

Geography Markup Language (GML) – Eine Einführung aus Informatiksicht ¹

(mit 4 Bildern)

Von Karl Neumann und Silke Eckstein, Braunschweig

ZUSAMMENFASSUNG: Wir geben eine kurzgefasste Einführung in die spezialisierte Auszeichnungssprache GML, indem wir auf deren Trägersprache XML und auf die von der OGC vorgegebenen Strukturen der Simple Features eingehen. Des Weiteren illustrieren wir anhand von Auszügen des GML-Geometrie-Schemas und einer eigenen Feature-Definition wichtige Konstrukte der GML und deren Verwendung.

ABSTRACT: We give a short introduction to the specialized markup language GML. This is done by pointing out some constructs of the GML's carrier language XML, and furthermore, by explaining the main aspects of OGC's so-called simple features. Also, we present parts of GML's geometry schema, and we discuss a self-defined GML feature type.

1 Einleitung

Zu den immer spezieller werdenden Anwendungen im Internet gehören in letzter Zeit auch verstärkt der Austausch von Geodaten und die rechnerbasierte Kartennutzung (Reinhardt 2001, Teege 2001). Dabei scheint sich als Standard zur Repräsentation und zur Darstellung von Geoinformationen die Geography Markup Language, kurz GML, zu entwickeln. Im vorliegenden Text wollen wir daher eine kurzgefasste Einführung in diese spezialisierte Auszeichnungssprache geben.

Dazu wird die Geography Markup Language zunächst knapp in den gegebenen Kontext eingeordnet, und dann gehen wir auf deren Trägersprache die Extensible Markup Language (XML) ein. Diese Sprache zur Generierung von Auszeichnungssprachen bietet mit den Dokument-Typdefinitionen die Möglichkeit, die Syntax von Dokumenten formal festzulegen. Dabei werden Notationsformen benutzt, die ähnlich zu kontextfreien Grammatiken und regulären Ausdrücken sind. Zur Spezifizierung von GML Version 2 wurden die weitergehenden Konstrukte der Sprache XML-Schema benutzt, auf die wir daher ebenfalls kurz eingehen. Danach stellen wir die Grundkonstrukte der "Simple Features" vor, ein Teil des OGC-Standards für den weltweiten Austausch von Geoinformationen. Schließlich werden einige Auszüge aus dem Geometrie-Schema der GML erläutert und ein Beispiel einer eigenen Feature-Definition präsentiert. Zum Schluss folgt ein kurzer

¹Erscheint in *Mitteilungen des Bundesamtes für Kartographie und Geodäsie, Frankfurt M., 2003.*

Ausblick, wie GML zusammen mit den weiteren XML-Anwendungen XSLT und SVG zu einer Standard-Internet-Kartographie führen könnte.

2 Das Umfeld der GML

GML ist eine spezialisierte Auszeichnungssprache, die durch XML erzeugt wird, man sagt auch, GML ist eine XML-Anwendung. Eine Auszeichnungssprache wiederum ist zunächst lediglich eine formale Sprache, also eine Wortmenge über einem festgelegten Alphabet, die spezielle Symbole (Auszeichnungs-Marken oder “Tags”) in den Zeichenfolgen enthält. Als Beispiel für solch ein Wort einer Auszeichnungssprache hier ein Ausschnitt einer HTML-Quelle, die Auszeichnungs-Marken werden in HTML durch spitze Klammern notiert (z.B. “<p>” oder “<i>” und “</i>”):

```
<p><u>Beispiel:</u><p> Einzelne Elemente eines Textes werden durch  
sogenannte <i>Tags</i> gekennzeichnet. Dabei gibt es &ouml;ffnende  
und schlie&szlig;ende Tags. Nicht alle Tags treten paarweise auf,  
f&uuml;r einige wird kein schlie&szlig;ender Tag ben&ouml;tigt.
```

XML ist eine Abkürzung für “Extensible Markup Language”, gelegentlich wird als Langfassung auch noch “Extended Markup Language” angegeben. Beide Namen sind nicht sehr zutreffend, da es sich eigentlich nicht um eine erweiterbare oder erweiterte Auszeichnungssprache handelt, sondern um eine Sprache zur Beschreibung und Erzeugung von Auszeichnungssprachen, also allgemein um eine Metasprache und speziell um eine Metaauszeichnungssprache. Aus Informatiksicht ist XML auch eine kontextfreie Grammatik, die kontextfreie Grammatiken erzeugt.

Es gab schon einen sehr frühen und deutlich komplexeren Vorgänger von XML, nämlich SGML (Standard Generalized Markup Language). Diese Metaauszeichnungssprache wurde Mitte der 80’er standardisiert und besonders im Bereich umfangreicher technischer Dokumentationen eingesetzt, so z.B. mittels der SGML-Anwendung DocBook (vgl. *Walsh* und *Muellner* 1999). Mit der Ausbreitung des Internets wuchs dann der Bedarf nach einer normierten Beschreibungssprache für die einzelnen WWW-Seiten. Hier konnte sich HTML durchsetzen, eine inzwischen sehr bekannte weitere SGML-Anwendung. Auf HTML wird hier nicht näher eingegangen, es soll das oben angegebene Beispiel reichen und der Verweis auf einen kurzen Einführungsabschnitt in *Neumann* und *Tabara* 1999.

Ende der 90’er wurde dann XML als stark vereinfachte Untermenge von SGML eingeführt. XML ist also wie SGML eine Metaauszeichnungssprache, und GML ist eine Anwendung von XML. Mit der spezialisierten Auszeichnungssprache GML wird letztlich ein Beschreibungsrahmen vorgegeben, innerhalb dessen Anwender ihre Geoinformationen strukturieren können. Diese Daten können dann problemlos z.B. über die durch das

Internet gegebene Technologie mit allen anderen Anwendern ausgetauscht werden, die ebenfalls die Auszeichnungssprache GML benutzen.

3 Die Metasprache XML

XML ist, wie bereits kurz ausgeführt, eine Metasprache zur Beschreibung von Klassen von Dokumenten, den XML-Anwendungen (*World Wide Web Consortium* 2000a). Ein konkretes Dokument ist dann eine Ausprägung zu einer solchen Anwendung, oder aus Informatiksicht: ein Wort der von der Anwendung spezifizierten Sprache. Als sehr einfaches Beispiel eines solchen Dokumentes hier die mögliche XML-Repräsentation eines Stadt-Objektes (die Zeilennummern sind zusätzlich eingefügt worden):

```
1 <?xml version="1.0" standalone="no"?>
2 <!DOCTYPE stadt SYSTEM "stadt.dtd">
3 <stadt><name>Braunschweig</name>
4     <einwohner>248000</einwohner>
5     <gebiet><polygon><punkt><x>1935</x><y>2520</y></punkt>
6                             <punkt><x>2565</x><y>2745</y></punkt>
7                             <punkt><x>2835</x><y>2835</y></punkt>
8                             <punkt><x>2970</x><y>2925</y></punkt>
9                             ...
10                            <punkt><x>1935</x><y>2520</y></punkt>
11                        </polygon>
12    </gebiet>
13 </stadt>
```

Nach der Spezifikation der XML-Versionsnummer und der Angabe, dass eine externe Dokumententypdefinition benutzt wird (Zeile 1), folgt in der 2. Zeile die Bekanntgabe dieses Dokumententyps ("stadt"), den wir weiter unten diskutieren. Mit diesem Dokumententyp wird die Ausprägung "Braunschweig" beschrieben. Braunschweig hat den Namen "Braunschweig" (Zeile 3), 248.000 Einwohner und als Stadtgebiet das hier angegebene geschlossene Polygon, wobei ein Polygon aus Punkten und diese aus XY-Koordinaten bestehen (Zeilen 4–13). Die auftretenden Auszeichnungs-Marken, z.B. stadt, name, polygon, werden in einer zugehörigen Dokumententypdefinition (kurz DTD) definiert:

```
1 <!ELEMENT stadt (name, einwohner?, gebiet)>
2 <!ELEMENT name (#PCDATA)>
3 <!ELEMENT einwohner (#PCDATA)>
4 <!ELEMENT gebiet (polygon+)>
5 <!ELEMENT polygon (punkt, punkt, punkt, punkt, punkt*)>
6 <!ELEMENT punkt (x, y)>
7 <!ELEMENT x (#PCDATA)>
8 <!ELEMENT y (#PCDATA)>
```

In dieser DTD legen wir fest, dass eine Stadt einen Namen, eine optionale Angabe der Einwohneranzahl (einwohner?) und ein Gebiet hat (Zeile 1). Die Struktur der so eingeführten Elemente wird weiter verfeinert. Im Prinzip handelt es bei der Definition der Elemente um eine Grammatik, die die Elemente als Nichtterminalsymbole einführt, wobei reguläre Ausdrücke benutzt werden können. Z.B. haben wir die Forderung, dass Polygone mindestens 4 Punkte enthalten müssen durch eine Sequenz mit viermaligem Auftreten des Symbols “punkt” gefolgt von “punkt*” dargestellt (Zeile 5). Alle Elemente, und damit alle Nichtterminalsymbole, werden letztlich auf Zeichenketten zurückgeführt (“#PCDATA”). Solche Dokumententypdefinitionen beschreiben immer hierarchische Strukturen (also Bäume) und können gut durch UML-Klassendiagramme veranschaulicht werden. Weiteres zu UML und den verschiedenen Diagrammart kann z.B. *Neumann* und *Eckstein* 2000 entnommen werden. In Bild 1 ist das UML-Klassendiagramm der Stadt-Dokumententypdefinition dargestellt. Die verschiedenen Elemente wie “polygon”, “punkt” etc. sind hier jeweils als Komponenten modelliert.

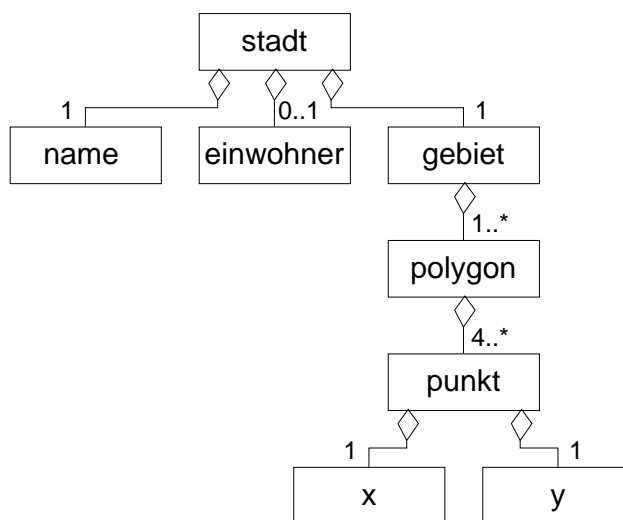


Bild 1 - UML-Klassendiagramm der DTD für Städte

Die XML-Dokumententypdefinitionen sind in den Strukturierungsmöglichkeiten noch etwas eingeschränkt, z.B. gibt es nur einen Datentyp (Zeichenketten), und man kann auch Einschränkungen wie “ein Polygon muss mindestens 4 Punkte enthalten” nur recht indirekt ausdrücken, wie weiter oben am Beispiel der Stadt-Definition gezeigt. Es wurde daher eine neue Schemabeschreibungssprache entwickelt, XML-Schema (vgl. z.B. *World Wide Web Consortium* 2001 und *Hansch et al.* 2002), die weitergehende Strukturierungen erlaubt: Angeboten werden zahlreiche Datentypen (integer, decimal, float, boolean, time, string), Einschränkungen (pattern, enumeration, maxOccurs, default), Typkonstruktoren (simpleType, complexType, sequence, choice, group), Schlüssel und Fremdschlüssel sowie ein Vererbungskonzept. Mit diesen zahlreichen Konstrukten können durch XML-Schema ähnlich komplexe Strukturen dargestellt werden, wie mit UML-Klassendiagrammen (*Eckstein* und *Eckstein* 2003). Weitere Einzelheiten zu XML können etwa den Büchern *Ray* 2001, *Harold* und *Means* 2002 sowie *Kazakos et al.* 2002 entnommen werden.

4 Die Simple Features des OGC

Das Open GIS Consortium (OGC) wurde 1994 mit dem Ziel gegründet, einen Standard für den weltweiten Austausch von Geoinformationen zu erarbeiten. Inzwischen umfasst das OGC ca. 200 Firmen, Behörden sowie Forschungseinrichtungen, und es wurde mit der sogenannten "Abstract Specification" (*Open GIS Consortium 2002*) ein solcher Standard vorgelegt. In dessen Mittelpunkt stehen geographische "Features", das sind Beschreibungen von Geoobjekten in einem Bezugssystem. Der gesamte Standard umfasst 16 meist recht umfangreiche Teilbereiche, z.B. Featuregeometrie, räumliche Referenzsysteme oder Beziehungen zwischen Features. Zur Umsetzung als XML-Anwendung wurde eine kleinere Teilmenge ausgewählt, die "Simple Features" genannt wird. Dabei handelt es sich um Merkmale (features) mit geometrischen Eigenschaften, die nur auf "einfachen" Geometrien beruhen: Die zugehörigen Koordinaten sind höchstens zweidimensional, und Strecken zwischen zwei Stützpunkten werden stets als Gerade angenommen. In Bild 2 ist das UML-Klassendiagramm eines Ausschnitts der Geometrien dargestellt, die hier erlaubt sind.

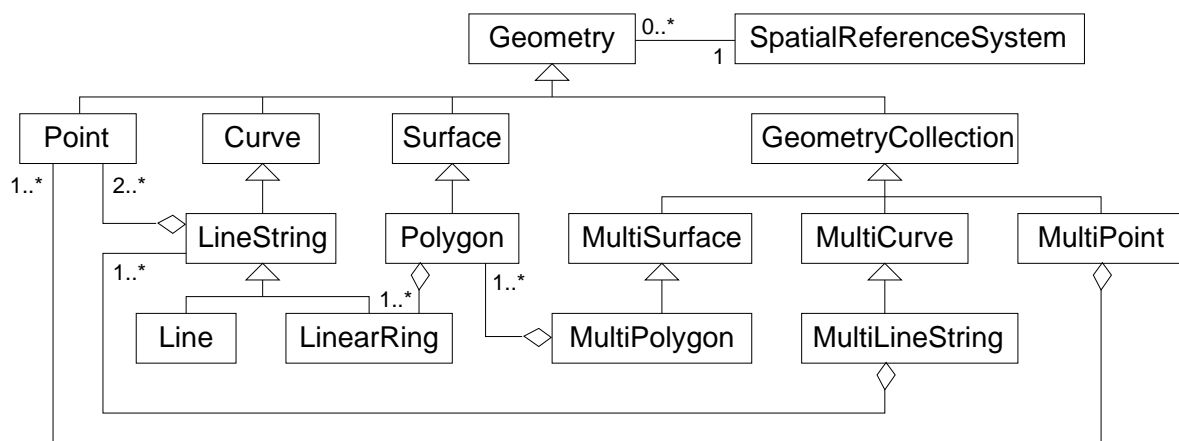


Bild 2 - Geometrien der Simple Features als UML-Klassendiagramm

Neben den geometrischen Eigenschaften können Merkmale noch weitere Eigenschaften haben, z.B. kann eine Stadt einen Namen und die Angabe der Einwohneranzahl aufweisen. Aus Informatiksicht entsprechen Merkmalklassen Objektklassen, Eigenschaften sind Attribute, und es gibt geometrische und nicht-geometrische Datentypen auf denen die Attribute beruhen. GML ist nun eine XML-Anwendung, die die Strukturen der Simple Features umsetzt.

5 Features und Geometrien in GML

Während die 1. Version von GML zur Kodierung der Geometrie- und Featureklassen noch ausschließlich auf Dokumententypdefinitionen aufbaut, wird in GML Version 2.0 XML-Schema eingesetzt, und zur besseren Strukturierung werden drei Teilschemata definiert: das Geometrie-Schema (geometry.xsd), das Feature-Schema (feature.xsd) und ein Schema, das die Vernetzung mit externen Quellen unterstützt (xlinks.xsd). In Bild 3 ist der Zusammenhang dieser drei Teilschemata in Form von UML-Paketen skizziert; in *Open GIS Consortium* 2001 werden die Quelltexte der Schemata bereitgestellt.

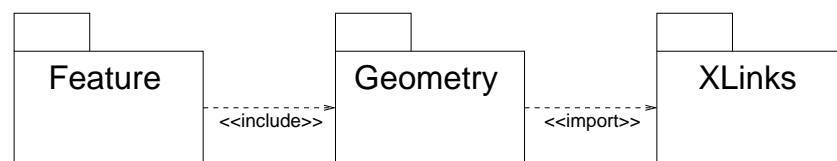


Bild 3 - die drei GML-Teilschemata als UML-Pakete

Im Folgenden geben wir zur Illustration zwei kurze Ausschnitte aus dem Geometrie-Schema an und definieren dann einen eigenen Feature-Typ. Zunächst als Beispiel die GML-Kodierung von Koordinaten. Werte von Koordinaten können entweder als einzelne Zahlen (CoordType) oder als Zeichenketten (CoordinatesType) angegeben werden. Die Definition des Typs CoordType (Zeilen 1–6) ist einfach eine Sequenz der Elemente X, Y und Z, die jeweils Dezimalzahlen sind und von denen nur die X-Koordinate obligat ist.

```
1 <complexType name="CoordType">
2   <sequence> <element name="X" type="decimal"/>
3             <element name="Y" type="decimal" minOccurs="0"/>
4             <element name="Z" type="decimal" minOccurs="0"/>
5   </sequence>
6 </complexType>
7
8 <complexType name="CoordinatesType">
9   <simpleContent>
10    <extension base="string">
11      <attribute name="decimal" type="string"
12                use="default" value="."/>
13      <attribute name="cs" type="string"
14                use="default" value=","/>
15      <attribute name="ts" type="string"
16                use="default" value="&#x20;"/>
17    </extension>
18  </simpleContent>
19 </complexType>
```

Der Typ `CoordinatesType` wird vom Datentyp `String` abgeleitet (Zeile 10) und erhält drei weitere Attribute (Zeilen 11–16): `decimal` bezeichnet das Symbol für den Dezimalpunkt, `cs` das Symbol zur Trennung einzelner Koordinatenwerte (`coordinate separator`) und `ts` das Symbol zur Trennung zusammengehöriger Koordinaten (`tupel separator`). Mit den in der Definition angegebenen Voreinstellungen für die Separatoren und den Dezimalpunkt wäre daher z.B. die Zeichenkette “0.0,0.0 100.0,0.0 100.0,100.0 0.0,100.0 0.0,0.0” eine beabsichtigte Angabe von Koordinatenwerten. Allerdings wäre auch jede andere Zeichenkette syntaktisch richtig, weil bei der Definition auf eine einschränkende Mustervorgabe verzichtet wurde.

Als etwas komplexeren Ausschnitt aus dem GML-Geometrieschema soll noch die Definition des vorgegebenen Typs `PolygonType` diskutiert werden (Zeilen 1–23). Polygone werden in GML als Subtyp des abstrakten Typs `AbstractGeometryType` definiert (Zeile 3) und erben damit dessen Merkmale, das sind ein optionales Identifikationsattribut und ein optionales Zeigerattribut auf ein räumliches Bezugssystem. Auf diese Attribute und auch auf weitere von GML bereitgestellte Daten- und Objekttypen soll hier aber nicht näher eingegangen werden. Polygone haben weiter eine äußere Begrenzung (Zeilen 5–11) und können mehrere innere eingeschlossene Flächen aufweisen (Zeilen 12–19). Alle diese Flächen sind vom Typ `LinearRing`, der ein weiterer GML-Typ ist und letztlich auf Koordinaten beruht, die ihrerseits vom Typ `CoordType` oder vom Typ `CoordinatesType` sind.

```

1 <complexType name="PolygonType">
2   <complexContent>
3     <extension base="gml:AbstractGeometryType">
4       <sequence>
5         <element name="outerBoundaryIs">
6           <complexType>
7             <sequence>
8               <element ref="gml:LinearRing"/>
9             </sequence>
10          </complexType>
11        </element>
12        <element name="innerBoundaryIs"
13                  minOccurs="0" maxOccurs="unbounded">
14          <complexType>
15            <sequence>
16              <element ref="gml:LinearRing"/>
17            </sequence>
18          </complexType>
19        </element>
20      </sequence>
21    </extension>
22  </complexContent>
23 </complexType>

```

Mit den hier nur beispielhaft skizzierten Sprachmitteln des GML-Geometrieschemas und des GML-Featureschemas können nun eigene Anwendungswelten dargestellt werden, d.h. vor allem, neue Featuretypen definiert werden. Wir zeigen das im Folgenden an dem relativ einfachen Featuretyp SeeTyp (Zeilen 1–13). Inhaltlich modellieren wir dabei Seen durch Angabe eines Namens, ihrer maximalen Tiefe, ihrer Fläche und ihrer räumlichen Ausdehnung, wozu wir den Typ PolygonType benutzen. Weil Polygone in GML durch eine umschließende Fläche und beliebig viele innen liegende Flächen charakterisiert sind, berücksichtigt bereits unsere gewählte Geometrie Seen mit Inseln.

```

1 <element name="See" type="SeeTyp"/>
2 <complexType name="SeeTyp">
3   <complexContent>
4     <extension base="gml:AbstractFeatureType">
5       <sequence>
6         <element name="seeName" type="string"/>
7         <element name="maxTiefe" type="integer"/>
8         <element name="flaeche" type="decimal"/>
9         <element ref="gml:extentOf"/>
10      </sequence>
11    </extension>
12  </complexContent>
13 </complexType>

```

Da unser SeeTyp ein Featuretyp (im OGC-Sinne der Simple Features) sein soll, muss er als Erweiterung der abstrakten GML-Klasse AbstractFeatureType definiert werden (Zeilen 4–11). Die angesprochenen Angaben wie Name, maximale Tiefe usw. werden als Sequenz von Elementen spezifiziert (Zeilen 6, 7, 8). Beim Geometrie-Element wird ein Verweis auf die in GML vordefinierte Klasse extentOf benutzt, die wie die ebenfalls vordefinierte Klasse coverage ein Synonym für die Benutzung des Typs PolygonType ist.

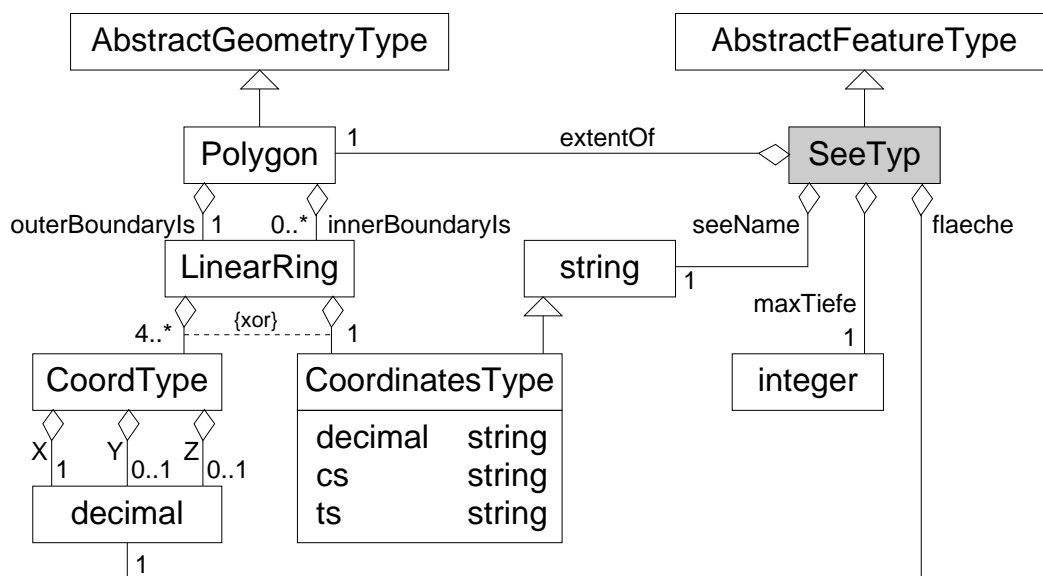


Bild 4 - selbst definierter Featuretyp und GML-Typen

In Bild 4 sind die Zusammenhänge zwischen dem eigenen Featuretyp SeeTyp und einigen von GML bereitgestellten Typen als UML-Klassendiagramm dargestellt. Auch der weiter oben diskutierte Typ Polygon wurde in das Diagramm aufgenommen. Bei realistischen Anwendungswelten, die vielleicht aus 20 bis 30 selbst definierten Featuretypen und auch einigen Assoziationstypen bestehen, ergeben sich entsprechend größere und kompliziertere Klassendiagramme.

6 Ausblick

Im vorliegenden kurzen Text haben wir Konzepte der spezialisierten Auszeichnungssprache GML skizziert, die als XML-Anwendung die Simple Features umsetzt, einen Teil der "Abstract Specification" des OGC. GML ist relativ komplex, weil zum einen die ganze Begriffswelt und die Sprachkonstrukte von XML zu berücksichtigen sind und zum anderen auch noch die vorgegebenen Strukturen und Zusammenhänge der Simple Features. Dennoch scheint es sich abzuzeichnen, dass GML der Standard für den internationalen Austausch von Geodaten wird.

Und GML könnte vielleicht sogar zur weltweiten Standard-Internet-Kartographie führen. Dazu müsste in einem ersten Schritt festgelegt werden, welche Geoobjekte überhaupt kartographisch dargestellt werden, und diese Geoobjekte wären in GML einmal zu spezifizieren. In einem zweiten großen Schritt könnten dann die kartographischen Darstellungen dieser gesammelten Feature-Klassen festgeschrieben werden. Dazu würde sich der Einsatz von XSLT (Extended Stylesheet Language Transformation, *World Wide Web Consortium* 1999) anbieten, einer speziellen XML-Anwendung, die XML-Dokumente in XML-Dokumente transformieren kann. Als Ziel dieser Transformation käme wiederum eine weitere XML-Anwendung in Betracht: SVG (Scalable Vector Graphics, *World Wide Web Consortium* 2000b), ein Graphikstandard, der zweidimensionale Vektorgraphik unterstützt. Mit SVG können z.B. vordefinierte und freie Formen erzeugt werden und etwa auch Farbübergänge. Für die verbreiteten Web-Browser gibt es auch bereits entsprechende Plug-Ins.

Insgesamt könnten so alle möglichen Geoobjekte, die in GML kodiert irgendwo weltweit im Internet vorhanden sind, automatisch kartographisch dargestellt werden: Dazu müssten die gefundenen GML-Dokumente jeweils aufgrund der passenden Transformation automatisch in SVG-Dokumente transformiert und dann vom örtlichen Web-Browser nur noch angezeigt werden. Der ganze Prozess ist aus Informatiksicht lediglich eine Transformation von Bäumen in Bäume, also zumindest vom Prinzip her nicht übermäßig kompliziert.

Literatur

- Eckstein, R.; Eckstein, S.:* XML und Datenmodellierung. dpunkt.verlag, erscheint 2003.
- Hansch, M.; Kuhlins, S.; Schader, M.:* XML-Schema. Informatik-Spektrum, Nr. 25(5), 2002, pp. 363–366.
- Harold, E.R.; Means, W.S.:* XML in a Nutshell. O’Reilly, 2002.
- Kazakos, W.; Schmidt, A.; Tomczyk, P.:* Datenbanken und XML. Springer, 2002.
- Neumann, K.; Eckstein, S.:* Einführung in die Unified Modeling Language am Beispiel von ATKIS. Mitteilungen des Bundesamtes für Kartographie und Geodäsie, Band 17, Frankfurt M. 2000, pp. 81–88.
- Neumann, K.; Tabara, C.:* Effiziente Darstellung wissenschaftlicher Arbeit: Ein Informationssystem zur Generierung von WWW-Seiten und Berichten. Mitteilungen des Bundesamtes für Kartographie und Geodäsie, Band 2, Frankfurt M. 1999, pp. 71–81.
- Open GIS Consortium (OGC):* Geography Markup Language (GML) 2.0. 2001, <http://www.opengis.net/gml/01-029/GML2.html>.
- Open GIS Consortium (OGC):* The OpenGIS Abstract Specification. 2002, <http://www.opengis.org/pressrm/summaries/20010812.TS.AbstrSpec.htm>.
- Ray, E.T.:* Einführung in XML. O’Reilly, 2001.
- Reinhardt, W.:* Zur Bedeutung der Internettechnologie in Geoinformationssystemen. Tagungsband “Internationale Geodätische Woche”, Obergurgl, 2001.
- Teege, G.:* Geodaten im Internet – Ein Überblick. Informatik-Spektrum, Nr. 24(4), 2001, pp. 193–206.
- Walsh, N.; Muellner, L.:* DocBook: The Definitive Guide. O’Reilly, 1999.
- World Wide Web Consortium (W3C):* XSL Transformations (XSLT). Version 1.0, 1999, <http://www.w3.org/TR/1999/REC-xslt-19991116>.
- World Wide Web Consortium (W3C):* Extensible Markup Language (XML). Version 1.0 (2nd Ed.), 2000, <http://www.w3.org/TR/1998/REC-xml-20001006>.
- World Wide Web Consortium (W3C):* Scalable Vector Graphics (SVG) 1.0 Specification. 2000, <http://www.w3.org/TR/2000/WD-SVG-20000629/>
- World Wide Web Consortium (W3C):* XML Schema – Part 0,1,2. 2001, <http://www.w3.org/TR/2000/REC-xmlschema-0|1|2-20010502>.