

**Datenaustauschnittstelle auf XML-Basis für Umweltdaten  
– insbesondere für die  
Landesweite Datenbank des Landes Niedersachsen**

**Dr. Gergely Lukacs, Claus Hofmann  
disy Informationssysteme GmbH  
www.disy.net**

## **Executive Summary**

**Auf Basis von XML-Standards werden mit freien bzw. marktüblichen Werkzeugen Dateien geschaffen, die maschinell oder durch Menschen bearbeitet, überprüft und interpretiert werden können.**

**XML-basierte Datenaustauschnittstellen haben mehrere Vorteile: Die Schnittstellendefinition z. B. im XML-Schema erlaubt die Überprüfung der Daten auch beim Datenlieferanten. Dadurch werden aufwendige Rückfragen des Datenempfängers vermieden. Die Daten können auch vom menschlichen Benutzer interpretiert und überprüft werden.**

**Diese Studie wurde durch die Datenübernahme in die Landesweite Datenbank für die Wasserwirtschaft des Landes Niedersachsen in ein Data Warehouse motiviert. Die Aussagen sind aber auch für andere Datenübernahmeszenarien zutreffend.**

**Die Vorteile von XML sind nicht automatisch gegeben. Bei der Schnittstellendefinition müssen einige Kriterien berücksichtigt werden: (1) es müssen Konventionen eingeführt werden; (2) Datentypen und Referenzangaben müssen restriktiv definiert werden; (3) bei inkrementeller Datenübernahme sind spezielle Konstrukte notwendig; (4) gemeinsame Schemateile bzw. gemeinsame Schlüssellisten müssen identifiziert und benutzt werden; (5) die Dateigrößen müssen eingeschränkt werden können. Es ist auch möglich, XML-Schnittstellen so zu definieren, dass die potenziellen Vorteile ungenutzt bleiben.**

**Auf Basis der dargestellten Konzepte und der umfangreichen Praxiserfahrungen über Datenaustauschnittstellen wurde die Datenaustauschnittstelle zwischen der Fachanwendung für die Einleiterüberwachung und der Landesweiten Datenbank definiert.**

## 1. Einführung

Die Landesweite Datenbank des Landes Niedersachsen (LDB) fasst die für die Europäische Wasserrahmenrichtlinie und für andere landesweite Aufgaben relevante Daten aus ca. 15. Fachanwendungen bzw. Datenquellen zusammen. Die LDB entspricht einem Data Warehouse: Die Daten werden weiterhin in den Fachanwendungen erfasst und gepflegt, aber sie werden periodisch in die nur für Auswertungen benutzte LDB – ein Data Warehouse – übernommen. Die Übernahme geschieht mittels Einlagerungsprozessen. (Die gängige englische Abkürzung: ETL – extraction, transformation and loading, d. h. extrahieren, umwandeln und laden).

Neben der einheitlichen Modellierung und einem geeigneten Zugriffs- und Analysewerkzeug – disy Cadenza – ist eine besonders wichtige Säule der LDB die relativ einfache und flexible Datenübernahme. Letztere stellt aber durch die große Heterogenität der Datenquellen sowohl für die Entwickler als auch für den Administrator der LDB eine Herausforderung dar. Standardisierte Datenaustauschnittstellen können die Heterogenität reduzieren und dadurch die unnötige Komplexität reduzieren.

Mit Datenaustauschnittstellen auf der Basis von XML-Standards und –Technologien, kann die Heterogenität beim Datenaustausch deutlich reduziert und die Schnittstellen formal definiert werden. XML-Daten können mit marktüblichen bzw. freien XML-Werkzeugen wie XMLSpy gegen eine Schnittstellendefinition validiert werden, d. h. sowohl bei der Datenquelle, wenn die Daten erzeugt werden, als auch bei der Datensinke (LDB) wenn sie eingespielt werden. Dadurch werden viele proprietäre Software-Entwicklungsarbeiten, zeitaufwendige Überprüfung der Daten durch den Administrator der Datensinke und Rückfragen an den Datenlieferanten überflüssig, bzw. die Datenqualität wird verbessert. Wenn die Datenaustauschnittstellen für mehrere Fachanwendungen gemeinsam entwickelt werden, können noch mehr Synergieeffekten genutzt werden: Die gemeinsamen Teile müssen nur einmal spezifiziert und deren Bearbeitung nur einmal umgesetzt werden.

Ziel dieser Studie ist es, die Aspekte beim Entwurf einer Datenaustauschnittstelle auf XML-Basis zu beschreiben. XML sollte bei der Datenübernahme aus neuen Fachanwendungen weitgehend benutzt werden, wie dies in vielen anderen Anwendungsfeldern bereits praktiziert wird.<sup>1</sup> Es werden zuerst Basistechnologien und Standards dann die wenigen vorhandenen umweltrelevanten Standards zusammengefasst. Im Anschluss folgen allgemeine Überlegungen zu den Schnittstellendefinitionen und zu allgemein einsetzbaren Schemakonstrukten. Danach werden zu vermeidende Fallen beschrieben, gefolgt von einer Zusammenfassung.

---

<sup>1</sup> Beim Austausch von Massendaten mit einer sehr einfachen Struktur ist der Einsatz von XML u. U. nicht vorteilhaft.

## 2. Kern-Standards

XML (eXtensible Markup Language, [1]) ist ein technischer Standard, der es erlaubt, strukturierte Daten bzw. Dokumente sowohl maschinell verarbeitbar, als auch für den Menschen lesbar zu beschreiben. Er ist unabhängig von Plattformen und Programmiersprachen.

Beispiel: Ausschnitt aus einer XML-Datei

```
<Lage>
  <Lokation>
    <Bezugssystem>GK Zone 2</Bezugssystem>
    <Hochwert>3405545</Hochwert>
    <Rechtswert>2456444</Rechtswert>
    <Erfassungsguete>1</Erfassungsguete>
  </Lokation>
  <GewaesserBezug>
    <GewaesserNr>233</GewaesserNr>
    <GewaesserAbschnittNr>2334</GewaesserAbschnittNr>
    <GewaesserAbschnittNrErfassungsguete>aus den
Koordinaten errechnet</GewaesserAbschnittNrErfassungsguete>
    <DirektamGewaesser>ja</DirektamGewaesser>
  </GewaesserBezug>
</Lage>
```

Eine XML-Datei besteht aus Elementen, beispielsweise `<Bezugssystem>GK Zone 2</Bezugssystem>`, wobei ein Element aus einem Öffnungstag (`<Bezugssystem>`), einem Schlusstag (`</Bezugssystem>`) und einem Wert besteht. Der Wert ist in diesem Fall eine Zeichenkette (GK Zone 2), könnte aber auch ein weiteres (geschachteltes) Element sein.

Für den technischen Standard XML können inhaltliche Standards definiert werden, die u. A. die erlaubten Elemente, deren Datentypen, Kardinalitäten, eindeutige Schlüssel und Fremdschlüssel definieren. Wie diese inhaltlichen Standards definiert werden müssen, wird durch weitere XML-Kernstandards definiert. Der heute relevanteste Standard ist das XML-Schema [2].<sup>2</sup>

Beispiel: Ausschnitt für ein XML-Schema:

```
<xsd:complexType name="LageType">
  <xsd:sequence>
    <xsd:element name="Lokation" type="lg:LokationType"/>
    <xsd:element name="GewaesserBezug"
```

---

<sup>2</sup> Der Vorgänger vom XML-Schema, die Document Type Definition, kurz DTD, hat für aktuelle Vorhaben keine Bedeutung mehr.

```
        type="lg:GewaesserBezugType"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="LokationType">
    <xsd:sequence>
        <xsd:element name="Bezugssystem"
            type="slbs:BezugssystemType"/>
        <xsd:element name="Hochwert" type="xsd:decimal"/>
        <xsd:element name="Rechtswert" type="xsd:decimal"/>
        <xsd:element name="Erfassungsguete"
            type="xsd:decimal" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="GewaesserBezugType">
    <xsd:sequence>
        <xsd:element name="GewaesserNr"
            type="xsd:nonNegativeInteger" minOccurs="0"/>
        <xsd:element name="GewaesserAbschnittNr"
            type="xsd:nonNegativeInteger" minOccurs="0"/>
        <xsd:element name="GewaesserAbschnittNrErfassungsguete"
            type="xsd:string" minOccurs="0"/>
        <xsd:element name="DirektamGewaesser"
            type="bs:BooleanGermanType" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

Eine XML-Schema-Datei ist ebenfalls eine XML-Datei, die aber nur bestimmten Elemente beinhalten kann. In diesem Beispiel wurde ein zusammengesetzter Datentyp (`LageType`) definiert, der aus den Elementen (`Lokation`) und (`GewaesserBezug`) besteht. Diese bestehen wiederum aus mehreren Elementen.

Falls das entsprechende XML-Schema vorhanden ist, kann eine XML-Datei dahingehend validiert werden, ob sie dem Schema entspricht. Die Validierung kann im Batch-Betrieb z. B. in einem Einlagerungsprozess (ETL-Prozess) eines Data Warehouse, wie die LDB, oder beispielsweise in einem Editor durchgeführt werden. Der Editor bringt dann den Benutzer direkt zu den Stellen, an denen die XML-Datei vom Schema abweicht.

Mit XML-Schema sind u. A. auch folgende fortgeschrittene Validierungen möglich:

- Liegt ein numerischer Wert in einem Intervall?
- Kommt eine Zeichenkette aus einer Schlüsseliste?
- Stimmt eine Zeichenkette mit einem Muster überein (z. B. Messwert mit optionalem Vorzeichen: „0,1“ oder „<0,1“)?
- Ist ein Datumswert ein gültiges Datum (mit Berücksichtigung von Schaltjahren)

Es gibt auch spezifische Validierungsanforderungen, die mit XML-Schema nicht möglich sind, wie z. B.:

- Liegt der Messwert im Trend?
- Liegt der Messwert in dem Intervall, das den gültigen Werten des aktuellen Parameters entspricht? (Falls alle möglichen Parameter im Schema aufgelistet sind, kann dies gemacht werden, falls beliebige Parameter übernommen werden können, nicht.)

### 3. Bisherige Umweltrelevante Schemadefinitionen

In vielen Anwendungsfeldern z. B. Finanzmärkte, Versicherungen, Telekommunikation, industrielle Produktion sind XML-Schemata weitgehend definiert und so ist ein standardisierter Datenaustausch möglich. Einige XML-Sites sind darauf spezialisiert, definierte XML-Schemata aus unterschiedlichen Anwendungsfeldern zu publizieren. Dadurch fördern sie deren Verbreitung und Kompatibilität [3, 4].

Im Umweltbereich gibt es, trotz der bisherigen Projekte, nur einige wenige isolierte Ansätze. Diese befinden sich vor Allem im Metadatenbereich für Datenaustauschnittstellen auf XML-Basis. Übersichten, die allerdings kaum konkrete Schemata beinhalten, sind in [5, 6, 7] zu finden. Im Umfeld der Geodaten hat sich GML (Geography Markup Language) des Open GIS Consortiums [8] etabliert.

## 4. Datenaustauschnittstelle für die LDB

### 4.1 Allgemeine Vorgaben

Ziel der Schemadefinitionen für Datenaustauschnittstellen ist, dass sie gut einsetzbar, ausreichend stabil und auch wiederverwendbar sind.

Die Heterogenität der Daten und die Komplexität der Schemata im Umweltbereich führen dazu, dass es tatsächlich schwierig ist, allgemeine Datenaustauschnittstellen zu definieren. Es ist aber mit den Strukturierungsmöglichkeiten von XML durchaus möglich, auch im Umweltbereich *generisch einsetzbare Teilschemata* zu definieren. Um dies zu erreichen, müssen die allgemeinen Teile identifiziert und dann bei der Schnittstelle als allein referenzierbare Teile definiert werden. Sie dürfen also nicht in komplexen, nicht allgemeingültigen Schemateilen eingebettet sein. Zwei Sorten von allgemeinen Teilschemata können definiert werden:

- Allgemeine Schemateile, bei denen nur das Schema, aber nicht der Inhalt allgemein ist (z. B. die Beschreibung der geografischen Lage eines Objektes mit gleichen Attributen, aber unterschiedlichen Werten) und
- Allgemeine Schemateile, bei denen der Inhalt identisch ist, beispielsweise aus einer festen Schlüsselliste (z. B. Schlüssellisten für die Einheiten oder für die Parameter).

Diese generisch einsetzbaren Teilschemata müssen dann an allen Stellen, wo dies möglich ist, benutzt werden.

In den Schemadefinitionen muss auch *allgemeinen Namens- und Strukturierungskonventionen* gefolgt werden, um einheitlichere Schnittstellen zu

bekommen. Auf Basis der Namenskonventionen der LDB können Namenskonventionen auch für die XML-Schnittstellen definiert werden. Folgende Namenskonventionen wurden bereits eingeführt:

- Nichtnumerische Schlüsselemente enden mit „ID“, numerische Schlüsselattribute mit „Nr“.
- Für die Versionierung wird das Element „Version“ benutzt.

Es wäre auch wünschenswert, dass der Datenlieferant möglichst die gleichen **fachanwendungsübergreifenden Schlüssellisten** z.B. für Parameter, Einheiten und Behörden benutzt, wie in der LDB. Aus unterschiedlichen, vor allem organisatorischen Gründen, kann dies allerdings oft nicht gewährleistet werden. In diesem Fall werden dann die Schlüssellisten der Anwendungen benutzt. Bei fachanwendungsspezifischen Schlüssellisten hat die Fachanwendung die Hoheit über die Werte. Es wäre ebenfalls wünschenswert, ist aber derzeit im Umfeld der LDB noch nicht realistisch, dass die Fachanwendungen vereinheitlichte Schlüsselattribute benutzen.

Die Schnittstelle muss genügend **restriktiv** sein. Der potenzielle Vorteil von XML besteht darin, dass der Datenlieferant mit der Definition der Schnittstelle mittels XML-Schema in der Lage ist, selbst die Daten zu validieren. Dadurch können Fehler näher zu der verantwortenden Stelle, nämlich dem Datenlieferanten, entdeckt und unnötige, zeitaufwändige Rückfragen vermieden werden. Alle Möglichkeiten von XML müssen also angewendet werden, um den Vorteil, der falsche Daten bei der Übernahme filtert, tatsächlich zu nutzen. Dies bezieht sich sowohl auf die Datentypen, als auch auf die Fremdschlüsselbeziehungen.

Die Datentypen müssen in der Schnittstellendefinition genau mit Länge bzw. Genauigkeit festgelegt werden. Es reicht nicht aus, den Standardtypen „string“ ohne Längenangabe anzugeben. Der Grund dafür ist, dass die übernommenen Daten in eine Datenbank, die LDB, übernommen werden müssen, in der die Datentypen festgelegt sind. Sonst filtert die Schnittstellendefinition die Daten nicht aus, die nachher nicht bearbeitet werden können. Das Abschneiden der Daten bei der Datenspeicherung, die LDB, wäre aufwendig und semantisch problematisch, da dadurch Inhalte verloren gehen oder sogar inhaltlich falsch werden können (z. B. Abschneiden von Ziffern einer Zahl).

Ein XML-Schema bietet zwei Möglichkeiten für die Abbildung der Referenzen zwischen Elementen: Implizit durch Schachtelung der XML-Elemente und explizit durch key und keyref Elemente. Diese Möglichkeiten müssen genutzt werden, damit die Schnittstelle genügend restriktiv ist.

Die Wiederverwendbarkeit muss, zumindest an den allgemeinen Stellen, mit einer ausreichenden **Schemadokumentation** unterstützt werden. Das XML Schema stellt die Elemente `documentation` für Menschen und `appinfo` für Anwendungen bereit. Beide müssen in einem `annotation` Element eingebettet werden.

Daten können immer komplett oder inkrementell übernommen werden. Bei der **inkrementellen Übernahme** wird nicht der komplette Datenbestand, sondern nur der, der seit disy Informationssysteme GmbH

der letzten Datenübernahme entstanden ist oder geänderte Daten, geladen. Dadurch kann die zu transportierende Datenmenge bzw. die Ladezeit deutlich verringert werden.

Die seit der letzten Datenübernahme erfassten Daten zu übernehmen, ist leider nicht immer akzeptabel. Beispielsweise werden Messwerte auch nachträglich korrigiert, wenn es sich um einen fehlerhaften Wert handelt. Vielmehr müssen die seit der letzten Datenübernahme korrigierten Daten übernommen werden. Dazu ist es wiederum notwendig, dass die Datensätze mit einem Zeitstempel versehen werden. Datensätze dürfen bei der inkrementellen Datenübernahme nicht einfach gelöscht, sondern nur als gelöscht gekennzeichnet werden, damit sie auch in der Zieldatenbank gelöscht bzw. entsprechend markiert werden können. Sonst ist der gelöschte Datensatz in den übernommenen Daten nicht vertreten und kann in der Zieldatenbank nicht gelöscht werden.

## 4.2 Schemadefinitionen

Die dargestellten Konzepte wurden am Beispiel der Datenaustauschnittstelle, die über die Fachanwendung „Einleiterüberwachung 3“ die LDB mit Daten versorgt, validiert. Die Fachanwendung wurde von der Firma Risa GmbH entwickelt. Bei der Festlegung dieser Schnittstelle durch die Firma disy Informationssysteme GmbH, wurden die inhaltlich wiederverwendbaren Schemateile (Datentypen mit Längenangaben, die Lage eines Objektes und die Einheiten) separat definiert.

Beispiel: Datentypen mit Längenangaben

```
<xsd:simpleType name="string64">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="64"/>
  </xsd:restriction>
</xsd:simpleType>
```

Die Definition der Lage beinhaltet einen Teil für die Position (räumliches Bezugssystem, Koordinaten, Genauigkeit der Koordinaten). Da aus der Position der Bezug zu einem Fließgewässer noch nicht direkt folgt, kann dies in dem zweiten Teil angegeben werden. Dazu gehören Gewässer-Nr. laut LAWA-Numerierung, Gewässerabschnitt-Nr., die Erfassungsgüte der Gewässerabschnitt-Nr. sowie die Angabe, ob das Objekt direkt am Gewässer oder nur im Teileinzugsgebiet liegt.

Beispiel: Ausschnitt aus dem XML-Schema der Lage-Angaben

```
<xsd:complexType name="LageTyp">
  <xsd:sequence>
    <xsd:element name="OrtsBezug"
      type="ldb:OrtsBezugTyp"/>
    <xsd:element name="GewaesserBezug"
      type="ldb:GewaesserBezugTyp" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="OrtsBezugTyp">
```

```
<xsd:sequence>
  <xsd:element name="Bezugssystem"
              type="ldb:BezugssystemTyp"/>
  <xsd:element name="Rechtswert" type="ldb:decimal9_2"/>
  <xsd:element name="Hochwert" type="ldb:decimal9_2"/>
...
```

Die Liste der Einheiten ist ebenfalls allgemein definiert. Bei Elementen, die Werte mit einer bestimmten Einheit beschreiben, soll ein Attribut die Einheit aus der vordefinierten Liste angeben.

Beispiel: Ausschnitt aus dem XML-Schema der Einheiten

```
<xsd:simpleType name="EinheitTyp">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="M3A">
      <xsd:annotation>
        <xsd:documentation xml:lang="de"> Kubikmeter pro
Jahr </xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
...
```

Die spezifische Datenaustauschnittstelle zwischen der Fachanwendung und der Landesweiten Datenbank nutzt die beschriebenen allgemeinen Definitionen durchgehend. Schlüssel und Fremdschlüssel wurden auch an allen entsprechenden Stellen definiert.

### 4.3 Negative Beispiele

Aus unserer Praxiserfahrung im Umgang von XML-Schnittstellen sollen hier einige negative Beispiele skizziert werden, die unbedingt vermieden werden müssen:

#### 4.3.1 „Generische Datenaustauschnittstelle“

Es ist auch möglich, eine „generische“ Datenaustauschnittstelle auf XML-Basis zu definieren. Bei einer „generischen“ Schnittstelle können allerdings beliebige Daten über die Schnittstelle ausgetauscht werden. Jegliche Überprüfung, ob die notwendigen Daten da sind, bzw. ob die Daten gültig sind, ist damit nicht möglich. Dadurch wird der wesentliche Vorteil von XML neutralisiert!

Ein typisches Schemakonstrukt wäre z. B.

```
<String ID="ART_ANLAGE">KOMMUNALE_KLAERANLAGE</String>
<String ID="ART_DER_EINLEITUNG">INDIREKTEINLEITUNG</String>
```

In diesem Fall kann z. B. lediglich überprüft werden, ob ein Wert, der eine Zeichenkette sein soll, tatsächlich eine Zeichenkette ist. Ob er bspw. in einer Schlüsselliste enthalten ist, kann allerdings nicht gewährleistet werden.

### 4.3.2 „Generische Verweise“

Eine Schnittstelle kann auch nur an einigen Stellen, z. B. bei den Verweisen (Fremdschlüsselbeziehungen) „generisch“ gestaltet werden. Diese Lösung ist aus dem gleichen Grund ungeeignet, wie die vorher erwähnte vollkommen generische Schnittstelle: Es kann bspw. nicht überprüft werden, ob eine Messstelle zu einer Anlage oder zu einem Nutzungsort zugeordnet werden muss.

## 5. Speicherplatz und Performanz

XML Dateien sind in der Regel deutlich größer als „einfache“ Textdateien, da die Tags relativ groß sind. Im Normalfall ist dies nicht weiter störend. Bei großen Datenmengen jedoch z. B. bei Messwerten, kann die Größe der Dateien zu einem Problem werden, wenn die Daten übertragen bzw. eingespielt werden müssen. Um ein grobes quantitatives Gefühl zu vermitteln, wurden einige Experimente durchgeführt.

Das benutzte XML-Fragment für die Beschreibung eines Messwertes:

```
<GMesswertX>
  <ParameterID>279</ParameterID>
  <ErfassterMesswert>0.4</ErfassterMesswert>
  <FreigegebenerMesswert>0.4</FreigegebenerMesswert>
  <Freigabekennung>1</Freigabekennung>
  <EinheitID>42</EinheitID>
  <unbekannt>9999</unbekannt>
</GMesswertX>
```

Es werden Testdateien generiert, die viele Messwerte enthalten. Die Größen dieser Dateien wurden zuerst ermittelt. Die Dateien wurden gegen eine Schemadefinition validiert. Die Validierungszeiten auf einer 900 MHz Pentium-Maschine mit der freien Bibliothek „Xerces“ wurden ebenfalls gemessen.

Die Datenübernahme in die LDB erfolgt mit der Unterstützung des graphischen ETL-Werkzeugs Oracle Warehouse Builder (OWB). Der OWB enthält auch eine Bibliothek, mit der XML-Dateien in eine Oracle-Datenbank eingelagert werden können. Die Einlagerungszeiten mit dieser Bibliothek wurden ebenfalls getestet.

Anzahl Messwerte	Dateigröße	Validierungszeit (Freie Bibliothek Xerces)	Einlagerung mit dem Oracle Warehouse Builder (JAVA_POOL_SIZE: 167 772 160)

385 750	54,6 MB	25 Sek	Gescheitert
100 000	29 MB		Gescheitert
50 000	15 MB		3 Min. 42 Sek.

Die Dateigrößen sind für die Orientierung auf jeden Fall interessant. Die Validierung ist offensichtlich kein großes Problem. Die Zahlen für die Einlagerung in die LDB beziehen sich auf die Bibliothek, die Teil vom OWB ist und hängen von der aktuellen Konfiguration von Oracle bzw. von den Hardware-Voraussetzungen ab. Da die Prozedur leicht ersetzt werden kann, sollen die Zahlen in der letzten Spalte lediglich für eine sehr grobe Orientierung bzw. Sensibilisierung gegenüber einem potenziellen Problem dienen.

## 6. Zusammenfassung

In der Literatur und in der Praxis wird – zu Recht – meist eine XML-basierte Datenaustauschnittstelle empfohlen. Die Vorteile liegen auf der Hand:

- XML ist standardisiert
- wird von zahlreichen Werkzeugen unterstützt
- kann validiert werden (wenn z. B. die Schnittstelle mit XML-Schema definiert ist)
- ist für Menschen und Maschinen lesbar

Im Umweltbereich gibt es bis jetzt kaum wiederverwendbare (Teil-)Schemata. Die Schaffung solcher ist aber mit der Berücksichtigung einiger in dieser Studie beschriebenen Regeln möglich und wurde beim Entwurf der Datenaustauschnittstelle zwischen der EU3-Fachanwendung als Datenquelle und der LDB als Datenziel auch erreicht.

Die wichtigsten behandelten Anforderungen für die Definition einer Datenaustauschnittstelle auf XML-Basis sind:

- Konventionen einführen
- gemeinsame Schemateile definieren
- gemeinsame Schlüssellisten verwenden
- die Schnittstelle mit genauen Datentyp- und Referenzangaben restriktiv definieren
- die Dateigrößen bei umfangreichen Daten beachten
- bei inkrementeller Datenübernahme sind entsprechende Konstrukte notwendig

- [1] Extensible Markup Language (XML): <http://www.w3.org/XML/>
- [2] XML Schema: <http://www.w3.org/XML/Schema>
- [3] <http://www.xml.org/>
- [4] <http://www.oasis-open.org>
- [5] <http://schlemmer.wiwi.hu-berlin.de/xml-eml.nsf>
- [6] [http://www.bum.iao.fhg.de/FG\\_BUIS/folien/FolienIsenmann.pdf](http://www.bum.iao.fhg.de/FG_BUIS/folien/FolienIsenmann.pdf)
- [7] H.W. Koepfel, R. Mayer-Föll, M. Haase (Herausgeber): Nutzung von XML/EML für den Datenaustausch im Wasserbereich und dem Bereich des Landschafts- und Naturschutzes. Dokumentation des Workshops am Forschungsinstitut für anwendungsorientierte Wissensverarbeitung an der Universität Ulm am 16.01.2003 in Ulm.
- [8] Open GIS Consortium: GML - the Geography Markup Language. <http://www.opengis.net/gml/>