Februari-1998" 1998.

<<http://www.w3.org/TR/REC-xml>>, 23 april 1998.

Westerling, J. "XML: Framtiden för World Wide Web",  
Datakommunikationsarkitekturer, 1997

<http://www.tcm.hut.fi/Opinnot/Tik-110.350/Tehtavat/essays/xml.html>> 1 maj 1998