XML- ein Überblick über das Chaos

XML gewinnt als Basis für Dokumentensprachen für spezielle Anwendungen immer mehr an Bedeutung. So gibt es bereits ausgefeilte Definitionen und Programme, um mathematische Formeln, chemische Strukturen und Verbindungen, aber auch multimediale Dokumentbestandteile in XML zu beschreiben und zu bearbeiten. In diesem Artikel werden kurz folgende XML-Sprachen und Anwendungen vorgestellt: MathML: Mathematical Markup Language, CML: Chemical Markup Language, SVG: Scalable Vector Graphics, SMIL: Synchronized Multimedia Integration Language und X3D: Extensible 3D Language.

Einleitung

Nachdem in Heft 18 der RZ-Mitteilungen eine Einführung in die grundlegenden Begrifflichkeiten und Funktionsweisen der eXtensible Markup Language erfolgt ist, sollen in diesem Beitrag nun überblicksartig einige typische XML-Anwendungen vorgestellt werden. Dabei erhebt die Zusammenstellung keinen Anspruch auf Vollständigkeit. Ein solcher wäre auch aufgrund der rasanten Entwicklung und Verbreitung, der XML als Konzept und Dokumentenbeschreibungssprache unterliegt, kaum realistisch.

Da XML nur das Grundgerüst bietet, um jede Art von Dokumenten in ihrer Struktur zu beschreiben, zu erfassen und eine Einheitlichkeit zur Verwaltung gleichartiger Dokumente zu schaffen, haben sich eine Menge sogenannter XML-Dialekte herausgebildet, die der Verarbeitung spezieller Dokumenttypen dienen. So wurde z. B. innerhalb des Projektes „Digitale Dissertationen“ am Rechenzentrum der Humboldt-Universität eine Dokumenttypdefinition für Dissertationen *DiML* [1] entwickelt, die die Archivierung und Recherche der elektronischen Dissertationen der Humboldt-Universität auf XML-Basis unterstützen soll. Dieses ist jedoch nur ein Beispiel für die Definition fachspezifischer Dokumenttypen. Viele der Definitionen werden vom World Wide Web Konsortium (W3C) [2] standardisiert und entwickelt. So zum Beispiel:

- die Mathematical Markup Language (*MathML*) [3],
- die Definition der Scalable Vector Graphics (*SVG*) [4],
- die Synchronized Multimedia Integration Language (*SMIL*) [5],
- eXtensible 3D (*X3D*) [18]
- das Resource Description Framework (*RDF*) [6].

Andere Standards werden durch die weltweite Zusammenarbeit in Organisationen erarbeitet, so zum Beispiel

- das Dublin Core Set (*DC*) [7] zur Beschreibung von WWW-Inhalten,
- die Wireless Markup Language (*WML*) [8] als neue Möglichkeit, Handys mit Internetapplikationen auszustatten,
- die Chemical Markup Language (*CML*) [9],
- oder auch spezielle Anwendungen wie die Entwicklung einer Commercial Markup Language (*cXML*), die innerhalb der BiZTalk-Initiative [10] entwickelt wird, um EDI-basierte Systeme auf XML umzuarbeiten und die dem elektronischen Geschäftsverkehr dienen soll.

Spezielle Entwicklungen wie die Music Markup Language (*MusicML*) [11] werden zum Teil auch von kleinen Gruppen oder Firmen vorgenommen.

Um sich einen Überblick über die verschiedenartigen Entwicklungen zu verschaffen, eignet sich das seit rund 2 1/2 Jahren gepflegte Forum *Schema.Net* [20]. Hier werden XML-Dokumenttypdefinitionen, XML-Schemas, Stylesheets, XML-Informationen und XML-Software gesammelt und zu einem wohlgeordneten Archiv zusammengestellt. Für XML-DTDs bzw. Schemata stehen z. B. folgende Kategorien zur Verfügung:

- Allgemein,
- E-Commerce, Geschäftsinformationen und Finanzbereich,
- Bildung
- Sprache- und Wissensrepräsentation,
- Metadaten und Archivierung,
- Multimedia und Graphik
- Technik und Wissenschaft
- Software
- Netzwerke, Internet, etc.

Was sind Dokumenttypdefinitionen?[2]

Eine Dokumenttypdefinition im Sinne von XML definiert Regeln oder Vorlagen, nach denen gleichartige Dokumente zu gestalten sind. Eine DTD beschreibt dabei das Inhaltsmodell einer Klasse von Dokumenten. Sie besteht aus folgenden Komponenten:

- einer *Notationsdeklaration*, die Notationen für Fremdformate, z. B. für Graphik (gif, jpeg), Mathematik (Tex, LaTeX), 3D-Objekte (VRML) und ähnlichem definiert, für die keine eigene XML-Dokumenttypdefinition existiert,
- einer *Entitydeklaration*, in der Zeichenersetzungen von Zeichen definiert werden, die nicht standardmäßig im benutzten Zeichensatz (z. B. ASCII) enthalten sind, z. B. wird Ü als Ü definiert.
- einer *Elementtypdeklaration*, in der die Hierarchie der Elementtypen[1] aufgebaut wird, wobei es immer ein

1 Die in spitzen Klammern geschriebenen Elemente z. B. <autor> heißen in der DTD Elementtypen.

2 Der folgende Abschnitt wurde einem Artikel aus dem Bibliotheksdienst 8/9 2000 entnommen.

eindeutiges oberstes „root“-Element geben muss, unter das die anderen Elementtypen untergeordnet werden. Elemente aus externen DTDs werden einfach in die Hierarchie unter Angabe der DTD und ggf. des zugehörigen Namespaces eingebunden. Namespaces sind ein Instrument, um gleichlautende Namen von Elementtypen aus verschiedenen DTDs eineindeutig zuzuordnen.
- und einer *Attributlistendeklaration*, in der die den Elementtypen zugehörigen Attribute und deren zulässige Wertebereiche definiert werden.

Zur Formulierung einer DTD wird eine spezielle Syntax (nicht XML) benutzt.

Das Publizieren von Dokumenten kann in XML auf zweierlei Art geschehen:
- durch sogenannte „wohlgeformte“ Dokumente, für die keine DTD benötigt wird, die jedoch streng hierarchisch mit einem root -Element aufgebaut und vollständig ausgezeichnet (mit sogenannten „Tags“ versehen, also getaggt) sind.
- außerdem kann eine Publikation mit sogenannten „validen“ oder „gültigen“ XML-Dokumenten erfolgen. Diese Dokumente benötigen immer eine DTD.

Zur Zeit befindet sich innerhalb des W3C eine Alternative zur Verwendung von DTDs in der Diskussion. XML-Schemas definieren ähnlich wie DTDs Grundstrukturen und Inhaltsmodelle für eine Klasse von Dokumenten. Ähnlich wie bei einer Speicherung in Datenbanken, bieten Schemas zusätzlich die Möglichkeit, vordefinierte Datentypen aus der Programmiersprachen (JAVA)- und Datenbankwelt zu benutzen, z. B. Real, String, Byte und auf vordefinierte Wertebereiche und deren Grenzen, z. B. minLength, MaxExclusive, scale, encoding, etc. zuzugreifen. Ein Vorteil der XML-Schemas ist der, dass zur Definition von Dokumentklassen im Gegensatz zu DTDs XML als Syntax benutzt wird.

MathML - Mathematical Markup Language

MathML wurde entwickelt, um mathematische und andere wissenschaftliche Inhalte über das WWW für andere Nutzer verfügbar zu machen. So zum Beispiel für Anwendungen, wie Computer Algebra Systeme (z. B. Maple, Mathematika), Textsatzsysteme (z. B. LaTeX) oder um Sprachausgaben zu erzeugen. MathML eignet sich somit nicht nur zur Formulierung mathematischer Inhalte für Visualisierungen, sondern auch zur Weiterverarbeitung aktiver Komponenten. Da MathML als XML-Applikation von den derzeit gängigen Browsern nicht direkt visualisiert werden kann, benötigt man entweder umfangreiche Stylesheets (XSL) oder eigene Browser, um die mathematischen Formeln darzustellen. Den Grund für die Definition einer eigenen mathematischen Sprache für das WWW bildete vor allem die Unzulänglichkeit von HTML bei der Auszeichnung und Visualisierung vom mathematischen Formeln. Die einzige Möglichkeit zur Visualisierung im WWW stellten bisher Rastergrafiken oder PDF-Dateien dar, die einen strukturierten Datenaustausch nicht zulassen. In der W3C Arbeitsgruppe MathML arbeiten Vertreter von Firmen wie IBM, Microsoft, Adobe, MacKichan, Stilo, NAG, Waterloo Maple Inc. Boeing und andere mit.

Am 28.03. 2000 wurde der MathML 2.0 Working Draft verabschiedet. Das bedeutet, MathML2.0 befindet sich in einer letzten Bearbeitung und wird danach als ein Empfehlung (Recommendation) des W3C herausgegeben. Dies entspricht im Prinzip der Definition eines Standards.

MathML ist eine Dokumenttypdefinition im Sinne von XML. Sie definiert eine Anzahl von XML-Tags, die für einerseits dem Auszeichnen von Elementen für eine Präsentation dienen, andererseits auch zur Auszeichnung von semantischen[3], d. h. bedeutungstragenden, Elementen genutzt werden können.

Ein kleines Beispiel für die Darstellung einer Mathematisches Formeln ist folgendes:

$$x^2 + 4x + 4 = 0$$

Die Repräsentation in MathML:

```
<mrow>
  <mrow>
    <msup> <mi>x</mi> <mn>2</mn> </msup> <mo>+</mo>
      <mrow>
        <mn>4</mn>
        <mo>&invisibletimes;</mo>
        <mi>x</mi>
      </mrow>
    <mo>+</mo>
    <mn>4</mn>
  </mrow>
    <mo>=</mo>
    <mn>0</mn>
</mrow>
```

Abb. 1 Darstellung einer mathematischen Formel in MathML (Visualisiert und im Quellformat)

Grundsätzlich gilt für die Tags: Sie beginnen mit „m“ und besitzen dann folgenden logischen Aufbau: ist der nachfolgende Buchstabe im Tag ein „o“ dann bezeichnet dieses Tag einen Operator, ein „i“ bezeichnet einen Identifier, ein „n“ eine Nummer, etc. Das Tag <mrow> z. B. organisiert die Ausdrücke in horizontale Gruppen.

Es gibt bereits eine Menge von Softwareprodukten, die MathML exportieren, visualisieren oder verarbeiten können. Dazu gehört zum Beispiel der Amaya-Browser des W3C, der gleichzeitig als Autorensystem für MathML benutzt werden kann; E-Lite, ein Java-basierter Browser der Firma ICESoft’s, der IBM

3 Semantisch orientierte Tags dienen zur Darstellung von Konzepten wie Multiplikation <times>, Potenzierung <power>, etc.

```
<apply>
  <plus/>
  <apply>
    <power/>
    <ci>x</ci>
    <cn>2</cn>
  </apply>
  <apply>
    <times/>
    <cn>4</cn>
    <ci>x</ci>
  </apply>
  <cn>4</cn>
</apply>
```

Abb. 2: Das Konzept semantischer Tags in MathML: <times> und <power>

Techexplorer, ein Programm zur Visualisierung von wissenschaftlichen Dokumenten, der ein plug-in für MathML und TeX/LaTeX besitzt. Das Computeralgebrasystem Maple ist in der Lage, MathML-Dokumente zu importieren, zu exportieren und darzustellen. Auch Mathematica ist in der Lage, MathML zu exportieren und darzustellen. Im LaTeX2HTML-Programm findet sich in der neuesten Version ein MathML Paket, welches eine Konvertierung von LaTeX nach MathML vornehmen kann. Es existieren weitere, die hier jedoch nicht genannt werden, stattdessen sei auf die MathML-Seiten des W3C verwiesen.

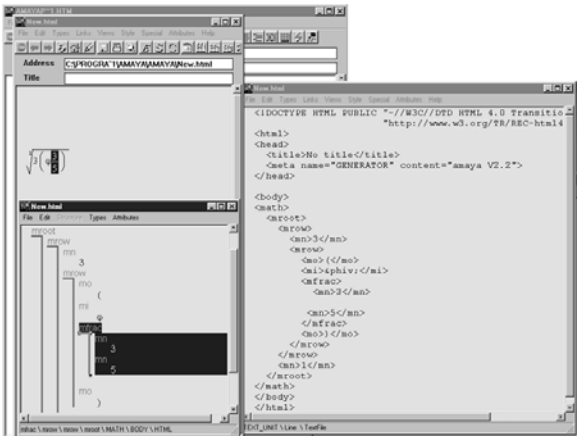

Abb. 3: Erstellung einer mathematischen Formel im Amaya-Browser

CML - Chemical Markup Language

Die Chemical Markup Language wurde mit ähnlichen Intensionen entwickelt wie MathML. Bereits 1995 begann Peter Murray-Rust damit, die erste DTD zu entwickeln. Chemische Informationen, die auf herkömmliche Weise im WWW präsentiert werden, verlieren viel von ihrem Informationsgehalt und der Möglichkeit, weiter verarbeitet zu werden. Das Einfügen von chemischen Molekülen, Spektren, Reaktionsgleichungen als gif- oder jpeg-Bilder in HTML-Dokumenten bringt für den Fachwissenschaftler wenig Informationen. Wichtiger ist die strukturierte Darstellung und Aufbereitung der Daten, um aus einer einzigen Beschreibung verschiedene Visualisierungsformen zu generieren. So findet man in der Chemie häufig Dateiformate, die da heißen: xyz-String, SMILS-String, cbin, cascii, ctx, molfile(+SD), smd5, jcamp, cif(Chiron), pdb, compass, 441, rxn,cbase, sharc, smd4, alchemy oder gaussian (input, archive, cube). Um die Struktur dieser Daten zu erhalten, wurde CML als Daten- und Dokumentformat entwickelt, um Dokumente mit chemischen Inhalten über das WWW zur Verfügung zu stellen. Die Version 1.0 der Markupsprache von Peter Murray-Rust und Henry Rzepa wurde 1999 formal veröffentlicht[4] und dient der Auszeichnung von 2- und 3-dimensionalen Molekülstrukturen. Dabei nutzt es zur Darstellung Standards, die im Umfeld des Faches Chemie existieren, wie Verbindungstabellen, SMILS-Strings, 2-D Koordinaten oder 3-D Koordinaten (Kartesisches Koordinatensystem), fraktale Koordinaten, Z-Matrizen, Kristallographische Zelleinheiten und andere.

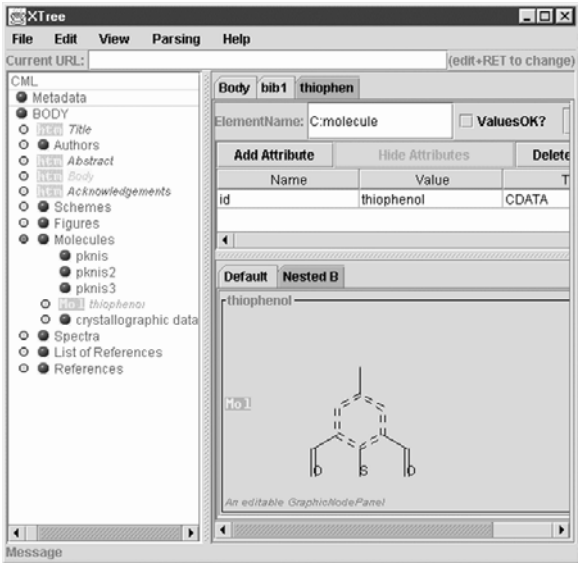

Abb. 4: Das Molekül Thiophenol im JUMBO-Browser/ Editor und im CML- Sourcecode, aus [16]

Defizite existieren noch bei der Beschreibung von Reaktionsgleichungen, kristallographischen und spektroskopischen Daten. Für die Auzeichnung makromolekularer Objekte wird von den Chemikern z. B. die BioML.DTD verwendet. Um jedoch die hohe Akzeptanz in der relativ kleinen Gemeinde der Markup-Chemiker nicht zu gefährden, wird CML einem kontrollierten Versionierungsprozess unterzogen. Viele freie Werkzeuge unterstützen CML bereits als Import- und Exportformat, kommerzielle Anbieter halten sich jedoch bisher bei einer Implementation zurück.

4 Journal Chem. Inf. Comput. Sci. 39 (1999) 928.

```
<!-- thiophenol for 02788; arrays are formatted SOLELY for readability -->
<?jumbo:namespace ns="http://www.xml-cml.org" prefix="C"
          java="jumbo.cmlxml.*Node" ?>
<C:molecule id="thiophenol">

  <C:atomArray builtin="elsym">
C C    C   C  C    C     C  S  C    C     O    O  </C:atomArray>

  <C:atomArray builtin="x2" type="float">0 0.866 0.866 0    -0.866 -0.866
          0.0  0.0  1.732 -1.732 1.732 -1.732  </C:atomArray>

  <C:atomArray builtin="y2" type="float">1 0.5   -0.5  -1.0 -0.5   0.5    -2.0
          2.0  1.0   1.0    2.0   2.0  </C:atomArray>

  <C:bondArray builtin="atid1"> 1 2 3 4 5 6 1 4 2 9  6  10
</C:bondArray>

  <C:bondArray builtin="atid2"> 2 3 4 5 6 1 8 7 9 11 10 12
</C:bondArray>

  <C:bondArray builtin="order" type="integer">4 4 4 4 4 4 1 1 1 2  1  2
</C:bondArray>

</C:molecule>
```

Abb. 5: Das Molekül Thiophenol im CML- Sourcecode, aus [16]

Die Darstellung hängt dabei, genau wie bei den anderen XML-Applikationen, immer davon ab, wie gut die Software ist, die CML interpretieren und visualisieren kann. Es existieren im Moment vor allem 2 CML-Browser/Editoren: JUMBO [16] von Peter Murray-Rust und Jmol von der Openscience Foundation [17].

SVG - Scalable Vector Graphics

Die Definition „Scalable Vector Graphics“ zielt darauf ab, zweidimensionale vektororientierte Grafiken über das Word Wide Web anzubieten. Dabei wird vor allem Wert auf die Möglichkeit gelegt, das Vektorgrafiken im Gegensatz zu Rastergrafiken, vor allem das Zoomen, das Vergrößern und Verkleinern von Bildern und Bildausschnitten ohne Qualitätsverlust ermöglichen, vgl. [12]. SVG besitzt im Moment den Status eines W3C Working Drafts, der sich in der letzten Überarbeitungsrunde befindet. D.h. es handelt sich hierbei noch nicht um eine fertig ausgearbeitete Standardisierung, sondern um ein Arbeitspapier, bis zu dessen endgültiger Fertigstellung noch geringfügige Änderungen zu erwarten sind. Da Firmen wie Microsoft, Adobe, Apple, Corel, IBM, Sun, Macromedia und Netscape in der W3C Arbeitsgruppe mitarbeiten, ist in Zukunft damit zu rechnen, daß sich SVG als WWW-fähiges Grafikformat durchsetzt. Es gibt heute schon eine Reihe kleiner Tools, die SVG unterstützen. So bietet Corel zum Beispiel einen Exportfilter für Corel Draw 9 [13] an, Adobe einen Exportfilter für Illustrator 9 und ein Browser-Plugin, der als Zusatz zu gängigen Webbrowsern installiert, SVG-Grafiken über das WWW visualisieren kann. IBM bietet auf seiner Alphaworks-Seite einen auf Java basierenden SVG-Viewer [14] an. Die Firma CSIRO Mathematical and Information Sciences bietet ebenfalls einen Java-basierten SVG-Viewer [15]

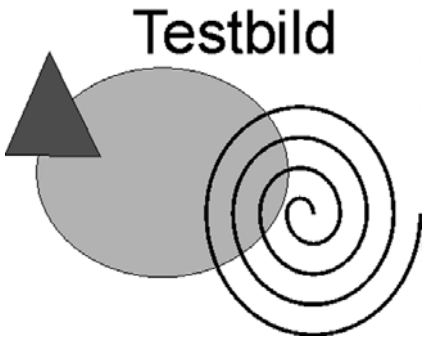

Abb. 6: Testbild im cdr-Format

an. Innerhalb des Mozilla-Projektes arbeitet eine Arbeitsgruppe daran, den Netscape/Mozilla-Browser mit SVG-Funktionalitäten auszustatten. Die Firma Digital Applications, Inc. hat bereits ein Konvertierungsprogramm von SVG nach PDF entwickelt.
Viele Grafiken, die zur Präsentation im WWW entworfen werden, werden mit vektororientierten Zeichenprogrammen, wie Corel Draw oder Adobe Illustrator erstellt. Der Vorteil ist, dass die hier erstellten Elemente und Formen ohne Qualitätsverlust beliebig vergrößert, verkleinert, gestreckt oder gestaucht werden können. Die so erstellten Grafiken werden dann oft, um

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg SYSTEM "svg-19991203.dtd">
<!-- Creator: CorelDRAW -->
<svg    xml:space="preserve"    width="3.93042in"
height="3.08358in"  style="shape-rendering:geome-
tricPrecision;  text-rendering:geometricPrecision;
image-rendering:optimizeQuality"  viewBox="-1898  0
3931 3084">
 <defs>
  <style type="text/css">
   <![CDATA[
    .str0 {stroke:#1F1A17;stroke-width:3}
    .str1 {stroke:#151619;stroke-width:28}
    .fil1 {fill:none}
    .fil2 {fill:#1F1A17}
    .fil3 {fill:#605E5D}
    .fil0 {fill:#AAA9A9}
   ]]>
  </style>
 </defs>
 <g >
  <title>Ebene 1</title>
  <ellipse  class="fil0  str0"  cx="-413"  cy="1544"
  rx="1187" ry="985"/>
  <path class="fil1 str1" d="M1009 1928c0,-79 -56,-
  143 -126,-143 -70,0 -126,64 -126,143 0,157 113,285
  252,285 140,0 253,-128 253,-285 0,-237 -170,-429
  -379,-429 -209,0 -378,192 -378,429 0,315 226,571
  504,571 279,0 505,-256 505,-571 0,-395 -282,-714
  -631,-714 -348,0 -631,319 -631,714 0,473 339,856
  757,856 418,0 757,-383 757,-856 0,-552 -395,-1000
  -883,-1000  -488,0  -883,448  -883,1000  0,631
  452,1142 1009,1142 558,0 1010,-511 1010,-1142"/>
  <text x="-714" y="412" class="fil2"
  style="font:normal 575 Arial">Testbild</text>
  <path class="fil3 str0" d="M-1479 411l-417 966 900
  19 -483 -985z"/>
 </g>
</svg>
```

Abb 7: Testbild als svg-Datei im Quelltext

sie WWW-fähig zu machen, als Rastergrafiken exportiert und verlieren so ihre Flexibilität und Qualität. Das Exportformat SVG ist ein textbasiertes Grafikformat, in welchem die Elemente durch ausgewählte tags und attributes beschrieben werden. Es ist möglich, Animationen durch bestimmte beschreibende Elemente oder durch das Einbinden von Scripten, ablaufen zu lassen. Dabei hat der Nutzer folgende Möglichkeiten:

- svg x, y: bildet das sogenannte root-Element von SVG-Dateien. Die x,y-Koordinaten dienen der Positionierung innerhalb einer WWW-Seite.

Weiterhin besteht die Möglichkeit, verschiedene grafische Elemente zu benutzen:

- Rechteckige Fläche mittels *rect* (*x,y, width, height*)
- Kreisflächen mit *circle* (*cx, cy, r*). Dabei geben cx, cy die Koordinaten für den Mittelpunkt und r den Radius an.
- Ellipsen: *ellipse* (cx, cy, rx,ry)
- Linien: *line x1,x2,y1,y2*
- Polylinien, d.h. Linien mit mehreren Eckpunkten: polyline (x1,y1,x2,y2,xn,yn)
- Polygone: polygon (*x1,y1,x2,y2,xn,yn*)
- Textteile werden mittels *text* (*String,x,y*) ausgezeichnet.

Es besteht ebenso die Möglichkeit, Bildateien, allerdings nur im Jpeg-Format einzubinden: *image* (*x, y, width, height*)
Vergleicht man das SVG-Grafikformat mit anderen, so stellt man beim Test fest, daß SVG-Bilder weniger Speicherplatz belegen. Das Testbild besaß in den verschiedenen Grafikformaten unterschiedliche Dateigrößen:

- CDR-Bild: 14 KBytes
- BMP-Bild: 111 KBytes
- GIF-Bild: 5 KBytes (komprimiert)
- SVG-Bild: 2 KBytes.

Eine SVG-Grafik ist folgendermaßen aufgebaut: zuerst enthält es, wie jedes XML-Dokument einen Header:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg SYSTEM „svg-19991203.dtd">
<!— Creator: CorelDRAW —>
```

Dieser gibt die XML-Version und die Dokumenttypdefinition SVG an, zu der das Dokument valide (zu einer DTD konform) sein soll.

Dann folgt ein Element <svg> als oberstes Element. Darin sind die Grundelemente enthalten. Möglich sind z. B. <circle/> für die Darstellung eines Kreises, <ellipse/> für Ellipsen, <text/> für Textelemente, <rect/> für Rechtecke, <line/> für Linien, <polygon /> für Polygone. Die eigentlichen Inhalte bzw. Größen und Ausprägungen der Elemente werden durch die Attribute bestimmt: so wird die Ellipse aus dem Beispiel beschrieben durch die Attribute:

- class für die Angabe von Klassenzugehörigkeiten eines Elements, um z. B. auf Stylesheet-Angaben zugreifen zu können,
- cx für die x-Koordinate des Mittelpunktes des Ellipse,
- cy für die y-Koordinate des Mittelpunktes des Ellipse,
- rx für den Radius der Ellipse auf der x-Achse,
- ry für den Radius der Ellipse auf der y-Achse

```
<ellipse class="fil0 str0" cx="-413"
cy="1544" rx="1187" ry="985"/>
```

SMIL - Synchronized Multimedia Integration Language

Die Synchronized Multimedia Integration Language dient der Synchronisation von multimedialen Präsentationen im WWW, ähnlich Quicktime. SMIL bietet sich zum Beispiel zur Erstellung TV-ähnlicher Medienpräsentationen für interaktive Kurse oder Powerpoint-artige Präsentationen an.

SMIL wird durch eine Menge von XML-Dokumenttypdefinitionen, die relativ leicht zu erlernen sind, beschrieben. Viele Elemente gleichen denen der XHTML-DTD. Im speziellen sind das die SMILE Animation-DTD, SMIL Layout-DTD, SMIL Linking-DTD, SMILE Media Object-DTD, SMILE Streaming Media Object-DTD, SMILE Basic Profile-DTD und ein XML Schema für Metadaten.

Eine SMIL Präsentation kann unterschiedliche Komponenten beinhalten: Streaming Audio und Video, Bilddaten, Text sowie weitere Medientypen. SMIL erlaubt die Positionierung von Medienelementen an jeder beliebigen Stelle auf dem Bildschirm und deren Synchronisation. Oft wird SMIL auch eingesetzt, um Präsentationen entsprechend den Benutzerpräferenzen wie Sprache, Bit-Rate, etc.abzuspielen. Je nach Wunsch des Benutzers kann dann die entsprechende Audiospur benutzt werden oder alternativ Untertitel eingeblendet werden. Einige gute Beispiele der Anwendung findet man unter [21]. Zu empfehlen ist das Tutorial unter [22]. Im November 1997 wurde der SMIL-Standard das erste Mal veröffentlicht, der letzte Working Draft für SMIL Boston datiert vom 22. Juni 2000. Seitdem unterstützen viele Systeme das Format: mit Apple QuickTime 4.1, Compaq HPAS, Helios SOJA, Microsoft Player Internet Explorer 5.5, NIST S2M2 Player, Oratrix Grins 1.0, Productivity Works LP-player, RealNetworks Realplayer 7 oder 8beta kann man bereits SMIL-Präsentationen abspielen. Allerdings sind diese untereinander nicht immer kompatibel. So bindet man Text für den Realplayer als Realtext .rt ein, für den SOJA-Player oder Quicktime müssen es .txt-Dateien sein. Der GRiNS-Player versteht beides. Die Firmen bieten auch eine Reihe von SMIL Autorenwerkzeugen an: Allaire von HomeSite, CWI SMIL Validator, HotSausage SMIL Composer SuperTool, LP Studio, Oratrix Grins, RealSlideshow 2.0 by RealNetworks, TAG Editor 2.0 - G2 release by Digital Renaissance , VEON authoring tool, u.s.w.

Das Einbinden einer SMIL-Präsentation erfolgt durch den Aufruf einer .smi-Datei:
Die eigentliche SMIL-Datei beginnt dann mit <smile> und endet mit </smile>. Dabei ist es aufgrund der Tat-

```
<html>
<head>
  <title>Test des Einbindens einer SMIL-
         Präsentation</title>
  <link rel="stylesheet" type="text/css"
         href="style.css">
</head>
<body>
<h1>Test des Einbindens einer SMIL-
         Präsentation</h1>
<h3> Autor: Susanne Dobratz </h3>
<center>
<a href="test.smi">This is a SMIL presentation</a>
</center>
<br>
<h3> OderAufruf eines Applets zum Abspielen mit
         dem Soja-Applet </h3>
<a href="soja.html">Zur Soja-Demo </a>
</body>
</html>
Möchte man das SOJA-Applet verwenden, muß man die-
se in der HTML-Seite aufrufen:
<APPLET CODE="org.helio.soja.SojaApplet.class"
        ARCHIVE="soja.jar" CODEBASE="./"
        WIDTH="400" HEIGHT="300">
  <PARAM NAME="source" VALUE="test.smil">
  <PARAM NAME="bgcolor" VALUE="#000066">
</APPLET>
```

Abb. 8: Einbinden einer SMIL-Präsentation in HTML

sache, dass SMIL case-sensitiv ist wichtig, dass alle Tagnamen klein geschrieben werden. Wie in HTML gibt es auch wieder einen Bereich <head>, in dem Metadaten erlaubt sind <meta name="autor" content="Susanne Dobratz"> und einen <body>-Bereich.

Im <head>-Bereich wird dann auch das Layout <layout> ... </layout> festgelegt. So gibt es generell ein oberstes Layout-Element, welches die Größe des Anzeigefensters festlegt, z. B.:

<root-layout width="300" height="200" background-color="white" />

Danach kann man verschiedene Bereiche <region> definieren, in denen die unterschiedlichen Präsentationselemente ablaufen. Dabei nimmt man eine absolute Positionierung der Layouts vor, die relativ zur linken oberen Ecke des Root-Bereiches ist. Die eigentlichen Elemente werden im <body> aufgerufen. Dabei ist es wichtig, jedem der Elemente auch einen <region>-Bereich zuzuordnen, in dem sie visualisiert werden. Eine Synchronisation kann über eine <switch>-Anweisung erfolgen. Die Synchronization der einzelnen Präsentationselemente erfolgt über die Tags: <par> für paralleles Abspielen, <seq> um das Abspielen zeitlich nacheinander anzuordnen und <switch> um entsprechend den Voreinstellungen des Benutzers und des Players eine Auswahl für bestimmte Präferenzen, wie Sprache oder Bildschirmauflösung zu ermöglichen.

```
<smil>
 <head>
  <layout>
<root-layout height="200" width="300" background-
     color="blue" title="Test SMIL Document"/>
   <region id="Kopfzeile" left="0" top="0"
          width="100%" height="15" background-
          color="navy"/>
   <region id="Fusszeile" left="0" top="175"
          width="100%" height="15" background-
          color="green"/>
   <region       id="Body" left="20" top="30"
          width="100%" height="80%" background-
          color="blue" />
    </layout>
 </head>
 <body>
<seq> <!--führt die nachfolgenden Aktionen nach-
     einander aus -->
   <par>  <!--führt die nachfolgenden Aktionen
        parallel aus -->
      <img src="images/Logo_en.gif" region="Body"
          dur="3s"/>
      <text src="texte/kopf.txt"
          region="Kopfzeile" dur="2s" begin="1s"/>
      <text src="texte/fuss.txt"
          region="Fusszeile" dur="4s" begin="9s"/>
      <text src="texte/mitte.txt" region="Body"
          dur="6s" begin="3s"/>
      <img src="images/Logo_en.gif" width="15"
          region="Body" dur="3s" begin="15s"/>
   </par>
      <video src="videos/didi.mov" dur="45s"/>
   <par>
      <a id="a1"  show="new"
        href="http://dissertationen.hu-berlin.de">
        <img src="images/hub.gif" left="45%"
            top="45%" region="Body"/>
        <text src="texte/text4.txt" left="45%"
            top="90%" region="Fusszeile" />
      </a>
   </par>
</seq>
 </body>
</smil>
```

Abb. 9: SMIL-Beispiel (auch live unter http://amor.rz.hu-berlin.de/~h0077dfz/smile/test.html)

Abb. 10: Ausschnitt aus dem SMIL-Beispiel dargestellt mit Quicktime 4.1

Leider müssen SMIL-Präsentationen an den jeweiligen Player angepasst werden, da die einzelnen sehr unterschiedlich auf genau dieselben Befehle reagieren.

X3D

Extensible 3D ist eine Neuentwicklung im Bereich 3D-Visualisierung und wird als der Nachfolger des VRML97-Standards angesehen. Zielbereiche sind z. B. Branchen, wie die Werbung oder der E-commerce-Bereich, in denen es für Produkt- und Datenbankenvisualisierungen oder zur Anreicherung von Dokumenten mit 3D-Objekten genutzen werden kann. Es ist sogar geplant, X3D 2002 durc die ISO zertifizieren zu lassen. Im Gegensatz zum VRML-Standard folgt X3D einem komponentenbasierten Ansatz, der den Einsatz scalierbarer Applikationen und kleiner Anwendungsprogramme, sogenannter thin clients, ermöglichen soll. Die Idee dahinter ist die, dass ein X3D-Betrachter als kleines, scalierbares Java-Applet von z. B. 60 KByte Größe laufen kann. Weitere Funktionen und Profile, die das Minimalmodell erweitern, können dann bei Bedarf dazugeladen werden. Ein weiteres Entwurfsziel ist die Standardisierung der Kommunikation mit anderen Komponenten und Programmen in der 3D-Welt durch die Nutzung des DOM (Document Object Model). X3D unterstützt dabei sowohl ein textbasiertes, d.h. „getaggtes“ Format als auch ein komprimiertes Format. Letzteres soll der Einsparung von Ladezeit im Internet dienen. Die Entwicklung, die innerhalb eines Industriekonsortiums (nicht W3C) geschieht, kann unter [23] beobachtet werden. Firmen, wie Sun, Shout Interactive, blaxxun, DRaW Computing, Cybelius, Perceptronics, Quadramix, Sony, NIST, Lucid Actual, Trapezium, NPS sind an dem Web3D-Konsortium beteiligt.

Es befinden sich im Moment 2 Standardprofile in der Entwicklung:

- ein X3D- Kernprofil, welches die Grundfunktionen eines X3D-Betrachters beschreibt und
- ein X3D-VRML97-Profile

Weiterhin existieren Ideen für benutzerspezifische Profile: So ist zum Beispiel im Moment ein GeoVRML-Profile für geographische Applikationen in Arbeit. Das X3D-Kernprofil enthält Elemente wie 3D-Geometrien, Animationen und Interaktionen sowie Rendering Mechanismen. Dieses Basisprofil unterstützt keine Navigation für einen Browser oder die Nutzung von Scriptsprachen. Die Nutzung von Texturen ist nur für JPEG oder PNG-Grafiken möglich.

```
#VRML V2.0 utf8
Transform {
  children [
    NavigationInfo { headlight FALSE } # We'll add
our own light
    DirectionalLight {        # First child
        direction 0 0 -1      # Light illuminating
the scene
    }

    Transform {               # Second child - a red
sphere
      translation 3 0 1
      children [
        Shape {
          geometry Sphere { radius 2.3 }

          appearance Appearance {
            material Material { diffuseColor 1 0 0 }
# Red
         }
        }
      ]
    }

    Transform {               # Third child - a blue
box
      translation -2.4 .2 1
      rotation     0 1 1  .9
      children [
        Shape {
          geometry Box {}
          appearance Appearance {
            material Material { diffuseColor 0 0 1 }
# Blue
         }
        }
      ]
    }

  ] # end of children for world
}
```

Abb. 11: Ein Beispiel in VRML

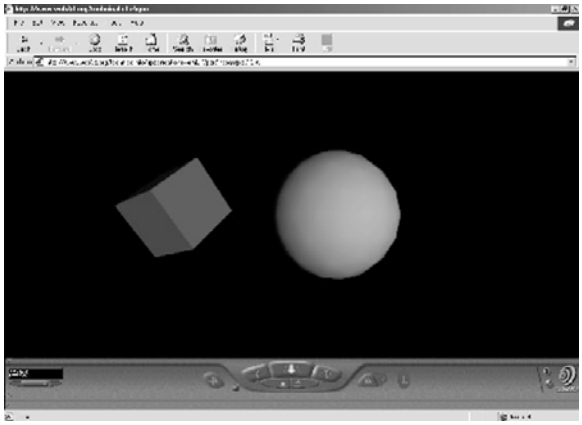

Abb. 12: Beispiel als VRML-Graphik im Netscape

Um bereits jetzt eine breite Öffentlichkeit für sich zu gewinnen, hat das Web3D-Konsortium ein X3D Software Development Kit (SDK mit 500MB Tools) als „Open Source“ veröffentlicht. Diese Editor- und Browsertechnologie beruht auf Java und nutzt u.a. den XML-Editor von IBM, Xeena sowie als Browser den blaxxun Contact browser. Weiterhin hat der Benutzer die Möglichkeit, ein Stylesheet, welches X3D-Dokumente nach VRML97 übersetzt, herunterzuladen.

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE X3D PUBLIC
„http://www.web3D.org/TaskGroups/x3d/translation/x3d-compromise.dtd"
„file://localhost/C:/www.web3D.org/TaskGroups/x3d/translation/x3d-compromise.dtd"
[<!ENTITY % Vrml97Profile  „INCLUDE"> <!ENTITY % CoreProfile    „IGNORE">
<!ENTITY % X3dExtensions  „IGNORE"> <!ENTITY % GeoVrmlProfile „IGNORE">]>
<X3D><Scene><Transform>
       <children>
       <NavigationInfo headlight="false" avatarSize=" 0.25 1.6 0.75" type="&#34;EXAMINE&#34;"/>
       <DirectionalLight/>
       <Transform translation="3.0 0.0 1.0">
       <children>
       <Shape>
       <geometry> <Sphere radius="2.3"/> </geometry>
       <appearance>
                <Appearance>
                         <material> <Material diffuseColor="1.0 0.0 0.0"/> </material>
                </Appearance>
       </appearance>
       </Shape>
       </children>
       </Transform>
       <Transform translation="-2.4 0.2 1.0" rotation="0.0 0.70710677 0.70710677 0.9">
       <children>
       <Shape>
       <geometry> <Box/> </geometry>
       <appearance>
                <Appearance>
                         <material> <Material diffuseColor="0.0 0.0 1.0"/> </material>
                </Appearance>
       </appearance>
       </Shape>
       </children>
       </Transform>
       </children>
       </Transform>
</Scene></X3D>
```

Abb. 13: Das gleiche Beispiel in X3D

Literatur

1. Matthias Schulz: DiML1.0: Dokumentation der Dissertation Markup Language; http://dochost.rz.hu-berlin.de/epdiss/software/dimldoc.pdf
2. World Wide Web Consortium: http://www.w3.org
3. MathML2.0-Working Draft 3 vom 11.Februar 2000: http://www.w3.org/Math
4. SVG1.0 - Working Draft 7 vom 3. Dezember 1999: http://www.w3.org/TR/1999/WD-SVG-19991203/
5. SMIL -Boston - Working Draft vom 15. November 1999: http://www.w3.org/TR/smil-boston/
6. RDF- Recommendation des W3C vom 3.März 1999: http://www.w3.org/RDF/
7. Dublin Core Element Set
8. WML: WAP Forum, 30-April-1998. URL: http://www.wapforum.org/
9. CML: http://www.xml-cml.org/
10. MusicML: http://195.108.47.160/3.0/musicml/index.html
11. BiZTalk-Inititive: http://www.biztalk.org
12. Tad Lane, Scalable Vector Graphics - Web Graphics with Original-Quality Artwork, in: BITS, November 1999, http://lanl.gov/orgs/cic/cic6/bits/november_99/novbits1.html
13. http://wvenus.corel.com/nasapps/DrawSVGDownload/index.html
14. http://www.alphaworks.ibm.com/tech/svgview
15. http://sis.cmis.csiro.au/svg/index.html
16. http://www.xml-cml.org/acs/index.html
17. http://www.openscience.org/jmol/
18. http://www.web3d.org/x3d.html
19. Oliver Bünte: XML auf dem Vormarsch; c't 2000, Heft 10, 200-213
20. http://www.schema.net
21. http://www.empirenet.com/~joseram/index.html
22. http://www.helio.org/products/smil/tutorial/index.html
23. http://www.web3d.org/x3d.html

Susanne Dobratz
susanne.dobratz@rz.hu-berlin.de