

# <XML/> DIGITAAL ARCHIVEREN

Filip Boudrez

Antwerpen 2002

## 0. Inhoudstafel

|         |                                                  |    |
|---------|--------------------------------------------------|----|
| 1.      | Inleiding.....                                   | 1  |
| 2.      | De nood aan XML: XML vs. HTML en SGML.....       | 2  |
| 3.      | De XML-documenten .....                          | 3  |
| 3.1     | Samenstelling .....                              | 3  |
| 3.2     | Documentmodellering .....                        | 4  |
| 3.3     | DTD en XML Schema.....                           | 6  |
| 3.4     | Voornaamste syntaxregels .....                   | 8  |
| 3.5     | Validatie .....                                  | 9  |
| 3.6     | Eigenschappen .....                              | 10 |
| 3.7.    | Opslag en terugvinden van XML-documenten .....   | 13 |
| 3.7.1   | In een bestandssysteem .....                     | 13 |
| 3.7.2   | In een databank .....                            | 13 |
| 3.7.2.1 | Opslag in een relationele databank.....          | 14 |
| 3.7.2.2 | Opslag in een object-georiënteerde databank..... | 14 |
| 3.7.2.3 | Opslag in een native XML databank.....           | 14 |
| 4.      | De XML familie .....                             | 14 |
| 5.      | XML voor archiveringstoepassingen .....          | 16 |
| 5.1     | XML als uitwisselingsformaat .....               | 16 |
| 5.2     | XML als archiveringsformaat .....                | 17 |
| 5.3     | XML als metadataformaat .....                    | 20 |
| 5.4     | XML als inkapselingsformaat .....                | 21 |
| 6.      | Besluit.....                                     | 22 |
| 7.      | Bronnen .....                                    | 22 |

## 1. Inleiding

XML (eXtensible Markup Language) is een markuptaal die van bij zijn vastlegging een grote vlucht heeft genomen. XML heeft zijn toenemende verspreiding in grote mate te danken aan het feit dat het op uiteenlopende wijzen binnen verschillende IT-toepassingen bruikbaar is<sup>1</sup>. XML is immers meer dan een puur bestandsformaat. Ondertussen is er rond deze markuptaal op vrij korte tijd al een hele XML-technologie ontstaan.

Eén van de mogelijke toepassingsgebieden voor XML is digitale archivering. XML biedt heel wat perspectieven voor de archivering van digitale documenten en de verwachtingen zijn binnen de archiefwereld dan ook hoog gespannen. Ondanks zijn relatief jonge leeftijd wordt XML al in veel richtlijnen en procedures als standaard aanbevolen.

In deze bijdrage wordt nagegaan hoe XML inzetbaar is voor archiveringsdoeleinden. Na een algemene voorstelling van XML en de gerelateerde technologieën volgt een bespreking van de verschillende manieren waarop archiefdiensten XML kunnen gebruiken bij de archivering van digitale archiefdocumenten. Hierbij wordt ook aandacht besteed aan de praktische implementatie en wordt verwezen naar praktijkvoorbeelden<sup>2</sup>.

---

<sup>1</sup> Voorbeelden van IT-toepassingen zijn onder andere: business-to-business, content providers, web content management, wireless applications, enz. XML wordt hierin gebruikt voor data uitwisseling tussen verschillende systemen, dynamische on the fly transformaties, informatie integratie en voor data verdeling in verschillende formaten. Een overzicht van XML-toepassingen wordt bijgehouden op <http://xml.coverpages.org/xmlApplications.html>

<sup>2</sup> De betekenis van technische termen die niet in deze bijdrage zelf worden verklaard, vindt u terug in de woordenlijst op de DAVID-website (<http://www.antwerpen.be/david> → publicaties). Op de DAVID-website zijn eveneens on line voorbeelden beschikbaar (→ cases).

## 2. De nood aan XML: XML vs. HTML en SGML

XML is een markuptaal. Een markuptaal is een technologie voor het structureren van een document door middel van tags. SGML en HTML, twee markuptalen waarmee XML is verwant, waren niet geschikt voor een vlotte en gebruiksvriendelijke uitwisseling van gestructureerde documenten via het web, zodat er een nieuwe markuptaal werd ontwikkeld<sup>3</sup>.

XML is ontstaan uit de samenwerking tussen de industriële en academische wereld. De computerindustrie (Sun Microsystems, Hewlett-Packard, Microsoft, Netscape, Adobe, Fuji Xerox) en SGML-producenten (ArborText, Inso, SoftQuad, Grif, Isogen, Texcel) waren van bij de start bij de ontwikkeling betrokken. Later sprongen ook IBM, Oracle en Omnimark op de wagen. De academische wereld werd vertegenwoordigd door onder meer het Text Encoding Initiative (TEI), NCSA en James Clark. De ontwikkeling van XML startte in 1996. Twee jaar later werd XML als Recommendation door het W3C vastgelegd.

De XML-specificatie definieert een standaard voor het toevoegen van markup aan documenten. De XML-specificatie bevat geen vaste set tags met een bepaalde betekenis, maar regels voor het samenstellen van een XML-document. XML is dus een meta-taal en biedt de gebruiker de mogelijkheid om zelf de tags, hun attributen en onderlinge relatie te definiëren. XML verschilt op dit punt fundamenteel met HTML. In de HTML-specificatie liggen zowel de beschikbare tags, hun attributen als hun betekenis vast. In HTML kan de gebruiker geen eigen tags creëren of geen eigen betekenis aan de tags geven zodat men met HTML-tags een document geen gebruikers-gedefinieerde structuur kan geven. Met XML stelt de gebruiker als het ware zijn eigen lexicon samen. Een ander belangrijk verschilpunt met HTML is de functie van de tags. De HTML-tags zijn in de eerste plaats op visuele presentatie gericht. De HTML-tags zeggen niets over de inhoud of structuur van de computerdata. Met andere woorden, een HTML-document bevat geen echte structuur die op de data zelf is gebaseerd. In tegenstelling tot XML is HTML niet data of object georiënteerd. Dit heeft onder meer voor gevolg dat binnen HTML-bronbestanden geen gestructureerde en semantische zoekopdrachten (bijv. maak een lijst van alle publicaties van auteur x gesorteerd op publicatiedatum) mogelijk zijn, wat een groot nadeel is voor accurate zoekopdrachten op het web. HTML is hierdoor niet geschikt voor informatie uitwisseling of voor geautomatiseerd gegevensbeheer (opzoeken, manipulatie, integratie). Naast de mogelijke versiegebondenheid van tags en attributen heeft dit ook voor gevolg dat bij HTML inhoud en opmaak niet van elkaar worden gescheiden.

XML is evenmin hetzelfde als SGML<sup>4</sup>. Beide markuptalen hebben met elkaar gemeen dat ze worden gebruikt voor dataopslag en -uitwisseling en dat de gebruiker de documenten zelf gedefinieerde tags en structuur kan geven, maar SGML-documenten zijn evenwel te complex om vlot via het web uit te wisselen. SGML vraagt zware systemen en kan niet gemakkelijk in een webbrowser worden geïmplementeerd. SGML wordt bijgevolg vooral in grote documentatieprojecten gebruikt. XML is een afgeslankte versie van SGML en kan misschien het best als een application profile van SGML worden beschouwd<sup>5</sup>. XML is eenvoudiger dan SGML. XML nam van SGML de mogelijkheid tot identificatie van informatie, uitbreiding en validatie over. In tegenstelling tot SGML, is bij XML de aanwezigheid van een DTD niet vereist. Een SGML-applicatie is in staat om een XML-document in te lezen, maar niet omgekeerd.

Het ontstaan van XML moet dus tegen de achtergrond van informatie-uitwisseling op het Internet worden gesitueerd. XML was initieel bedoeld om HTML te vervangen en kreeg de reputatie het nieuwe 'ASCII van het web' te zijn. XML vervult nu al een belangrijke rol op het web, maar ondertussen wordt XML ook voor andere doeleinden en binnen andere toepassingen gebruikt. XML gaat om digitale informatie in het algemeen.

<sup>3</sup> Voor een meer gedetailleerde bespreking van HTML en SGML, zie: *Standaarden voor digitale archiefdocumenten* (<http://www.antwerpen.be/david> onder 'cases' of 'publicaties').

<sup>4</sup> Voor een gedetailleerde vergelijking, zie: J. CLARK, *Comparison of SGML and XML*, 1997 (<http://www.w3.org/TR/NOTE-sgml-xml.html>).

<sup>5</sup> Dit kan mooi geïllustreerd worden aan de hand van de lengte van de specificatie. De SGML-specificatie is meer dan 600 pagina's lang, terwijl die van XML nauwelijks 26 pagina's in beslag neemt.

### 3. De XML-documenten

#### 3.1 Samenstelling

XML is een op tekstkarakters gebaseerde markuptaal waarmee de structuur en betekenis van documenten kan worden vastgelegd. Een XML-document bestaat uit maximaal drie delen: de header, de body en de epiloog.

Een XML-bestand begint met een header of proloog die onder meer de declaratie bevat. De declaratie bevat minstens het versienummer van de XML-specificatie die bij de creatie van het document werd gevolgd. Optioneel kan men in de header vastleggen welke karakterset wordt gebruikt en of het bestand afhankelijk is van externe bronnen<sup>6</sup>. De declaratie kan ook worden uitgebreid met een verwijzing naar een DTD, een XML Schema, een stylesheet of de toegepaste namespaces (zie 4. De XML-familie).

De XML-body begint met het rootelement en wordt afgesloten door de eindtag van het rootelement. De body bestaat uit de elementen waaruit het XML-document is opgebouwd. De elementen (of objecten) kunnen enigszins met de gegevensvelden van een databank worden vergeleken. De lengte van een element kan variëren van één karakter tot een heel boek. De elementen in een XML-bestand worden hiërarchisch gemodelleerd zoals een tree: een element kan een onderdeel zijn van een ander element (de parent) of uit andere elementen bestaan (de childs)<sup>7</sup>. Elementen die dezelfde parent hebben, worden siblings of sister elementen genoemd. Het ordenen van de elementen wordt *nesting* genoemd. Het element op het hoogste niveau is het rootelement. Het XML-rootelement is geen onderdeel van een ander element, maar bevat de elementen waaruit het XML-bestand is opgebouwd. Aan elk element kunnen attributen worden toegekend die meer informatie verschaffen (bijv. <persoonsgegevens openbaar="nee">). De elementen worden altijd afgebakend door middel van een begin- en een eindtag.

De epiloog is optioneel en kan processing instructions, whitespace en commentaren bevatten.

Afbeelding1: De opbouw van een XML-document



<sup>6</sup> encoding: UTF-8, UTF-16 en ISO-8859-1, ISO-10646-UCS-2 of ISO-10646-UCS-4. UTF-8 is de voorgedefinieerde encoding, maw applicaties gaan er van uit dat UTF-8 werd toegepast wanneer geen encoding werd gespecificeerd. standalone="yes": document bevat alle entiteit declaraties; standalone="no": document heeft een externe DTD nodig.

<sup>7</sup> De inhoud van elementen kan mixed, simple, element en empty zijn. (mixed content: bevat tekst en andere elementen; simple: bevat enkel tekst; element content: bevat enkel andere elementen; empty: geen content)

De markup- en inhoudskarakters van een XML-documenten bestaan allemaal uit tekstkarakters. XML ondersteunt Unicode met uitzondering van een aantal controlekarakters. De tekstkarakters worden lineair opgeslagen. Dit betekent niet dat een XML-document niet de gegevens van een binair bestand kan bevatten. Het XML-document kan niet de binaire bitstream zelf bevatten want deze bevat ongeldige karakters. Er zijn twee methoden om binaire data in XML te bewaren: de binaire data worden als unparsed entiteiten (CDATA) bewaard of er wordt Base64-encoding toegepast. Deze laatste oplossing wordt het meest toegepast. Bij Base64-encoding (rfc-1341, rfc-2045<sup>8</sup>) worden de binaire data naar tekstkarakters omgezet. De Base64 standaard gebruikt een 64 ASCII-karaktersubset (A-Z a-z 0-9 +/) om binaire data weer te geven en het karakter "=" voor opvulling. De omzetting van binaire data naar Base64 gebeurt door de applicatie. Op die manier kan de bitstream van een TIFF-bestand naar tekstkarakters worden omgezet en opgeslagen worden in een XML-bestand. Om het TIFF-bestand te reconstrueren worden de tekstkarakters opnieuw naar binaire data omgezet<sup>9</sup>. De data van een binair bestand in een XML-document opnemen, kan interessant zijn voor gegevensuitwisseling en/of -inkapseling.

Alle data in een XML-document is pure tekst. Strikt genomen bevat een XML-document ook geen andere datatypen zoals datums en integers. Het gebruik van tekstkarakters voor zowel markup als inhoud betekent dat een XML-document altijd in een gewone teksteditor kan worden geopend en dat de karakters altijd leesbaar zijn. XML-documenten worden bijgevolg menselijk leesbaar genoemd.

## 3.2 Documentmodellering

Met behulp van apparatuur en programmatuur wordt kennis als weergave van de realiteit op een medium opgeslagen in de vorm van bitstreams. Deze bitstreams zijn op zich niets meer dan computerdata die uit zichzelf geen betekenis of context bevatten. Digitale archivering heeft echter niet tot doel bitstreams te bewaren, maar kennis op een digitale wijze in de tijd over te brengen zodat de documenten interpreteerbaar blijven. Rekening houdende met de technologische veroudering dient de digitale representatie van kennis op een infrastructuur onafhankelijke wijze te gebeuren. Men kan dit bereiken door data- of documentmodellering<sup>10</sup>. Dit is de geijkte manier om kennis op basis van computerdata op een menselijk interpreteerbare wijze te bewaren. XML biedt de mogelijkheid om documenten te modelleren.

Ontstaan in de databankomgeving, is modellering een methode om computerdata aan een vastgelegd kennismodel te koppelen. De bouwstenen van het kennismodel zijn gedefinieerde en gerelateerde computerdata. Door identificerende en karakteriserende informatie aan data toe te voegen, wordt al de betekenis van data duidelijk, maar dit is nog onvoldoende om tot kennis te komen. De informatie moet hiervoor nog gestructureerd worden. Enkel door relaties tussen de informatie vast te leggen, komt men tot kennis. Kennis is met andere woorden gestructureerde informatie. Modellering is niet alleen noodzakelijk om in de toekomst kennis uit data te reconstrueren, maar zorgt er ook voor dat bij uitwisseling onduidelijkheden en misverstanden worden vermeden<sup>11</sup>.

Het datamodeleringsproces van documenten kan men in verschillende stappen opsplitsen: 1. het benoemen van de data, 2. de identificatie en definiëring van de data door ze in een classificatieschema te gieten, 3. het vastleggen van de relaties tussen de data, 4. het bepalen van de eigenschappen van de data. Het resultaat van dit proces is een kennismodel, waar de computerdata worden ingepast bij de creatie van documenten. Het ontwerpen van documenten is bijgevolg een vertaling van het kennismodel naar een geheel van regels die bij de creatie worden gevolgd.

De kern van het XML-mechanisme voor het modelleren van documenten is het toevoegen van semantische tags aan data (bitstreams) en de nesting van elementen. De dataobjecten komen overeen met de inhoud van

<sup>8</sup> Groepen van 3 binaire octetten worden hierbij omgezet naar vier tekstkarakters, waarbij elk tekstkarakter 6 bits van de oorspronkelijke 24 bits voorstelt. Volgens de MIME-specificatie is de lijnlengte beperkt tot 76 karakters, maar dit geldt niet binnen een XML document.

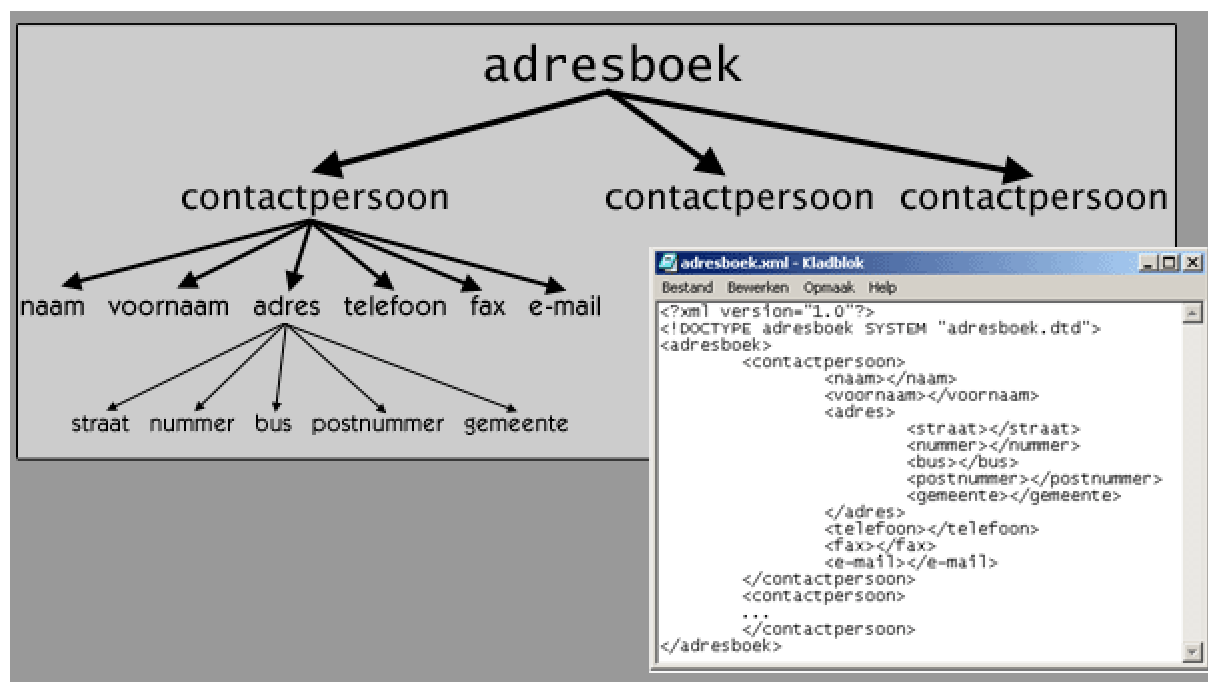
<sup>9</sup> Dit houdt niet in dat het voor archiveringsdoeleinden een goede optie is om om het even welk binair formaat in een XML-bestand te bewaren. Men dient altijd nog over een applicatie te beschikken om het binair bestand te kunnen lezen.

<sup>10</sup> Binnen de IT wordt hiervoor soms ook de term "informatiemodellering" gebruikt.

<sup>11</sup> R.W. MOORE, *The preservation of data, information and knowledge*, 2001.

de elementen waaruit een XML-document is opgebouwd. Door semantische tags aan de elementen toe te kennen, worden de logische componenten van een document geïdentificeerd en krijgen ze betekenis. De data "01012000" bijvoorbeeld kunnen van alles betekenen. Door de tags <startdatum\_davidproject> er aan toe te voegen krijgt deze cijferreeks echter betekenis. De tags kunnen beschouwd worden als attributen en bevatten informatie over de data waardoor interpretatie mogelijk wordt. De nesting van de elementen weerspiegelt de relatie tussen de informatiecomponenten van het document. Aan de tags kunnen attributen worden toegekend. De attributen geven aanvullende informatie over de inhoud van de tag of over de tag zelf. De keuze van tags en attributen is volledig vrij zodat de gebruiker ze volledig kan afstemmen op de documentvereisten.

Afbeelding 2: Een hiërarchische boomstructuur in een XML-document gegoten.



Het toevoegen van tags (tagging) is een heel belangrijke stap in het modelleren van een XML-document. De functie van tagging gaat bij documentmodellering een stap verder dan louter het annoteren van documenten waarbij informatie aan de computerdata wordt toegevoegd en ze betekenis krijgen. Door de elementen te nesten of te structureren worden ze gerelateerd aan een vastgelegd kennismodel. Het resultaat in XML is dan hiërarchisch gestructureerde informatie. Beide stappen vormen de basis van het gebruik van tags binnen een XML-document. Dit neemt niet weg dat de XML-tags nog tal van andere voordelen opleveren:

- onderscheiden als unieke delimiters de verschillende atomen van het bestand.
- identificeren of beschrijven de informatie waarvan zij de envelop zijn. De tags labelen als het ware de gegevensvelden van een document en verschaffen informatie over de betekenis en eventueel het datatype. Voor de betekenis van bepaalde data is men niet meer afhankelijk van externe documentatie.
- voegen samen met hun attributen semantiek toe aan de computerdata
- laten nesting van de documentcomponenten toe en weerspiegelen de structuur
- maken krachtige en gestructureerde zoekopdrachten mogelijk, met meer accurate zoekresultaten als gevolg.
- laten sorteeracties toe
- zorgen ervoor dat computers gemakkelijk XML-documenten kunnen verwerken en interpreteren. In XML-applicaties is vastgelegd hoe bepaalde tags en hun inhoud worden verwerkt.
- functioneren als nodes (bijv. adresboek/contactpersoon/adres/straat) binnen de XML-structuur en geven toegang tot de data voor raadpleging en/of manipulatie op basis van scripting tegen de nodes.
- zijn de primaire instrumenten voor het begrijpen en beheren van documenten. Documenten hoeven hierdoor geen specifieke headers of applicatiespecifieke karakters te bevatten.

Met het oog op digitale archivering levert documentmodellering een aantal interessante voordelen op. Bij XML staat het document zelf en de organisatie van de data centraal, en niet de applicatie die de data verwerkt. Met XML kan de wijze van dataopslag in principe volledig worden bepaald door de aard en de vereisten van de data, het type archiefdocument of het documentmodel zelf. In een gewone desktopomgeving zijn het overwegend de applicaties die de wijze van dataopslag bepalen.

Het toepassen van een documentmodel bij creatie, eventueel in combinatie met een DTD of XML Schema, biedt mogelijkheden op het vlak van kwaliteitscontrole en -handhaving. Hierdoor kan van de gebruiker een vaste structuur bij bepaalde types archiefdocumenten afgedwongen worden. Het biedt de archivaris eveneens de kans om van bij de creatie de archiefvorming te begeleiden. Documentmodellering speelt ook een belangrijke rol bij het archiveren van hetzelfde type archiefdocument afkomstig van verschillende platformen of informatiesystemen. Op die manier kan men per type archiefdocument één documentmodel vastleggen en toepassen bij opname in het digitaal archief zodat er uniformiteit is. Zo vermijdt men de aanwezigheid van verschillende documentmodellen voor hetzelfde type archiefdocument. Een praktische toepassing is bijvoorbeeld het vastleggen van één documentmodel voor e-mails dat bij archivering wordt gevolgd, ongeacht de applicatie waarbinnen de e-mails werden opgemaakt of ontvangen.

Door een documentmodel toe te passen wordt in het XML-document meer dan alleen maar de data bewaard. Naast de inhoud bevat een XML-document ook de structuur op een expliciete wijze. Dit levert een belangrijk verschil op met bijvoorbeeld door komma's gescheiden tekstbestanden. Het vastleggen van relaties door middel van nesting is binnen door komma's gescheiden bestanden niet mogelijk. Het derde component van een traditioneel document, de presentatie, worden doorgaans in een afzonderlijke stylesheet bewaard<sup>12</sup> (zie: 3. De XML-familie). De opmaakgegevens worden niet in het XML-document bewaard, maar dit levert het voordeel op dat XML-documenten technologie-onafhankelijk zijn. Tenslotte zorgt de nesting van semantische tags er ook voor dat het XML-document zelf informatie en kennis (contextuele informatie) bevat die nodig is om achteraf te archiefdocumenten te begrijpen.

### 3.3 DTD en XML Schema

De XML-syntax laat de gebruiker toe een eigen documentmodel te ontwikkelen met zelf gekozen elementen, attributen en structuur. Dit documentmodel is eigenlijk een verdere verfijning van de XML-specificatie waarbij een eigen lexicon voor een (type-)document wordt samengesteld. Het documentmodel bevat bijkomende regels die naast de algemene XML-syntaxregels bij documentcreatie worden gevolgd. Het model van een XML-document kan formeel worden vastgelegd in een Document Type Definition (DTD) of een XML Schema. Een DTD en/of een XML Schema documenteren het documentmodel, kunnen naar de ontvangende computer of de gebruiker worden gecommuniceerd en zijn bruikbaar als controlemiddel.

De XML-specificatie 1.0 nam de mogelijkheid om DTD's samen te stellen over van SGML. DTD's zijn bijgevolg nog sterk documentgericht en worden zelf niet volgens de XML-syntax opgesteld. In een DTD worden de elementen met hun attributen, samenstelling, relatie en frequentie gedefinieerd. Dit komt in grote mate neer op de structuur en de nesting van het document. Met een DTD kan dus enkel gecontroleerd worden of de juiste tags en de correcte nesting worden gecontroleerd. In beperkte mate kan een DTD de frequentie van de elementen definiëren en verifiëren.

XML Schema's zijn in vergelijking met DTD's niet zuiver op documenten gericht, maar bieden ook een aantal mogelijkheden die werden overgenomen uit de databankwereld. In tegenstelling tot DTD's bieden XML Schemas wel de mogelijkheid om het datatype, de veldlengtes, de exacte minimum- en maximumfrequentie van de elementen te definiëren. Met XML Schema's kunnen er beperkingen op de geldige invoermogelijkheden worden opgelegd, zodat bij de creatie al kwaliteitscontrole mogelijk is. XML Schema is gebaseerd op de XML-syntax en ondersteunt namespaces (zie 4. De XML-familie).

---

<sup>12</sup> Het is eveneens mogelijk om met behulp van een stylesheet semantiek aan XML-data toe te kennen, maar hierdoor verliest het XML-document een stukje onafhankelijkheid en wordt niet optimaal gebruik gemaakt van zijn zelfbeschrijvende mogelijkheden.

De functie van een DTD of XML Schema kan met die van een dataschema van databanken worden vergeleken. De DTD of XML Schema bevat de definitie van alle mogelijke geldige documenten of legt vast welke beperkingen er gelden, wat de voorgedefinieerde waarden zijn, enz. Een DTD of een XML Schema documenteert eveneens het XML-document. Ze verschaffen expliciete informatie over de betekenis, het gebruik en de functie van de elementen zodat auteur en lezer hetzelfde begrijpen.

Een DTD of een XML Schema communiceert meta-informatie over het document naar de parser<sup>13</sup>. Die meta-informatie bevat onder andere de volgorde en de nesting van de tags, de attribuutwaarden, de datatypen en standaardwaarden van de elementen, de namen van externe bestanden waarnaar wordt verwezen en de entiteiten in het bestand.

Het opstellen van een DTD of een XML Schema komt in grote mate neer op het samenstellen van het documentmodel of prototype. Door vervolgens tags aan het archiefdocument toe te voegen wordt als het ware het archiefdocument gemodelleerd aan de DTD of het XML Schema en wordt het documentmodel toegepast.

Tabel 1: DTD vergeleken met XML Schema.

|                               | DTD                              | XML Schema                            |
|-------------------------------|----------------------------------|---------------------------------------|
| <b>ALGEMEEN</b>               |                                  |                                       |
| syntax                        | Extended Backus Naur Form (EBNF) | XML                                   |
| ondersteuning namespaces      | nee                              | ja                                    |
| uitbreidbaar                  | nee                              | ja                                    |
| overerving                    | nee                              | ja                                    |
| oriëntatie                    | documenten                       | documenten - data object georiënteerd |
| opslagplaats tov XML-document | intern/extern                    | intern/extern                         |
| <b>VALIDATIEFUNCTIES</b>      |                                  |                                       |
| structuur archiefdocument     | ja                               | ja                                    |
| correcte tags                 | ja                               | ja                                    |
| nesting tags                  | ja                               | ja                                    |
| datatypen                     | nee                              | ja                                    |
| veldlengtes                   | nee                              | ja                                    |
| frequentie elementen          | beperkt                          | ja                                    |

#### DTD voor adresboek.xml

```

<!ELEMENT adresboek (contactpersoon*)>
<!ELEMENT contactpersoon (naam, voornaam*, adres*, telefoon*, fax*, e-mail*)>
  <!ELEMENT naam (#PCDATA)>
  <!ELEMENT voornaam (#PCDATA)>
  <!ELEMENT adres (straat*, nummer*, bus*, postnummer*, gemeente*)>
    <!ELEMENT straat (#PCDATA)>
    <!ELEMENT nummer (#PCDATA)>
    <!ELEMENT bus (#PCDATA)>
    <!ELEMENT postnummer (#PCDATA)>
    <!ELEMENT gemeente (#PCDATA)>
  <!ELEMENT telefoon (#PCDATA)>
  <!ELEMENT fax (#PCDATA)>
  <!ELEMENT e-mail (#PCDATA)>

```

<sup>13</sup> parsen: ontleden van de structuur van een XML-document

## XML Schema voor adresboek.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="adresboek">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="contactpersoon" type="contactpersoonType" minOccurs="0"
          maxOccurs="unbounded"/></xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="contactpersoonType">
    <xs:sequence>
      <xs:element name="naam" ref="naam" minOccurs="1" maxOccurs="1"/></xs:element>
      <xs:element name="voornaam" ref="voornaam" minOccurs="0" maxOccurs="5"/></xs:element>
      <xs:element name="adres" type="adresType" minOccurs="0" maxOccurs="2"/></xs:element>
      <xs:element name="telefoon" ref="telefoon" minOccurs="0" maxOccurs="3"/></xs:element>
      <xs:element name="fax" ref="fax" minOccurs="0" maxOccurs="2"/></xs:element>
      <xs:element name="e-mail" ref="e-mail" minOccurs="0" maxOccurs="2"/></xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="adresType">
    <xs:sequence>
      <xs:element name="straat" ref="straat" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element name="nummer" ref="nummer" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element name="bus" ref="bus" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element name="postnummer" ref="postnummer" minOccurs="0"
        maxOccurs="1"/></xs:element>
      <xs:element name="gemeente" ref="gemeente" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="bus" type="xs:string"/></xs:element>
  <xs:element name="e-mail" type="xs:string"/></xs:element>
  <xs:element name="fax" type="xs:string"/></xs:element>
  <xs:element name="gemeente" type="xs:string"/></xs:element>
  <xs:element name="naam" type="xs:string"/></xs:element>
  <xs:element name="nummer" type="xs:string"/></xs:element>
  <xs:element name="postnummer" type="xs:integer"/></xs:element>
  <xs:element name="straat" type="xs:string"/></xs:element>
  <xs:element name="telefoon" type="xs:string"/></xs:element>
  <xs:element name="voornaam" type="xs:string"/></xs:element>
</xs:schema>

```

### 3.4 Voornaamste syntaxregels

De voornaamste XML-regels zijn:

- elk XML-bestand moet één basis XML-element (het root element of het document element) bevatten. Alle andere XML-elementen van het documenten worden genest binnen dit basiselement. Een XML-document kan slechts één rootelement bevatten.
- XML-tags omringen de elementen. De tags beginnen met het karakter < en eindigen met >. Elk element heeft een begin- en een eindtag. Een elementtag bestaat uit <elementnaam>. De elementnaam van begin- en eindtag is identiek. De eindtag onderscheidt zich van de begintag door de toevoeging van een schuine streep voor de elementnaam (bijv. <elementnaam></elementnaam>).
  - geen beperking op de lengte van de tagnamen
  - geldige karakters voor elementnamen:

- alle letters, alle cijfers en de volgende leestekens: dubbelpunt, onderstreep, koppelteken en punt.
- beperkingen: spaties zijn niet toegaten. Tagnamen mogen enkel met een karakter, onderstreep of dubbelpunt beginnen (niet met een cijfer, leesteken of "xml" beginnen).
- de tags moeten in correcte volgorde worden afgesloten. Er is geen beperking op de tagniveaus of structuurdiepte van het document
- een aantal tekstkarakters zijn gereserveerd voor markupdoeleinden en kunnen niet als inhoudskarakter worden gebruikt. Deze karakters worden in de content van het XML-bestand vervangen door entiteiten.

| Tekstkarakter | Entiteit |
|---------------|----------|
| <             | &lt;     |
| >             | &gt;     |
| &             | &amp;    |
| "             | &quot;   |
| '             | &apos;   |

- XML is hoofdlettergevoelig: respecteer het onderscheid tussen hoofd- en kleine letters
- de attribuutwaarden staan tussen aanhalingstekens.

### 3.5 Validatie

De XML-specificatie 1.0 schrijft voor dat elk XML-document de XML-syntaxregels van de Recommendation moet respecteren. Een document dat hieraan voldoet, is *well-formed*. Elk XML-document is dus per definitie *well-formed*. Een document dat zondigt tegen één van de syntaxregels is geen XML-document.

De XML-specificatie voorziet ook nog een tweede categorie XML-documenten: de geldige of *valid* XML-documenten. Deze documenten zijn niet alleen *well-formed*, maar zijn eveneens volgens een bepaalde DTD of XML Schema opgesteld. *Valid* XML-documenten voldoen dus niet alleen aan de algemene XML-syntaxregels, maar beantwoorden ook aan het documentmodel dat in een DTD of XML Schema is gedefinieerd.

Voor de validering van XML-documenten wordt een *validating* XML-processor of -parser ingeschakeld. XML-applicaties werken doorgaans niet rechtstreeks op de documenten, maar via een parser die hen toegang verschaft tot de inhoud en de structuur. Een parser ontleedt een XML-document en controleert of het document conform een bepaald documentmodel is opgesteld. Validatie op basis van een XML Schema is krachtiger dan op basis van een DTD. Een *non-validating* parser controleert het XML-document enkel op *well-formedness*.

### 3.6 Eigenschappen

De lijst met eigenschappen van XML-documenten is vrij uitgebreid. Hier worden in de eerste plaats de eigenschappen beschreven die XML geschikt maken voor toepassing bij digitale archivering.

XML:

➤ ... *is het resultaat van een standaardiseringsproces en heeft de status van publieke defacto standaard*<sup>14</sup>

XML werd op 10 februari 1998 als Recommendation van het W3C (*World Wide Web Consortium*) gepubliceerd. Een tweede editie van de specificatie XML 1.0 verscheen op 6 oktober 2000. De tweede editie is de coördinatie van de eerste editie en de erratalijst. Versie 1.1 zit momenteel in het stadium van working draft (versie van 25 april 2002). Het W3C is een consortium dat zich bezig houdt met standaardisatie voor het web. Het W3C is geen officiële standaardiseringsinstantie. De W3C Recommendations worden gepubliceerd en zijn vrij en gratis beschikbaar. Een specificatie doorloopt een heel traject alvorens het een W3C Recommendation wordt: working draft → last call working draft → candidate recommendation → proposed recommendation → recommendation. De opeenvolgende stadia zijn gekoppeld aan adviezen, commentaren en goedkeuringen van andere W3C werkgroepen, het W3C team, de leden en het publiek. Substantiële wijzigingen aan een Recommendation moeten het hele creatieproces (publieke commentaren, adviezen en goedkeuringen inbegrepen) van een Recommendation doorlopen<sup>15</sup>. Eén van de doelstellingen bij het ontwerp van XML was compatibiliteit met SGML. SGML werd in 1986 een officiële ISO-standaard (ISO-8879). SGML-applicaties moeten in staat zijn om XML-bestanden in te lezen.

➤ ... *is uitbreidbaar*

XML is een metataal waardoor het mogelijk is dat elke gebruiker zijn eigen XML-lexicon creëert.

➤ ... *is platformafhankelijk*

XML-bestanden kunnen tussen diverse computerplatformen worden uitgewisseld.

➤ ... *is softwareafhankelijk*

XML-bestanden zijn niet gebonden aan één bepaalde computerapplicatie of versie voor creatie, raadpleging of editeren.

➤ ... *is vrij van patentrechten*

XML is niet het eigendom van één bepaalde producent. Niemand kan zich de eigenaar van XML noemen of op eigen initiatief de XML-specificatie wijzigen.

➤ ... *is niet zo versiegebonden zoals andere markuptalen*

De gebruiker definieert zijn eigen tags en attributen. De tags en attributen kunnen bijgevolg niet verouderd raken bij versieveranderingen.

<sup>14</sup> Het DLM-forum onderscheidde in 1997 drie niveaus van standaarden: 1. defacto standaarden, 2. publiek beschikbare specificatie, 3. juridische standaarden. Volgens deze indeling zit XML in de tweede groep. (*Guidelines on best practices for using electronic information*, p. 50).

<sup>15</sup> [http://www.w3c.org/About the World Wide Web Consortium \(W3C\).htm](http://www.w3c.org/About%20the%20World%20Wide%20Web%20Consortium%20(W3C).htm) ; [http://www.w3c.org/5 Technical Reports.htm](http://www.w3c.org/5%20Technical%20Reports.htm)

### ➤ ... *is transformeerbaar*

Bij output kunnen de XML-data in een andere structuur en in om het even welk outputformaat worden getoond.

De structuur van de opgeslagen data hoeft niet noodzakelijker wijze gevolgd te worden bij output. Voor de transformatie is een stylesheet (HTML: XSL(T); PDF: XSL(FO) of JAVA nodig. Bij de transformatie kunnen ook verschillende acties worden uitgevoerd: verplaatsen van de tekstelementen, toevoegen of weglaten van tekst, sorteren op basis van één element, nummeren van items (bijv. kopteksten), uitvoeren van berekeningen, enz.

De inhoud van XML-bestanden kan in verschillende (toekomstige) bestandsformaten worden gepubliceerd. Momenteel worden er op basis van XML-bestanden vooral (X)HTML- en PDF-documenten gecreëerd, maar het is ook mogelijk om XML-content om te zetten naar toekomstige doeltechnologieën die momenteel nog onbekend zijn.

Hetzelfde XML-bronbestand kan naar meerdere outputformaten worden omgezet. XML is bijgevolg een opslagformaat op basis waarvan verschillende rendities kunnen gebeuren.

### ➤ ... *bestaat enkel uit tekstkarakters (Unicode)*

XML is een formaat waarbij alle markup- en inhoudsgegevens als tekstuele karakters worden geëncodeerd. XML-bestanden zijn geen binaire bestanden die een welbepaalde softwaretoepassing nodig hebben om de binaire data te interpreteren en hun inhoud te reconstrueren. Hierdoor kunnen XML-bestanden worden geopend in elke applicatie die tekstuele karakters kan inlezen. XML-bestanden zijn bijgevolg eenvoudig, gemakkelijk uitwisselbaar en digitaal duurzaam. XML beantwoordt hiermee aan de vereiste van een zo laag mogelijke platform- en softwareafhankelijkheid. De enige afhankelijkheid is het correct omzetten van de bits naar Unicodekarakters.

XML gebruikt Unicode. XML-documenten kunnen karakters bevatten uit elke taal die door Unicode wordt ondersteund. Hierdoor krijgt XML het kenmerk van *universeel* te zijn.

Binaire bestanden kunnen in een XML-bestand worden opgeslagen, maar de binaire data moeten eerst als tekstkarakters geëncodeerd zijn. De encoding moet gedocumenteerd worden.

Volledige tekstuele encoding is belangrijk voor uitwisseling. XML-documenten kunnen hierdoor gemakkelijk verwerkt worden door computers en zijn bovendien hierdoor ook gemakkelijk te begrijpen door mensen.

### ➤ ... *scheidt inhoud en opmaak*

XML laat gescheiden archivering van inhoud en opmaak toe. De inhoud wordt in het XML-bestand bewaard, de opmaak van het archiefdocument kan in een CSS of XSL(T)-bestand worden vastgelegd. Het scheiden van inhoud en opmaak biedt voordelen op het vlak van herbruikbare informatie en grootschalig informatiebeheer. Door de media onafhankelijke opslag en de late koppeling van XML-data met presentatie in het productieproces, kunnen XML-documenten naar elk gewenst bestandsformaat (XML, HTML, VoiceXML, WML, PDF, TeX, RTF, enz.) voor output getransformeerd worden. Deze media onafhankelijke publishing is mogelijk in 'real-time'. Hetzelfde XML-bronbestand kan eveneens in combinatie met verschillende stylesheets worden gebruikt. Het ontbreken van hard coded opmaakgegevens in de data zorgt mee voor een gemakkelijke uitwisseling.

### ➤ ... *is zelfbeschrijvend*

XML-documenten zijn zelfbeschrijvend en zijn hierdoor gemakkelijk te interpreteren door mensen of door computers.

De XML-tags identificeren en labelen de componenten of velden waaruit de informatie is samengesteld. De tags geven het begin en einde van elk veld aan, geven weer wat het datatype van de informatie is en geven uitleg over de betekenis. De tagattributen kunnen de documentatomen verder beschrijven. De geneste structuur is een weerspiegeling van de relatie tussen de verschillende elementen. XML laat eveneens de inkapseling van metadata of kontekstuele informatie toe waardoor XML-bestanden zelfbeschrijvend zijn. Voor de semantiek, interpretatie of visualisatie

is men bijgevolg niet afhankelijk van externe documentatie of bronnen. Een XML-bestand kan als envelop of container functioneren die samen met het archiefdocument ook zijn beschrijving bevat. Dit is een invloed van het object-georiënteerd denken waarbij het gearriveerd computerbestand zowel het archiefdocument als zijn metadata bevat. XML-documenten kunnen hierdoor gemakkelijk geïnterpreteerd en ontleed worden.

XML-documenten zijn ook zelfbeschrijvend in IT-termen. Alle gegevens die computers nodig hebben voor de ontleding en de interpretatie van XML-data zit in de bestanden zelf vervat. Hierdoor is verwerking mogelijk zonder menselijke tussenkomst.

#### ➤ ... *biedt verschillende zoekmogelijkheden*

Bij XML zijn de zoekmogelijkheden niet beperkt tot full text zoekopdrachten, maar is het ook mogelijk om documenten terug te vinden op basis van hun structuur, metadata en semantische tags.

#### ➤ ... *bewaart de structuur van een archiefdocument op een expliciete en formele wijze*

XML is geschikt voor een gestructureerde opslag van informatie. De markuptags worden volgens een bepaalde structuur genest. De hiërarchische structuur van de elementen weerspiegelt de relaties tussen de componenten van het archiefdocument en is expliciet in het document aanwezig. Een DTD of een XML Schema kan de structuur van een archiefdocument verder documenteren.

#### ➤ ... *is flexibel*

In de opslag van XML-documenten zit niet verweven wat er met de data moet gebeuren. De verwerking van XML-data is volledig afhankelijk van de applicatie. Hierdoor kan de dataopslag geen enkele beperking stellen op het toekomstig gebruik van de computerdata.

#### ➤ ... *is een combinatie van data- en documentcentrisch*

Met XML zijn de grenzen tussen databankgerichte (regelmatige structuur, kleine veldlengtes, beheer) en documentgerichte (onregelmatige structuur, lange veldlengtes, menselijke consumptie) toepassingen vervaagd. Beide werelden evolueren immers naar elkaar toe, want tekstuele informatie is in de realiteit meestal semi-gestructureerd. Documenten krijgen alsmaar meer structuur (transmissie, standaardisatie) en databanken moeten in staan zijn om data met onregelmatige structuur te bevatten. XML is in staat om het hele spectrum van zwak tot sterk gestructureerde informatie te omspannen.

#### ➤ ... *kent grote commerciële ondersteuning*

Alle grote softwareleveranciers ondersteunen XML<sup>16</sup>. XML kan worden gebruikt op Windows, MacOS en Unix. Er zijn nu al meer XML-tools voor handen dan er ooit voor SGML beschikbaar waren (o.a. JAVA en PERL-based)<sup>17</sup>. XML wordt ook in bestaande softwarepakketten geïntegreerd (bijv. webbrowsers<sup>18</sup>, MS Office 2000<sup>19</sup>, MS Office XP<sup>20</sup>, OpenOffice.org<sup>21</sup>, WordPerfect 9, enz.).

<sup>16</sup> <http://www.sun.com/software/xml/>; <http://www.novell.com/products/edirectory/dirxml/>; <http://www-106.ibm.com/developerworks/xml/>; <http://xml.coverpages.org/xmlSupport.html>

<sup>17</sup> Voor een overzicht van XML-tools: <http://www.w3.org/XML/#software>; <http://www.oasis-open.org/cover/publicSW.html>; <http://xml.coverpages.org/xml.html#xmlSoftware>; [http://www.xml.org/xml/resources\\_focus\\_programming.shtml](http://www.xml.org/xml/resources_focus_programming.shtml)

<sup>18</sup> Internet Explorer: Versie 5.0 is gebaseerd op de XML Recommendation 1.0 en op een working draft van de XSL-specificatie. Met IE is client-side XSL verwerking dus mogelijk.

Netscape: Versie 6.0 is volledig conform XML 1.0, maar ondersteunt XSL nog niet omdat op het ogenblik van de release van Netscape de XSL-specificatie nog niet was afgerond.

<sup>19</sup> MS Office 2000: gebruik van XML data islands wanneer Excel-, Powerpoint- en Worddocumenten als webpagina's worden bewaard. Hierdoor zijn de documenten in webbrowsers raadpleegbaar. Data islands zijn blokken xml content in een HTML-pagina. Hiervoor wordt gebruik gemaakt van de niet officiële xml-tag.

➡ ... heeft ook nadelen:

- lage performantie:
  - door de volledige tekstuele encoding van de data zijn XML-documenten soms langdradig en nemen ze meestal meer bestandsomvang in beslag dan bestanden met binaire encoding. Omwille van een betere performantie worden XML-documenten bij netwerkuitwisseling veelal gecomprimeerd ((g)zip, http 1.1, modems, enz.). Bij XML-documenten kunnen hoge compressiegraden worden bereikt. De compressie is lossless.
  - de parsing en de omzettingsoperaties van en naar XML kosten tijd
- nog niet veel clientcomputers zijn XML en/of XSL-compatibel zodat bij de meeste XML-applicaties parsing en transformaties op de server dienen te gebeuren

### 3.7. Opslag en opzoeken van XML-documenten

XML-documenten kunnen op twee wijzen worden opgeslagen: in het bestandssysteem van een courant besturingssysteem of in een databank. De keuze inzake opslag is belangrijk voor de manier waarop achteraf XML-documenten worden teruggevonden. Bij XML komt het er niet alleen op aan om het nodige XML-document te vinden, maar ook om toegang te hebben tot de dataobjecten binnen een XML-document.

#### 3.7.1 In een bestandssysteem

Bewaring binnen een bestandssysteem is eenvoudig en heeft als voordeel dat de XML-data snel toegankelijk zijn. De XML-documenten worden gewoon als afzonderlijke computerbestanden met de extensie \*.xml in bepaalde mappen opgeslagen. De mogelijkheden om XML-documenten op te sporen zijn beperkt tot zoeken op basis van map- en bestandsnamen en tot full text zoekopdrachten. De resultaten van full-textzoekopdrachten zullen echter niet al te zuiver zijn. Full-textzoekopdrachten maken geen onderscheid tussen markupkarakters en inhoudskarakters. De nadelen van deze opslagmethode zijn de limieten van de besturingssystemen: beperkingen op maximale bestandsomvang, moeilijke navigatie in grote bestanden, geen versiecontrole, geen multi-user toegang, lage beveiliging, enz. XML-documenten in een bestandssysteem bewaren is eenvoudig, maar is niet aangewezen voor grote archiveringstoepassingen. Voor geen al te grote hoeveelheden XML-documenten kan deze opslagmethode echter volstaan.

#### 3.7.2 In een databank

Voor grote toepassingen is de opslag in een databank meer aangewezen. Databanken bieden doorgaans ook een oplossing voor de tekortkomingen van gewone besturingssystemen. Databanken bieden eveneens meer zoekfunctionaliteiten. XML-documenten worden doorgaans in relationele, object geïntegreerde of native XML databanken bewaard.

<sup>20</sup> MS Office XP: import van XML documenten in Excel en Access, bewaring van rekenbladen, tabellen, queries en views in Excel en Access als XML documenten, gebruik van XML smart tags in Word, Excel en Outlook. In Outlook worden views als XML-bestanden opgeslagen.

<sup>21</sup> OpenOffice.org: met uitzondering van Math gebruiken alle applicaties binnen OpenOffice.org (Writer, Calc, Draw, Impress, Writer global document) hetzelfde formaat dat op XML is gebaseerd. Binnen de OpenOffice.org-bestanden worden zowel metadata (ongecomprimeerd) als de bitstream van de digitale objecten (ZIP-compressie) bewaard. De OpenOffice.org bestanden zijn samengesteld uit: meta.xml, styles.xml, content.xml, settings.xml, meta-inf/manifest.xml en de mappen pictures, dialogs, basic en obj. (<http://www.openoffice.org>)

### 3.7.2.1 Opslag in een relationele databank

Aangezien XML-documenten een hiërarchische structuur hebben, kunnen ze niet als dusdanig in een relationele databank worden bewaard. De XML-documenten worden als BLOB's in de databank opgenomen of de XML-inhoud wordt naar een relationeel datamodel vertaald. Hierbij worden de elementen naar tabellen en kolommen gemapt en hun relaties worden door verschillende jointypes weer gegeven. Veel commerciële relationele databanken bevatten al bovenop de relationele motor een XML-mapping laag voor het bewaren en zoeken van XML-data. Ook eventuele XSLT transformaties, XPathexpressies en XQueries moeten worden omgezet. In XML-applicaties moet achteraf ook de output opnieuw als een XML-document worden verpakt. Al deze omzettingen bij input, bevraging en output kosten tijd. Toch levert een relationele databank meestal sneller zoekresultaten af dan een object-georiënteerde databank. Relationele databanken zijn goed bruikbaar wanneer de XML-documenten niet al te complex zijn of goed gestructureerd zijn. De bewaring van slecht gestructureerde of ongestructureerde documenten in een RDBMS is niet efficiënt want er gaat veel opslagruimte verloren.

### 3.7.2.2 Opslag in een object-georiënteerde databank

Het hiërarchisch datamodel van object-georiënteerde databanken sluit beter bij XML-documenten aan dan dat van relationale databanken. In een object-georiënteerde databank worden evenmin de XML-documenten zelf als één entiteit bewaard, maar worden de nodes als afzonderlijke objecten in een bepaalde hiërarchie opgeslagen. XML-documenten worden teruggevonden en gemaniupuleerd op basis van XML-queries op de nodes. Elke node kan afzonderlijk worden verwerkt. Het gebruik van object-georiënteerde databanken wordt aanbevolen in de gevallen waaraan de XML-documenten complexe relaties en een structuur met grote diepte hebben.

### 3.7.2.3 Opslag in een native XML databank

Native XML databanken zijn speciaal ontworpen om XML-documenten te bewaren. Het grote verschil met andere databanksystemen is dat hun intern model gebaseerd is op XML-documenten en niet op een relationeel of object-georiënteerd dataschema. De XML-documenten vormen de basiseenheid van native XML databanken. Native XML databanken respecteren de orde en interne structuur van de XML-documenten. De XML-engine zoekt in querytalen die op XML zijn gebaseerd (bijv. XQL, XML-QL, XMAS) en retourneert de data als XML-documenten. In een XML native databank worden de XML-documenten lineair opgeslagen. Er gaat dus geen tijd verloren aan mappings, joinopdrachten en XML-wrapping van de output. De XML-documenten kunnen rechtstreeks in de databank worden gestockeerd en bevroegd. De performantie van native XML databanken is bijgevolg meestal groter dan relationele of object-georiënteerde databanken. Deze databanktechnologie is nog maar vrij jong zodat er nog maar een beperkt aantal tools voor handen zijn.

## 4. De XML-familie

Een XML-document doet op zichzelf niets want in essentie is het niets meer dan een geheel van Unicodekarakters, verpakt in XML-tags. XML wordt gebruikt voor het beschrijven van data die wordt gebruikt voor het structureren, bewaren en versturen van informatie. Om presentatie en functionaliteiten aan XML-documenten toe te voegen, is er nood aan andere mechanismen. Rond XML worden een heleboel andere standaarden gedefinieerd die in combinatie met XML worden gebruikt. Samen met XML vormen deze standaarden de XML-technologie<sup>22</sup>. De meeste standaarden worden eveneens door het W3C vastgelegd en vormen de basis voor de XML-toepassingen die door producenten worden ontwikkeld en in hun producten worden geïmplementeerd.

---

<sup>22</sup> XML encryption, XML key management en XForms zijn nog in ontwikkeling.

|                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CSS</b><br><i>Cascading StyleSheet</i><br>Versies: Level 1 (CSS1): W3C Recommendation (17 december 1996),<br>herzien op 11 januari 1999; Level 2 (CSS2):<br>W3C Recommendation (12 mei 1998) | CSS is de stylesheettaal voor HTML, maar kan eveneens in combinatie met XML worden gebruikt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>XML Base</b><br>Versies: W3C Recommendation 1.0 ( 27 juni 2001)                                                                                                                              | Faciliteit voor het produceren van base URI's om delen van XML documenten aan te duiden.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>XML Signature</b><br>Versies: W3C Recommendation 1.0 (12 februari 2002)                                                                                                                      | Definieert een kader gebaseerd op XML om digitale handtekeningen aan bronnen op het web toe te voegen.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>XML Query</b><br>Versies: W3C Working draft (30 april 2002)                                                                                                                                  | Querytaal voor XML-documenten, ook wel XQL genoemd. Nog niet formeel vastgelegd.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>XML Schema</b><br>Versies: W3C Recommendation 1.0 (2 mei 2001)                                                                                                                               | zie 3.3 DTD en XML Schema                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>XPointer</b><br>Versie Candidate Recommendation (11 september 2001)                                                                                                                          | toepassing gebaseerd op XPath voor het wijzen naar bepaalde onderdelen van XML-documenten.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>XLink</b><br><i>XML Linking Language</i><br>Versie: Recommendation 1.0 (27 juni 2001)                                                                                                        | biedt de mogelijkheid om hyperlinks aan XML-documenten toe te voegen zoals in HTML-pagina's, maar met XLink is het ook mogelijk om meer gesofisticeerde links te maken<br><link xml:link="simple" href="locator">tekst</link>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>XSLT</b><br><i>XSL Transformations</i><br>Versies: W3C Recommendation 1.0 (16 november 1999)                                                                                                 | XSLT wordt hoofdzakelijk gebruikt voor de transformatie van data van de ene XML structuur naar een andere. Zo kan de XML-data in een ander formaat of een andere structuur worden weergegeven. In die zin is XSL vergelijkbaar met CSS, maar XSL is krachtiger (o.a. selecteren van elementen, filteren, sorteren transformatie van de XML boom, data afhankelijke opmaak)                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>XSL</b><br><i>eXtensible Stylesheet Language</i><br>Versies: W3C Recommendation 1.0 (15 oktober 2001)                                                                                        | XSL voegt aan XSLT de mogelijkheid tot <i>formatting objects</i> en <i>formatting properties</i> toe. XSL bestaat uit drie delen die meestal met elkaar worden gecombineerd: XPath, XSLT en XSL Formatting Objects. Via XPath heeft men toegang tot de nodes van een XML-document, XSLT zorgt voor de koppeling met templates terwijl XSL de formatting objects bevat. XSL-FO (Formatting Objects) wordt ondermeer gebruikt om PDF-documenten te creëren op basis van XML-data. De elementen in de XSL:FO vocabulary definiëren zaken zoals marges, headers, footers, uitlijning, spaties, word wrapping, enz. Eerst wordt de XML inhoud naar formatting objecten geconverteerd, vervolgens zet een FOP-tool deze naar PDF om. De presentatiemogelijkheden binnen XSL-FO zijn vergelijkbaar met die van HTML en CSS. |
| <b>XPath</b><br>Versies: W3C Recommendation 1.0 (16 november 1999)                                                                                                                              | XPath is een string syntax gebaseerd op URI's voor het maken van adressen naar XML-documenten, -elementen of -attributen. XPathexpressies worden gebruikt voor het lokaliseren van info in XML documenten. XPath baseert zich op de hiërarchische positie van de componenten in de tree. De nodes worden als unieke ID's gebruikt. XPath biedt geen andere queryfunctionaliteiten zodat XPath niet echt een querytaal is. XPath wordt ondermeer gebruikt door XSLT en XPointer                                                                                                                                                                                                                                                                                                                                       |
| <b>MathML</b><br><i>Mathematical Markup Language</i><br>Versie: 1.0 (juli 1999), 2.0 (februari 2001)                                                                                            | wordt gebruikt voor de markup van wiskunde op het web.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>DOM</b><br><i>Document Object Model</i>                                                                                                                                                      | Gemeenschappelijke <i>Application Programming Interface</i> (API: verzameling standaardfuncties) waarmee de elementen, de                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

|                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Versies: Level 1 (oktober 1998) en Level 2 (november 2000)                                                                                           | structuur, de evenementen, enz. afzonderlijk worden benaderd vanuit verschillende programmeertalen en/of scripts. DOM levert ook een standaard interface voor andere software om XML-documenten te verwerken.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>RDF</b><br><i>Resource Description Framework</i><br>Versies: W3C Recommendation (22 februari 1999)                                                | W3C's metadatastandaard RDF bevat een standaardlexicon en coderingstechnieken om metadata te koppelen aan gelijk welke bron op het web. RDF moet het zoeken op het Internet verbeteren. Een RDF-toepassing is bijvoorbeeld een bestand die in de rootmap van een website wordt geplaatst met metadata over de website. De gebruiker beschrijft objecten, en kent er attributen/eigenschappen aan toe. Hij kan ook statements over de objecten vastleggen. Deze statements kunnen bijvoorbeeld relaties tussen objecten zijn.                                                                                                                                                                                                                                                                                                   |
| <b>XML Namespaces</b><br>(januari 1999)                                                                                                              | Met XML kan iedereen zijn eigen vocabularium samenstellen. De kans dat elementen met dezelfde tags een verschillende betekenis hebben, is dus bijzonder groot. Om tagbenamingsconflicten en verdere uitwisselingsproblemen te voorkomen, is XML namespaces ontwikkeld. Hiermee kan het vocabularium van de gebruiker worden geïdentificeerd zodat ambigue informatie wordt vermeden. De namespace wordt geïdentificeerd door een URI (URL of URN). De tags die deel uitmaken van een bepaalde namespace hebben een gemeenschappelijke prefix. Binnen hetzelfde XML-document kunnen ook verschillende namespaces worden gebruikt. Als er een default namespace wordt gedeclareerd dan is dit geldig voor heel het document. Een namespace die een attribuut is van een parentelement heeft ook betrekking op de childelementen. |
| <b>SAX</b><br><i>Simple API for XML</i><br>Versie: 2.0.1                                                                                             | Java API ontwikkeld om met XML te worden gebruikt. Voordelen: eenvoudig, het volledige document moet niet in het geheugen worden geladen. Nadelen: niet ondersteund door alle browsers, enkel voor XML-documenten, complexe zoekopdrachten kunnen moeilijk zijn. SAX is niet ontwikkeld binnen de schoot van het W3C <sup>23</sup> . Inmiddels bestaat SAX ook voor andere omgevingen van Java.                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>SVG</b><br><i>Scalable Vector Graphics</i><br>Versie: W3C Recommendation 1.0 (4 september 2001), W3C Candidate Recommendation 1.1 (30 april 2002) | Specificatie gebaseerd op XML voor de verspreiding van vectoriële afbeeldingen op het web.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>XTM</b><br><i>XML Topic Maps</i><br>Versie: TopicMaps.org specificatie 1.0 (2001), annex van ISO-13250 (oktober 2001).                            | XML namespace voor het samenstellen van een topic map. XML wordt als notatiewijze gebruikt, XML base URI's worden voor de links gebruikt. Een topic map bestaat uit topics die met elkaar geassocieerd zijn en geeft kennis weer.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

## 5. XML voor archiveringstoepassingen

### 5.1 XML als uitwisselingsformaat

Met XML kan gestructureerde informatie worden uitgewisseld tussen verschillende informatiesystemen. Dit levert een aantal belangrijke voordelen op: snellere uitwisseling, vermijden van fouten, uitwisseling kan automatisch verlopen, enz. XML heeft zijn eigenschap van geschikt uitwisselingsformaat onder meer te danken

<sup>23</sup> <http://www.saxproject.org/>

aan de tekstuele encoding van alle markup en inhoud, de scheiding van inhoud en opmaak, het gebruik van tags als delimiters, de expliciete aanwezigheid van structuur, enz.

XML-documenten zijn hierdoor heel gemakkelijk uitwisselbaar tussen verschillende toepassingen zonder dat de ontvanger de organisatie van de computerdata bij de verzender hoeft te kennen. Er is evenmin bijkomende code nodig. De enige vereiste is het volgen van de regels die bij de documentcreatie werden gevolgd en die in een DTD of XML Schema werden vastgelegd.

De toepassingsgebieden van XML als uitwisselingsformaat zijn legio. Content providers gebruiken momenteel XML al volop voor de verspreiding van digitale informatie. Zij maken hierbij ook gebruik van de mogelijkheid om hetzelfde brondocument in verschillende formaten bij de gebruiker af te leveren. Incompatibele computersystemen kunnen met elkaar interopereren op basis van XML<sup>24</sup>. Zo wordt XML al volop als intermediair formaat ingezet om data tussen verschillende typen databanken (bijv. van relationeel naar object-georiënteerd) of databanken met verschillende structuur uit te wisselen. In de databankomgeving vervangen XML interfaces meer en meer ODBC. Momenteel wordt binnen IETF gewerkt aan een XML-uitwisselingsformaat voor e-mails tussen verschillende applicaties als variatie op RFC822<sup>25</sup>.

In de archiefcontext is XML inzetbaar als transmissieformaat tussen de archiefvormer en de archiefinstelling/dienst. XML kan bijvoorbeeld gebruikt worden wanneer de archiefinstelling/dienst bepaalde softwaretoepassingen van de archiefvormer niet ondersteunt, maar toch de documenten met archiefwaarde archiveert. De archiefvormer brengt de documenten met archiefwaarde als XML-data buiten de oorspronkelijke applicatie en legt ze als dusdanig neer bij de archivaris. De archiefdienst kan de ontvangen XML-data naar hun eigen toepassing overbrengen of als XML-documenten archiveren.

## 5.2 XML als archiveringsformaat

In het tweede geval wordt XML als bestandsformaat voor archivering op lange termijn gebruikt. Als bestandsformaat benadert XML heel sterk de kenmerken van het ideale archiveringsformaat. Het ideaal archiveringsformaat is:

- stabiel en gestandaardiseerd
- vrij van patentrechten (publiek domein)
- zo eenvoudig mogelijk
- volledig zelf beschrijvend, niet afhankelijk van externe bronnen
- geschikt om gestructureerde en zelf gekozen metadata te bevatten
- volledig gedocumenteerd (open specificatie)
- niet applicatie- of platform gebonden, maar gemakkelijk uitwisselbaar
- wereldwijd verspreid gebruikt en geïmplementeerd
- gemakkelijk te controleren op fouten

XML komt als bestandsformaat in grote mate aan al deze eisen tegemoet. XML is hierdoor uitermate geschikt als applicatieonafhankelijk opslagformaat. De gebruiker is hierdoor niet meer gebonden aan versies, applicaties, platformen of zelfs producenten om toegang te hebben tot zijn computerdata. In tegenstelling tot producentgebonden opslagmethoden is men bijvoorbeeld niet meer afhankelijk van gesloten formaten of schema's om gegevens weg te schrijven of te openen. XML-applicaties zijn bijgevolg open en transparant en laten toe dat XML-data gemakkelijk in toekomstige ontwikkelingen inpasbaar zijn. De XML-documenten moeten immers geen migratieproces ondergaan wanneer ze naar een ander informatiesysteem worden overgebracht.

<sup>24</sup> Een voorbeeld XML-toepassing is bijvoorbeeld de CEVI E-Loket waarbij de gegevens ontvangen via on line invulformulieren, in XML-formaat naar de administratieve toepassingen worden doorgestuurd. (<http://www.cevi.be/cevi/site/realisaties/cevigem/default.htm>). Een ander voorbeeld is de *Universal Message Engine* voor de uitwisseling van gestructureerde boodschappen tussen heterogene applicaties binnen de overheid (bijv. Rijksregister).

<sup>25</sup> G. KLYNE, *An XML format for mail and other messages*.

De kansen op informatieverlies of schending van de authenticiteit en integriteit ten gevolge van migraties neemt bijgevolg gevoelig af. XML is geen applicatiegebonden formaat en levert de gebruiker dus grotere onafhankelijkheid op. Dezelfde XML-data zijn bruikbaar binnen desktopapplicaties, databanken, intranet- en Internettoepassingen. In die zin kunnen XML-data enigszins met SQL-data worden vergeleken<sup>26</sup>.

De autonomie van de XML-bestanden is een belangrijke troef. De onafhankelijkheid van XML-documenten aan externe factoren biedt garanties voor de reconstructie van digitale archiefdocumenten. XML-bestanden blijven wel digitaal. Er blijft software nodig voor raadpleging, maar een grotere mate van software-onafhankelijkheid is niet mogelijk of zelfs wenselijk. Het gaat tenslotte toch om digitale archiefdocumenten en er zal altijd software nodig blijven om vanuit de bewaarde bits de archiefdocumenten te reconstrueren.

In vergelijking met andere bestandsformaten met de status van defacto of open standaard, voldoet XML beter aan de vereisten van geschikt archiveringsformaat. Het Portable Document Format (PDF) van Adobe wordt ook regelmatig als archiveringsformaat naar voor geschoven, maar heeft toch een aantal nadelen in vergelijking met XML. PDF is eigendomsgebonden, bevat geen semantische markup, biedt minder zoekfunctionaliteiten en is niet "menselijk leesbaar". Een PDF-document kan niet zomaar met behulp van een teksteditor geïnterpreteerd worden want de gebruiker krijgt enkel nietszeggende tekens te zien. PDF-bestanden zijn met andere woorden meer software-afhankelijk dan XML-documenten.

De relatie XML-bestand en archiefdocumenten kan variëren van één-op-één, één-op-veel en veel-op-één. In XML gaat het hem in de eerste plaats om de logische structuur van computerdata. Eén XML-bestand kan één of meerdere archiefdocumenten bevatten en één archiefdocument kan over meerdere XML-bestand verspreid worden.

XML is in de eerste plaats aangewezen als archiveringsformaat voor tekstuele documenten. Naast de digitale duurzaamheidsgaranties is het een pluspunt van XML dat de structuur van de archiefdocumenten op een expliciete wijze mee gearchiveerd wordt. Het documentmodel kan formeel in een DTD of XML Schema worden vastgelegd. Na neerlegging bij de archiefdienst en voorafgaand aan opname in het archiefbeheerssysteem kan parsing van de archiefdocumenten tegen de DTD of XML Schema als een soort kwaliteitscontrole dienen. Door een DTD of een XML Schema aan het XML-document toe te voegen, neemt ook het zelfbeschrijvend karakter toe. Aangezien XML-documenten kunnen variëren van zwak tot sterk gestructureerd, kunnen zowel gewone tekstbestanden, rekenbladen, e-mails en databanken als XML-bestanden worden gearchiveerd. De hiërarchische tree van een XML-document kan evengoed relationele, object-georiënteerde als gewone tekstuele data van een brief, een e-mail of een rekenblad bevatten.

Aan het archiveren van tekstuele documenten als XML-bestanden zal meestal een migratiefase voorafgaan. Er wordt momenteel wel werk gemaakt van de implementatie van XML in computerapplicaties, maar zolang de applicaties de data zelf niet als XML-data bijhouden, zal migratie van native formaten naar XML zich opdringen. In veel gevallen betekent dit dat een statische versie van het digitale document wordt bewaard, en dat er een aantal dynamisch functionaliteiten verloren gaan. Typisch voor deze functionaliteiten is dat ze overwegend applicatiegebonden zijn en maar zelden archiefwaarde hebben. Het exporteren van computerdata als XML-bestanden begint ingang te vinden in computerprogramma's (bijv. MS Office). *OpenOffice* is bij de implementatie van XML wellicht het verst geëvolueerd en gebruikt XML als native formaat voor alle toepassingen binnen het pakket<sup>27</sup>.

Ondanks de toenemende verspreiding van XML bieden nog heel wat bestaande applicaties geen XML-functionaliteiten. Nochtans zijn met veel van deze computertoepassingen al documenten met archiefwaarde gecreëerd. Om deze archiefdocumenten van hun applicatieformaat naar XML om te zetten, dient men bijgevolg een oplossing te vinden zodat de archiefdocumenten als XML-documenten archiveerbaar zijn. Het inschakelen van één van de vele omzettingstools is een optie. Maar ook al beschikt men over een exportfunctionaliteit of een omzettingstool<sup>28</sup>, het is de vraag of het aangewezen is om er gebruik van te maken. De bestaande automatische exportfunctionaliteiten of omzettingstools laten immers zelden toe dat de gebruiker de tags en de structuur van het document definieert of dat er een extern documentmodel wordt gevolgd. Hierdoor is de

<sup>26</sup> Een voorbeeld hiervan uit de museumsector is Adlib. In versie 5.0 werd XML geïntroduceerd zodat bijvoorbeeld alle beschrijvingen van museumobjecten als XML-data kunnen weggeschreven worden.

<sup>27</sup> *OpenOffice.org XML File Format, 1.0*, juli 2002.

<sup>28</sup> MS Wordbestanden kunnen bijvoorbeeld naar XML worden omgezet met XML Spy. Standalone converters voor omzettingen WORD → XML zijn bijvoorbeeld Majix (<http://www.tetrasix.com>) en UpCast (<http://www.infinity-loop.de>). Tools geïntegreerd in MS Word zelf, zoals S4/Text (<http://www.i4i.com>), maken XML-documenten die opnieuw als Wordbestanden kunnen worden geopend en bewerkt. Omzettingen van Word naar XML kunnen ook gerealiseerd worden op basis van VBA, Omnimark of Perlscripts. Veel tools zetten Wordbestanden naar XML om via het Rich Text Format (RTF).

tagging en structurering van de XML-documenten meestal afhankelijk van de gebruikte export- of omzettingstool. Dit levert moeilijkheden op wanneer men één documentmodel voor één bepaald type archiefdocument wil toepassen, terwijl dit net één van de krachtpunten van XML is. Het gebruiken van verschillende export- en omzettingstools met vaste DTD's resulteert voor hetzelfde type archiefdocument in verschillende documentmodellen waarvoor verschillende DTD's, stylesheets en raadplegingsapplicaties nodig zijn. Het digitaal archiefbeheer wordt hierdoor opnieuw een stuk complexer.

Vooraleer men beroep doet op een bestaande exportfunctionaliteit of omzettingstool is het bijgevolg belangrijk om na te gaan in welke mate een vastgelegd documentmodel kan worden toegepast. De mogelijkheid tot configuratie van de export- of omzettingstool is een belangrijk criterium bij de toolkeuze. Voldoet de XML-opslag niet aan het vastgelegde documentmodel, dan zijn er twee opties. Men zet de XML-bestanden na de export naar het vastgelegde documentmodel (bijv. via XSLT) om of men voorziet een gecustomiseerde XML-wrapper die het gedefinieerde documentmodel wel onmiddellijk respecteert<sup>29</sup>. Deze tweede oplossing is efficiënter want er wordt een tussenstap uitgespaard en problemen in verband met het opsplitsen van XML-elementen worden vermeden.

De XML-wrapper moet ondermeer zorgen voor een tekstuele output, toepassing van de correcte encoding en de vervanging van gereserveerde karakters. Het spreekt voor zich dat voor de encoding best Unicode wordt toegepast, al is dat nog altijd geen evidentie. Unicode is relatief nieuw. Het merendeel van de tekstuele digitale informatie is niet in Unicode opgeslagen en slechts de jongste generaties besturingssystemen of applicaties ondersteunen Unicode<sup>30</sup>. Bij een retroactieve migratie naar XML is de kans ook groot dat gereserveerde karakters (&, <, >, ', ") als inhoudskarakters worden gebruikt. Deze karakters moeten bij de omzetting naar XML worden omgezet naar een entiteit. Door de toevoeging van semantische tags kan ook de bestandsomvang gevoelig toenemen. Bij een omzetting naar XML is het ten slotte ook belangrijk om aandacht te besteden aan de opslagplaats van DTD's of XML Schema's. Om het zelfbeschrijvend karakter en de autonomie van de XML-documenten te vergroten, zou men kunnen overwegen om deze in het XML-document als interne DTD's of interne XML Schema's op te nemen. Dit druist echter tegen de principes van herbruikbaarheid en delen van informatiebronnen in. De DTD of het XML Schema in een bepaalde map bewaren (lokaal of op (web)server), heeft als nadeel dat de pathaanduiding in de URI aanpassing behoeft, wanneer de DTD of het XML Schema worden verplaatst of de domeinnaam van een webserver wijzigt.

In XML-bestanden kan ook de tekstuele inhoud van databanken worden gearchiveerd. Op zich zijn XML-bestanden geen databanken, maar documenten want ze bevatten enkel lineaire tekst en geen indexen, relaties of enige andere logica voor het bewerken of bevragen van de gegevens. XML biedt wel de mogelijkheid tot gestructureerde dataopslag en de XML-familie voorziet in querytools en programmeerinterfaces. Met XML is in ieder geval de grens tussen databank en document vager geworden. Hoe de inhoud van databanken naar XML wordt omgezet, is afhankelijk van het type databank. Hiërarchische databanken bevatten op zich al een structuur en nesting die grote gelijkenissen met XML vertoont (root-parent-child). Bij export dienen XML-tags toegevoegd te worden. Het datamodel van een hiërarchische databank vormt de basis voor de DTD of het XML Schema<sup>31</sup>. Bij relationele databanken is de omzetting naar XML wat complexer. De velden van één record zijn immers over verschillende tabellen verspreid. De constructie van XML-elementen of -documenten kan op basis van SQL gebeuren. Aan de query in de hoofdtabel worden subqueries gekoppeld die in de gerelateerde tabellen worden uitgevoerd. In een volgende stap wordt de output getagd en lineair geordend (nesting) zodat er een XML-document wordt gecreëerd. De primaire sleutel kan als (uniek) ID worden gebruikt zodat alle children onder de juiste parent komen te staan. Bij object-georiënteerde databanken is er een grote overeenkomst tussen objecten en XML-elementen<sup>32</sup>.

XML is niet alleen bruikbaar voor de archivering van tekstuele documenten. Scalable Vector Graphics is een bestandsformaat gebaseerd op XML voor afbeeldingen. XML-bestanden kunnen eveneens de tekstuele bitstream van binaire bestanden bevatten (afbeeldingen, geluid, enz). Men dient wel rekening te houden met de nood

<sup>29</sup> Deze oplossing werd door DAVID gevolgd bij de XML-archivering van e-mails. Op basis van het vastgelegde documentmodel voor e-mail werd een script geschreven waardoor e-mails rechtstreeks vanuit het clientprogramma als XML-bestand worden bewaard.

<sup>30</sup> Kladblok van Windows 95 en 98 bijvoorbeeld ondersteunt Unicode niet, wat problemen met een aantal diakritische tekens kan veroorzaken. Men kan dit oplossing door in de XML-declaratie de toegepaste encoding expliciet op te geven. Bijv: <?xml version="1.0" encoding="windows-1252"?> of: <?xml version="1.0" encoding="ISO-8859-1"?>. Vanaf Windows 2000 ondersteunt Kladblok wel Unicode.

<sup>31</sup> De archivering van het kiezersregister en het bevolkingsregister van de stad Antwerpen was een toepassing van de XML-archivering van een hiërarchische databank (voor meer info: <http://www.antwerpen.be/david> onder 'cases').

<sup>32</sup> R. BOURRET, *XML and databases*, febr. 2002.

aan de passende software voor het openen van de bestanden nadat de tekstkarakters opnieuw naar binaire tekens zijn omgezet. Men kiest bijgevolg het best voor een formaat met de status van standaard en waarvan de specificatie vrij beschikbaar is. In de toekomst wordt het wellicht ook mogelijk om multimediatebestanden als pure XML-documenten te bewaren en niet enkel in een XML-wrapper te verpakken. XML is immers uitbreidbaar.

In de archiefwereld wordt XML algemeen als één van de meest aangewezen bestandsformaten voor lange termijn archivering bestempeld. In veel landen wordt XML formeel als geschikt archiveringsformaat aangeduid. XML voldoet aan de vereisten van het MTIC en is een aanbevolen standaard binnen de Franse overheid<sup>33</sup>. In Nederland wordt XML in de *Regeling geordende en toegankelijke staat archiefbescheiden* (art. 6) samen met PDF als standaard voor tekstdocumenten opgesomd. Het *Public Record Office* beveelt XML aan wanneer de migratiepiste wordt gevolgd<sup>34</sup>. Het stadsarchief Antwerpen startte in 2000 met de eerste toepassingen van XML bij de archivering van het kiezersregister en het bevolkingsregister. XML wordt er ook gebruikt bij de archivering van e-mails<sup>35</sup>. XML speelt ook een belangrijke rol in de *Persistent Object Preservation* archiveringsstrategie van het NARA en het San Diego Super Computer Center (SDSCC). Op basis van de documentaire vorm wordt een DTD opgesteld voor de verschillende types digitale archiefbescheiden. Op basis van deze DTD's worden de archiefdocumenten bij opname in het digitaal archief dan geannoteerd met XML-tags.

### 5.3 XML als metadataformaat

De metadata van digitale archiefbescheiden zijn de administratieve, technische en archivistische beschrijvingen waardoor toekomstige reconstructie en interpretatie mogelijk blijft. Ongeacht de gevolgde archiveringsstrategie, is er nood aan metadata. De metadata moeten net zoals de archiefdocumenten zelf toegankelijk blijven. Om die reden werden metadata gangbaar op papier afgedrukt of in een puur ASCII-bestand bijgehouden. XML biedt meerdere mogelijkheden om metadata te archiveren. Door semantische tags aan een document toe te voegen, wordt ten eerste in het XML-document zelf al identificerende en karakteriserende gegevens over de data bijgehouden. De metadata van digitale archiefbescheiden kunnen ten tweede altijd in een XML-document worden opgeslagen.

Door de metadata in een XML-document te bewaren, voldoet men al in grote mate aan de leesbaarheidsvoorwaarde. Bovendien is het mogelijk om de metadata op een gestructureerde wijze te bewaren. De semantische aard van XML-documenten maakt XML een geschikt formaat voor de bewaring, bevraging en uitwisseling van metadata. Een metadatatamodel kan gemakkelijk in XML worden aangelegd. Vanuit de XML-bestanden met metadata kan in principe om het even welk beschrijvingssysteem worden gevoed. XML is duidelijk een betere optie dan de bewaring van metadata op papier of in een gewoon tekstbestand.

In de bibliotheekwereld heeft men METS, Metadata Encoding and Transmission Standard<sup>36</sup>, ontwikkeld. Dit is een standaard in XML voor de opslag en uitwisseling van de metadata van samengestelde objecten (multimedia). De Nederlandse *Regeling geordende en toegankelijke staat archiefbescheiden* (art. 6) schrijft bijvoorbeeld voor dat documentatie over de databankstructuur of dat de metagegevens van een tekstdocument in TIFF of PDF in een XML-bestand wordt opgeslagen. Het Stadsarchief Antwerpen houdt de metadata van de gearhiveerde websites als XML-bestanden bij. Het archiveringsprototype dat door CEDARS werd uitgewerkt, past XML toe om de *Preservation Description Information* (PDI) als onderdeel van het *Archival Information Package* (AIP) vast te leggen. In Frankrijk wordt XML eveneens aanbevolen als meest geschikt formaat voor de digitale vastlegging van metadata<sup>37</sup>.

De *Persistent Object Preservation* methode van het NARA en het SDSCC maakt niet alleen gebruik van XML, DTD's en stylesheets, maar ook van XML Topic Maps om kennis weer te geven. De XML Topic Maps<sup>38</sup> leggen het

<sup>33</sup> MTIC, *Guide to the conservation of data and electronic documents for teleprocedures, intranets and Internet sites. Formats, media*. Parijs, 2001, p. 7.

<sup>34</sup> *Sustainable electronic records: strategies for the maintenance and preservation of electronic records and documents in the transition to 2004*, p. 23.

<sup>35</sup> Meer informatie hierover vindt u in DAVID-publicaties en op de DAVID-website (onder 'cases').

<sup>36</sup> <http://www.loc.gov/standards/mets/>

<sup>37</sup> C. DHÉRENT, *Les archives électroniques. Manuel pratique*, febr. 2002, (onder IV. 2. Constituer les métadonnées).

<sup>38</sup> <http://www.topicmaps.org/xtm/index.html>. Zie ook de DAVID-bijdrage *XML Topic Maps voor digitale archivering*.

verband vast tussen de archiefstukken en het werkproces waarbinnen ze werden ontvangen of opgemaakt. Er wordt eveneens een DTD opgesteld om de (hiërarchische) structuur van het geheel van archiefbescheiden vast te leggen. Op die manier worden de relaties tussen de archiefbestanddelen formeel en expliciet gearcheveerd. De logische relatie tussen de attributen van het geheel wordt beschreven op basis van Data Definition Language (DDL) en wordt gemapt naar de DTD van het geheel<sup>39</sup>

Een eenvoudige toepassing hiervan is de beschrijving van een mappenstructuur in een XML-bestand. Digitale documenten worden traditioneel in mappen geordend. De mappenindeling is belangrijk voor de bewaring van de archivalische band tussen de documenten onderling en met het werkproces. Aangezien mappen in courante besturingssystemen niets meer zijn dan een soort index, zou men kunnen overwegen om de structuur en samenstelling van een geheel archiefdocumenten op een meer formele en expliciet wijze in een XML-bestand te beschrijven. Een dergelijk XML-bestand beschrijft dan de 'oude orde'. De DTD voor een mappenstructuur kan relatief eenvoudig worden gehouden<sup>40</sup>:

DTD voor een mappenstructuur:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ELEMENT mappenstructuur (map | bestand)*>
<ELEMENT map (mapnaam, inhoud)>
<ELEMENT inhoud (map | bestand)*>
<ELEMENT mapnaam (#PCDATA)>
<ELEMENT bestand (#PCDATA)>
```

## 5.4 XML als inkapselingsformaat

*Encapsulation* is de archiveringsstrategie waarbij het archiefdocument samen met informatie die nodig is voor het bewaren, het interpreteren en het terugvinden in een nieuw digitaal object wordt gegroepeerd. Het nieuwe digitaal bestand functioneert als een combinatie van archiveringsformaat voor het archiefdocument en als metadataformaat. De metadata worden ingekapseld zodat deze informatie onlosmakelijk verbonden zijn met het digitale archiefstuk. Op die manier blijft de band tussen archiefdocument en metadata bewaard en beperkt men ook de afhankelijkheid van het archiefdocument aan externe bronnen. Dit laatste is niet onbelangrijk, want bij digitale archivering moet men er immers van uitgaan dat verlies van de externe bron in verlies van het archiefdocument kan resulteren. Door die bron of zijn bouwstenen mee op te nemen in het gearcheveerde document blijft die informatie beschikbaar of is het mogelijk om een externe bron opnieuw samen te stellen. Op die manier is het bijvoorbeeld mogelijk om de afhankelijkheid van de archiefbeheersapplicatie te beperken. Op een gegeven moment zal ook de archiefbeheersapplicatie te kampen hebben met technologische veroudering. Een alternatief voor het omzetten van de data uit deze applicatie is het opnieuw samenstellen van externe bronnen zoals databanken en indexen op basis van de metadata die in de digitale archiefdocumenten zelf worden bijgehouden. Bij inkapseling worden de metadata immers voornamelijk op het niveau van de gearcheveerde computerbestanden bijgehouden. Het nieuwe digitale object komt dan fysiek overeen met de logische eenheid archiefdocument en metadata.

Het principe van een inkapselingsformaat zit vervat in het *Open Archival Information System* (OAIS)<sup>41</sup>, het standaardmodel voor een archiveringssysteem voor digitale archiefdocumenten. Binnen een OAIS wordt bij opname in het digitaal depot een AIP samengesteld dat bestaat uit de componenten van het archiefdocument, informatie over de archivering en informatie nodig voor de presentatie. XML beantwoordt perfect aan de vereisten om als inkapselingsformaat te gebruiken zoals voorgeschreven binnen OAIS. Eén XML-bestand kan overeenkomen met één AIP en combineert de functies van archiveringsformaat en metadataformaat. De metadata in het XML-bestand opnemen zorgt er meteen ook voor dat deze informatie op een duurzame en

<sup>39</sup> K. THIBODEAU, R. MOORE en C. BARU, *Persistent Object Preservation: Advanced Computing Infrastructure for Digital Preservation*.

<sup>40</sup> Een uitgewerkt voorbeeld van deze toepassing is beschikbaar op de DAVID-website (<http://www.antwerpen.be/david> → cases).

<sup>41</sup> [www.ccsds.org/documents/pdf/CCSDS-650.0-R-1.pdf](http://www.ccsds.org/documents/pdf/CCSDS-650.0-R-1.pdf)

leesbare wijze wordt gearchiveerd. XML wordt hierbij als een combinatie van archiverings- en metadataformaat gebruikt.

Het soort metadata die men in het XML-bestand opneemt kan heel divers zijn: kontekstuele informatie (archiefnummer, referentie naar de zaak of het dossier, gerelateerde archiefdocumenten), compressiealgoritme, hashcode, trefwoorden, verwijzingen naar documentatie, specificatie bestandsformaat, indexgegevens. Zelfs de bitstream van de viewer die nodig is voor het bekijken van het archiefdocument kan mee in het XML-bestand worden bewaard. Op basis van die ingekapselde metadata is het bijvoorbeeld mogelijk om achteraf een viewer of externe index opnieuw op te bouwen, wanneer die verloren zouden zijn. Voor dezelfde metadata voor een grote hoeveelheid archiefdocumenten ga je op voorhand wel best na of je de metadata zelf of een enkel verwijzing naar de gemeenschappelijk gedeelde metadata (bijv. specificatie van een bestandsformaat) mee opneemt.

Inkapseling van digitale documenten in XML-bestanden is toepasbaar op elk type computerbestand. Het verpakken van digitale archiefbescheiden in een XML-bestand beperkt dus het aantal verschillende soorten items die binnen het digitaal archief worden bijgehouden. Door toepassing van inkapseling in XML-bestanden is het digitaal archief vrij homogeen samengesteld, wat het archiefbeheer vergemakkelijkt.

Inkapseling is voor digitale archiveringsdoeleinden op meerdere wijzen toepasbaar. Aan de bytestream van het archiefdocument zelf kunnen de nodige metadata worden toegevoegd. Inkapseling kan ook toegepast worden om meerdere versies van hetzelfde archiefdocument als één digitaal bestand te bewaren. Hetzelfde XML-bestand kan zowel de tekstuele (XML) als de rastervoorstelling (TIFF) van het archiefdocument bevatten.

XML wordt onder meer door VERS (*Victorian Electronic Records Strategy*<sup>42</sup>) voor de inkapseling van archiefbescheiden gebruikt. Het XML-bestand bevat ondermeer het archiefobject zelf, de digitale handtekening en de metadata. Het archiefobject wordt binnen VERS in het PDF-formaat gearchiveerd. Het Nederlandse *E-Archive* project gebruikt XML-bestanden als containers waarin de metadata, de originele bitstream, de viewerlijst (bestandsnamen van de AIP's met de broncode van de passende viewers), en mogelijk alternatieve representaties voor bibliografische items. De AIP's voor de viewers zijn eveneens XML-bestanden<sup>43</sup>. Ook de XML-toepassing van de *Persistent Object Preservation* methode kan als inkapseling worden beschouwd.

## 6. Besluit

XML is niet de ultieme oplossing voor alle moeilijkheden die bij digitale archivering om de hoek komen kijken, maar kan de archivaris wel een flink stuk op weg helpen. Binnen de archiefwereld wordt XML geleidelijk aan meer en meer gebruikt als metadata- en archiveringsformaat. Hiermee lost men al grotendeels het probleem van de technologische veroudering op en geniet men van de voordelen van gestructureerde en semantische opslag van digitale archiefdocumenten. Op dit ogenblik is er geen enkele technologie die betere waarborgen voor lange termijn leesbaarheid biedt. Bovendien zijn XML-documenten gemakkelijk om te zetten naar nieuwe technologieën. Documentmodellering wordt nog niet op zo'n brede schaal toegepast. Door XML toe te passen als inkapselingsformaat beperkt men de afhankelijkheid van externe bronnen en kan men de autonomie van de digitale archiefdocumenten nog vergroten.

## 7. Bronnen

<http://www.w3C.org>

<http://www.xml.com>

<http://xml.coverpages.org>

<http://www.covax.org/>

---

<sup>42</sup> <http://www.prov.vic.gov.au/vers/welcome.htm>

<sup>43</sup> <http://www.library.tudelft.nl/e-archive/>

<http://www.topicmaps.org/xtm/index.html>

<http://www.w3schools.com>

R. BOURRET, *XML and databases*, febr. 2002. (<http://www.rpbourret.com/xml/XMLAndDatabases.htm> )

J. CLARK, *Comparison of SGML and XML*, 1997 (<http://www.w3.org/TR/NOTE-sgml-xml.html>).

S. GILHEANY, *XML for records managers*, (<http://www.archivebuilders.com>)

D. MARTIN e.a., *Professional XML*, Birmingham 2000.

R.W. MOORE, *The preservation of data, information and knowledge*, 2001.

T. SHINKLE en R. MARCIANO, *Using "XML" Standards for the preservation of records*, lezing gegeven tijdens de ARMA 2001 Annual Meeting, 3 oktober 2001 te Montreal.

K. THIBODEAU, R. MOORE en C. BARU, *Persistent Object Preservation: Advanced Computing Infrastructure for Digital Preservation*, in: *Proceedings of the DLM-Forum on electronic records. European citizens and electronic information: the memory of the Information Society*, Brussel, 18-19 oktober 1999, p. 113-118.

E. VAN DER VLIST, *XML for document preservation*, 2001. (<http://www.xmlhack.com/read.php?item=1030&r=1>)

K. WILLIAMSON, *XML: The complete reference*, Berkeley, 2001.