

# eCH-0118 GML-Kodierungsregeln für INTERLIS

<b>Name</b>	GML-Kodierungsregeln für INTERLIS
<b>eCH-Nummer</b>	eCH-0118
<b>Kategorie</b>	Standard
<b>Reifegrad</b>	Implementiert
<b>Version</b>	2.0
<b>Status</b>	Genehmigt
<b>Genehmigt am</b>	2016-09-07
<b>Ausgabedatum</b>	2016-09-09
<b>Ersetzt Version</b>	1.00
<b>Voraussetzungen</b>	eCH-0118 1.00
<b>Beilagen</b>	ili2.xsd, ili2c.xsd
<b>Sprachen</b>	Deutsch (Original), Französisch (Übersetzung)
<b>Autoren</b>	eCH-Fachgruppe INTERLIS <i>Claude Eisenhut, Eisenhut Informatik AG</i> <i>Michael Germann, infoGrips GmbH</i> <i>Rolf Zürcher, swisstopo</i>  <i>Pasquale Di Donato, swisstopo</i> <i>Olivier Ertz, HEIG-VD</i> <i>Jens Ingensand, HEIG-VD</i> <i>Dominic Kottmann, KK GEO</i> <i>Thorsten Reitz, wetransform GmbH</i>
<b>Herausgeber / Vertrieb</b>	Verein eCH, Mainaustrasse 30, Postfach, 8034 Zürich T 044 388 74 64, F 044 388 71 80 <a href="http://www.ech.ch">www.ech.ch</a> / <a href="mailto:info@ech.ch">info@ech.ch</a>

## Zusammenfassung

Dieser eCH-Standard definiert Kodierungsregeln, um aus einem INTERLIS-Datenmodell ein GML-Transferformat abzuleiten. INTERLIS ist eine Schweizer Norm zur Beschreibung von konzeptionellen Datenmodellen und zum modellbasierten Datenaustausch. GML ist ein Standard zur Kodierung von geografischen Daten, basierend auf XML.

Zunächst werden grundlegende Konzepte der verwendeten Sprachen INTERLIS und GML sowie Kodierungsregeln und XML-Anwendungsfälle erläutert (Kapitel 4). Danach folgt eine Darstellung der Unterschiede zwischen der GML-Kodierung und der Kodierung von INTERLIS-spezifischem XML (Kapitel 5).

Der Hauptteil dieses Standards beschäftigt sich mit der detaillierten, umfassenden Definition von Schema-Kodierungsregeln (Kapitel 6) sowie Instanz-Kodierungsregeln (Kapitel 7).

Im Anhang A ist das vordefinierte Basis-XML-Schema zur GML-Kodierung von INTERLIS eingefügt. Im Anhang B ist das ili2c-Schema eingefügt. Beide Dokumente sind neben dem vorliegenden Standard separat als XML-Schemadateien verfügbar.

Die vorliegende Spezifikation wird durch ein einfaches Anwendungsbeispiel abgeschlossen (Anhang C).

## Inhaltsverzeichnis

<b>1</b>	<b>Status des Dokuments</b> .....	<b>6</b>
<b>2</b>	<b>Einleitung</b> .....	<b>7</b>
2.1	Überblick .....	7
2.2	Voraussetzungen.....	7
2.3	Allgemeine Ziele und Einschränkungen .....	7
<b>3</b>	<b>Normative Referenzen</b> .....	<b>8</b>
<b>4</b>	<b>Begriffe, Konzepte</b> .....	<b>9</b>
4.1	INTERLIS .....	9
4.2	Geography Markup Language (GML) .....	9
4.3	Kodierungsregeln .....	10
<b>5</b>	<b>Unterschiede zu den INTERLIS-XML-Kodierungsregeln</b> .....	<b>11</b>
5.1	Kodierungsregeln .....	11
5.2	Aufbau eines Transfers .....	11
5.3	Lesen von erweiterten/übersetzten Modellen.....	11
5.4	Zeichenvorrat.....	11
5.5	Transferarten.....	12
<b>6</b>	<b>Schema-Kodierungsregeln</b> .....	<b>13</b>
6.1	Vorbemerkung.....	13
6.2	Zeichenkodierung .....	13
6.3	Allgemeine Regeln.....	13
6.4	Abbildung der INTERLIS-Modellelemente auf XML-Schema-Namen .....	13
6.5	Abbildung von Basistypen.....	14
6.6	Allgemeiner Aufbau des XML-Schemas .....	14
6.7	Kodierung von Wertebereichen .....	15
6.8	Kodierung von Themen .....	16
6.9	Kodierung von Klassen.....	18
6.10	Kodierung von Strukturen .....	19
6.11	Kodierung von Beziehungen.....	19
6.11.1	Eingebettete Referenzen.....	20
6.11.2	Zwischentabelle.....	20

6.11.3 Beispiele.....	22
<b>6.11.3.1 Nur eingebettete Referenzen, keine Zwischentabelle .....</b>	<b>22</b>
<b>6.11.3.2 Eingebettete Referenzen und Zwischentabelle .....</b>	<b>23</b>
<b>6.11.3.3 Keine eingebettete Referenzen, nur Zwischentabelle.....</b>	<b>24</b>
<b>6.11.3.4 Teilweise eingebettete Referenzen und Zwischentabelle.....</b>	<b>25</b>
6.12 Kodierung von Sichten.....	27
6.13 Kodierung von Grafikdefinitionen.....	27
6.14 Kodierung von Attributen .....	27
6.15 Abbildung der INTERLIS-Typen auf XML-Schema-Typen .....	29
6.15.1 Übersicht.....	29
6.15.2 Kodierung von Zeichenketten.....	30
6.15.3 Kodierung von Aufzählungen.....	31
6.15.4 Kodierung von numerischen Datentypen.....	35
6.15.5 Kodierung von formatierten Wertebereichen .....	35
6.15.6 Kodierung von Gefässen.....	36
6.15.7 Kodierung von Klassentypen.....	36
6.15.8 Kodierung von Attributpfadtypen .....	37
6.15.9 Kodierung von Koordinaten .....	37
6.15.10 Kodierung von Linienzügen.....	38
6.15.11 Kodierung von Einzelflächen.....	39
6.15.12 Kodierung von Gebietseinteilungen .....	40
6.15.13 Kodierung von LINE ATTRIBUTES.....	40
6.15.14 Kodierung von Multigeometrien.....	41
6.15.15 Kodierung unterschiedlicher Geometrieattribute innerhalb derselben Klasse .....	42
6.15.16 Kodierung von Typen zur Objektidentifikation .....	43
6.15.17 Kodierung von weiteren Kurvenstück-Formen.....	44
<b>7 Instanz-Kodierungsregeln .....</b>	<b>45</b>
7.1 Kodierung mehrerer Behälter (TOPICs) in einer Transferdatei .....	45
7.2 Kodierung von Objekten und Strukturelementen .....	45
7.3 Kodierung von Objekt- und Behälteridentifikationen .....	45
7.4 Kodierung von Beziehungen.....	46
7.5 Kodierung von Geometrien.....	46

---

7.6	Kodierung von Code Lists.....	46
<b>8</b>	<b>Haftungsausschluss/Hinweise auf Rechte Dritter .....</b>	<b>47</b>
<b>9</b>	<b>Urheberrechte.....</b>	<b>47</b>
	<b>Anhang A – Vordefiniertes Basis-Schema .....</b>	<b>48</b>
	<b>Anhang B – ili2c-Basis-Schema .....</b>	<b>50</b>
	<b>Anhang C – Beispiel.....</b>	<b>51</b>
	INTERLIS 2-Modell.....	51
	INTERLIS 1-Modell.....	52
	GML-Applikationsschema.....	53
	INTERLIS 1-Transferdaten.....	55
	INTERLIS 2-Transferdaten («INTERLIS-XML»).....	56
	GML-Daten.....	58
	<b>Anhang D – Mitarbeit &amp; Überprüfung .....</b>	<b>61</b>
	<b>Anhang E – Abkürzungen und Glossar .....</b>	<b>61</b>
	<b>Anhang F – Änderungen gegenüber Vorversion .....</b>	<b>62</b>

# 1 Status des Dokuments

**Genehmigt:** Das Dokument wurde vom Expertenausschuss genehmigt. Es hat für das definierte Einsatzgebiet im festgelegten Gültigkeitsbereich normative Kraft.

## 2 Einleitung

### 2.1 Überblick

In diesem Dokument werden die Regeln definiert, um aus einem INTERLIS-Modell ein GML-Applikationsschema abzuleiten.

### 2.2 Voraussetzungen

Dieses Dokument setzt INTERLIS- und GML-Kenntnisse voraus. Für eine vertiefte Auseinandersetzung sei auf die Quellen [1,2], [3] verwiesen. Der Text ist aber mit Beispielen ergänzt, so dass dem Inhalt auch von Nicht-Experten gefolgt werden kann.

### 2.3 Allgemeine Ziele und Einschränkungen

- Es wird eine möglichst genaue Modellabbildung angestrebt
- Es entsteht kein Datenverlust gegenüber einem Datentransfer mit INTERLIS 1 oder 2 (ITF oder XTF)
- Das resultierende GML-Applikationsschema lässt sich durch GML-konforme Software nutzen
- Das resultierende GML-Applikationsschema enthält nicht alle Beschreibungsmerkmale des INTERLIS-Modells, weil GML diese Möglichkeiten nicht kennt. Auf der Stufe Modell (Datenbeschreibung) gibt es also einen Verlust, d.h. das GML-Applikationsschema ist kein Ersatz für das INTERLIS-Modell.
- Das resultierende GML-Applikationsschema eignet sich für Geowebdienste (insbesondere WFS), ist aber weniger geeignet für den Massendatentransfer (wegen der Art, wie die Beziehungen kodiert werden).
- Dieses Dokument verwendet INTERLIS 2.3, INTERLIS 2.4 und GML 3.2.1. Da INTERLIS 1 mehr oder weniger 1:1 nach INTERLIS 2.3 übersetzt werden kann, gilt es aber entsprechend auch für INTERLIS 1.

*Anmerkung:* Es entsteht kein Datenverlust, aber Fähigkeiten von XTF (INTERLIS 2-XML-Transferformat), die in GML nicht vorhanden sind (z.B. inkrementeller Transfer), werden nicht nachgebildet. Das resultierende Transferformat ist in diesem Fall kein vollständiger Ersatz für XTF.

Die GML-Basistypen werden nach Möglichkeit nicht erweitert, um eine möglichst breite Nutzung der Daten zu ermöglichen.

Das resultierende GML-Applikationsschema ist nach Möglichkeit GML Simple Feature (GML-SF) [8] konform, um eine möglichst breite Nutzung der Daten zu ermöglichen. Wenn das INTERLIS-Modell Elemente enthält die sich nicht 1:1 nach GML-SF abbilden lassen, ist das resultierende GML-Applikationsschema nicht GML-SF-konform.

### 3 Normative Referenzen

- [1] eCH, 2016. *eCH-0031 Geoinformation: INTERLIS 2- Referenzhandbuch Version 2.0*,  
<http://www.ech.ch> → eCH Dokumente → Standards → eCH-0031
- [2] eCH, 2006. *CH-0031 Geoinformation: INTERLIS 2- Referenzhandbuch Version 1.0*,  
<http://www.ech.ch> → eCH Dokumente → Standards → eCH-0031
- [3] Open Geospatial Consortium (OGC), 2007. *OpenGIS Geography Markup Language (GML) Encoding Standard*. OGC 07-036, Version 3.2.1  
[http://portal.opengeospatial.org/files/?artifact\\_id=20509](http://portal.opengeospatial.org/files/?artifact_id=20509)  
Online: 2016-01-25
- [4] World Wide Web Consortium (W3C), 2004. *XML Schema 1.1*. Standard  
<http://www.w3.org/XML/Schema>  
Online: 2016-01-25
- [5] ISO/IEC 14977, 1996. *Information technology – Syntactic metalanguage – Extended BNF*. International standard.
- [6] ISO/IEC 10646:2003, 2003. *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*. International standard.
- [7] eCH, 2010. *eCH-0117 Meta-Attribute für INTERLIS Modelle Version 1.0*,  
<http://www.ech.ch> → eCH Dokumente → Standards → eCH-0117
- [8] Open Geospatial Consortium (OGC), 2007. *Geography Markup Language (GML) simple features profile (with Corrigendum)*. OGC 10-100r3, Version 2.0  
[http://portal.opengeospatial.org/files/?artifact\\_id=42729](http://portal.opengeospatial.org/files/?artifact_id=42729)  
Online: 2016-01-25
- [9] Open Geospatial Consortium (OGC), 2012. *OGC Geography Markup Language (GML) – Extended schemas and encoding rules*. OGC 10-129r1, Version 3.3.0  
[https://portal.opengeospatial.org/files/?artifact\\_id=46568](https://portal.opengeospatial.org/files/?artifact_id=46568)  
Online: 2016-01-25



## 4 Begriffe, Konzepte

### 4.1 INTERLIS

INTERLIS ist eine konzeptionelle Datenbeschreibungssprache und wird im Bereich der Geodatenmodellierung und des Geodatenaustausches verwendet [1,2].

Das INTERLIS-Referenzhandbuch [1,2] besteht aus zwei Teilen:

- eine Datenbeschreibungssprache, und
- XML-Kodierungsregeln.

Werden die XML-Kodierungsregeln auf ein mit Hilfe der Datenbeschreibungssprache definiertes Datenmodell angewendet, entsteht ein zum Datenmodell passendes Transferformat.

Für die XML-Kodierungsregeln wird im vorliegenden Dokument eine Alternative definiert.

Im Anhang C ist ein Beispiel-Datenmodell in INTERLIS inkl. einer Transferdatei gemäss diesem Datenmodell eingefügt.

### 4.2 Geography Markup Language (GML)

GML [3] ist ein Satz von Basis-XML-Schemas [4] und ein Satz von Regeln, wie diese Schemas in eigenen Schemas (sog. *Applikationsschemas*) für die Definition von Transferformaten zu verwenden sind. Aus der Kenntnis der Basis-Schemas und der Regeln für Applikationsschemas lässt sich somit auch eine Datenbeschreibung herleiten.

*Anmerkung:* Weil die Regeln für GML-Applikationsschemas definiert sind, kennt man nicht nur die zulässige Folge von XML-Elementen (aufgrund des XML-Schemas), sondern weiss, welche XML-Elemente Klassen darstellen und welche Elemente Attribute dieser Klassen sind. Diese Beschreibung ist weniger genau als mittels INTERLIS, darum geht es aber in diesem Dokument nicht primär.

Die INTERLIS 1-FMT-Datei ist auch eine Formatbeschreibung. Im Gegensatz zu einem XML-Schema handelt es sich dabei aber um keine formale Beschreibung.

Im Anhang C ist ein Beispiel-Applikationsschema inkl. einer Transferdatei gemäss diesem Schema eingefügt.

*Anmerkung:* Die GML 3.1-Schemas validieren teilweise nicht! Das ist nicht nur ein Problem von GML; siehe dazu auch <http://schemas.opengis.net/gml/3.1.1/readme.txt>

In der GML 3.2-Spezifikation [3] steht dazu:

«21.2.6 Property Type Derivation [...]

*NOTE As derivation-by-restriction of property types has created problems with commonly used XML parsers in the past, all instances of such derivations have been removed from the GML schema. It is recommended to avoid derivation by restriction in property types in application schemas, to.»*

### 4.3 Kodierungsregeln

Kodierungsregeln definieren, wie aus einem fachlichen Datenmodell durch die Anwendung von Regeln das Transferformat abgeleitet werden kann (modellbasierter Datentransfer). Die Regeln sind dabei so formuliert, dass sie auf ein beliebiges Datenmodell angewendet werden können.

## 5 Unterschiede zu den INTERLIS-XML-Kodierungsregeln

### 5.1 Kodierungsregeln

Das INTERLIS-Referenzhandbuch definiert Instanz-Kodierungsregeln. Im vorliegenden Dokument werden Schema-Kodierungsregeln definiert und die Instanzen ergeben sich aus dem so erzeugten Schema.

### 5.2 Aufbau eines Transfers

Das INTERLIS-Referenzhandbuch unterteilt einen Transfer in Vorspann und Datenbereich. Auf den Vorspann wird in diesem Dokument verzichtet, da dies durch eine GML-konforme Software als Daten interpretiert würde (statt als Metadaten zum Transfer).

### 5.3 Lesen von erweiterten/übersetzten Modellen

Das INTERLIS 2.3 Referenzhandbuch [2] definiert als Teil des Vorspanns eine Abbildungstabelle, so dass ein Leseprogramm das für ein bestimmtes Datenmodell programmiert oder konfiguriert worden ist, Daten von Erweiterungen (oder Übersetzungen) dieses Datenmodells ohne Kenntnis der erweiterten Modelldefinitionen lesen kann. In diesem Dokument wird auf eine solche Abbildungstabelle verzichtet.

Ein Leseprogramm muss also das erweiterte Datenmodell kennen, um Erweiterungen korrekt ignorieren zu können.

Für INTERLIS 2.4 [1] sind keine speziellen Regelungen notwendig, da in INTERLIS 2.4 die Abbildungstabelle im Vorspann abgeschafft wurde.

Für übersetzte INTERLIS-Datenmodelle (TRANSLATION OF) wird kein GML-Applikationsschema erzeugt, d.h. das Transferformat ändert sich bei einer Übersetzung nicht.

### 5.4 Zeichenvorrat

Das INTERLIS-Referenzhandbuch definiert einen eingeschränkten Zeichenvorrat (Anhang D im Referenzhandbuch) [1].

Das vorliegende Dokument definiert standardmässig den Unicode-Zeichenumfang. Ein eingeschränkter Zeichenvorrat muss im Rahmen einer Transfergemeinschaft definiert werden.

## 5.5 Transferarten

Das INTERLIS-Referenzhandbuch definiert drei Transferarten: «FULL», «INITIAL» oder «UPDATE». Zusätzlich können Angaben zur Konsistenz übermittelt werden: «COMPLETE», «INCOMPLETE», «INCONSISTENT», «ADAPTED».

Die in diesem Dokument definierten Kodierungsregeln gelten für den Transfer von einzelnen Objekten oder von ganzen Datensätzen.

*Anmerkung:* Inkrementeller Transfer lässt sich nur mit GML-Standardsoftware nicht realisieren, da GML keine solche Semantik definiert. Konsistenzangaben könnte man als GML-Metadaten codieren. In diesem Dokument wird auf den inkrementellen Transfer verzichtet.

## 6 Schema-Kodierungsregeln

### 6.1 Vorbemerkung

Für die Formalisierung der Transferformat-Ableitungsregeln wird die im INTERLIS-Referenzhandbuch [1] in Kapitel 2.1 eingeführte EBNF-Notation benutzt [5].

### 6.2 Zeichenkodierung

Es gelten die XML-Regeln: Unicode-Zeichensatz und UTF-8 (empfohlen) oder UTF-16 als Zeichencodierung [6].

### 6.3 Allgemeine Regeln

Das generierte XML-Schema [4] kann beliebig viele Kommentare (<!-- ... -->) oder zusätzliche `xsd:annotation`-Elemente enthalten.

### 6.4 Abbildung der INTERLIS-Modellelemente auf XML-Schema-Namen

Jedes INTERLIS-Datenmodell enthält einen eigenen XML-Schema-Namensraum. Für die XML-Element- und -Typ-Definitionen werden die unqualifizierten INTERLIS-Namen verwendet. Bei Konflikten wird der qualifizierte Namen verwendet. Die Namen auf Stufe MODEL haben Vorrang gegenüber Namen aus der Stufe TOPIC. Bei Konflikten zwischen Namen aus unterschiedlichen TOPICs hat der Name aus dem im Modell zuerst definierten TOPIC Vorrang.

Beispiel INTERLIS	GML Element-Namen
<pre> INTERLIS 2.4;  MODEL ModelA (de) AT "mailto:ceis@localhost" VERSION "2016-01-31" =   TOPIC TopicA =     CLASS ClassA =       END ClassA;     CLASS TopicA = !! qualifizierter Name                   !! wegen Konflikt       END TopicA;   END TopicA;  END ModelA. </pre>	<pre> TopicA ClassA TopicA.TopicA </pre>

## 6.5 Abbildung von Basistypen

Für modellunabhängige Basistypen aus INTERLIS wird ein Basisschema (Anhang A) definiert.

## 6.6 Allgemeiner Aufbau des XML-Schemas

Jedes INTERLIS-Modell wird in ein XML-Schema in einem Schemadokument abgebildet.

```

ModelDef = '<xsd:schema
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns="%ModelNamespaceIdentifier%"
    targetNamespace="%ModelNamespaceIdentifier%"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    xmlns:gml="http://www.opengis.net/gml/3.2"
    { 'xmlns:%nsprefix% namespace=
      "%Imported-ModelNamespaceIdentifier%" ' }
    '>'
  IliModelInfo
  '<xsd:import
    namespace="http://www.opengis.net/gml/3.2"/>'
  { ModelImport }
  { ClassDef | AssociationDef | DomainDef }
  { TopicDef }
  '</xsd:schema>'.

IliModelInfo = '<xsd:annotation>
  <xsd:appinfo source="http://www.interlis.ch/ili2c">
  <ili2:model>%ModelName%</ili2:model>
  <ili2:modelVersion>%ModelVersion%
  </ili2:modelVersion>
  <ili2:modelAt>%ModelAt%</ili2:modelAt>
  </xsd:appinfo>
  </xsd:annotation>'.

ModelImport = '<xsd:import
  namespace="%Imported-ModelNamespaceIdentifier%"/>'.

ModelNamespaceIdentifier = StdModelNamespaceIdentifier
  | %MetaAttributeValue%.

StdModelNamespaceIdentifier =
  'http://www.interlis.ch/ILIGML-2.0/'%ModelName%.

```

Das ili2c-Modell ist im Anhang B definiert.

Der XML-Namensraumidentifikator (Regel «ModelNamespaceIdentifier») ergibt sich aus dem Modellnamen oder kann über das Metaattribut «ili2.iligml20.namespaceName» definiert werden [7].

Beispiel	
INTERLIS	<pre> INTERLIS 2.4;  MODEL ModelA (de) AT "mailto:ceis@localhost" VERSION "2016-01-31" =   IMPORTS Units;  END ModelA. </pre>
GML	<pre> &lt;xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"   xmlns="http://www.interlis.ch/ILIGML-2.0/ModelA"   targetNamespace=     "http://www.interlis.ch/ILIGML-2.0/ModelA"   elementFormDefault="qualified"   attributeFormDefault="unqualified"   xmlns:gml="http://www.opengis.net/gml/3.2"   xmlns:ns1=     "http://www.interlis.ch/ILIGML-2.0/Units"&gt;   &lt;xsd:annotation&gt;     &lt;xsd:appinfo source="http://www.interlis.ch/ili2c"&gt;       &lt;ili2:model&gt;ModelA&lt;/ili2:model&gt;       &lt;ili2:modelVersion&gt;2016-01-31&lt;/ili2:modelVersion&gt;       &lt;ili2:modelAt&gt;mailto:ceis@localhost&lt;/ili2:modelAt&gt;     &lt;/xsd:appinfo&gt;   &lt;/xsd:annotation&gt;   &lt;xsd:import namespace="http://www.opengis.net/gml/3.2"/&gt;   &lt;xsd:import namespace     ="http://www.interlis.ch/ILIGML-2.0/Units"/&gt; &lt;/xsd:schema&gt; </pre>

## 6.7 Kodierung von Wertebereichen

Wertebereichsdefinitionen werden wie folgt abgebildet:

```

DomainDef = DomainDefSimple | DomainDefComplex.

DomainDef
Simple = '<xsd:simpleType name="%DomainName%">
  <xsd:restriction base="%BaseType%">
    <!-- any applicable xsd restrictions -->
  </xsd:restriction>
  </xsd:simpleType>'.

DomainDefComplex = '<xsd:complexType name="%DomainName%">
  <xsd:complexContent>
  </xsd:complexContent>
  </xsd:complexType>'.

```

Je nach Abbildungsregel des entsprechenden INTERLIS-Datentyps (siehe Kapitel 6.15) wird die Regel «DomainDefComplex» oder «DomainDefSimple» angewendet.

Beispiel	
INTERLIS	DOMAIN  Horizontbezeichnung = TEXT*20; BlackboxXml = BLACKBOX XML;
GML	<pre> &lt;xsd:simpleType name="Horizontbezeichnung"&gt;   &lt;xsd:restriction base="xsd:normalizedString"&gt;     &lt;xsd:maxLength value="20"/&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt; &lt;xsd:complexType name="BlackboxXml"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:any namespace="##any" minOccurs="0" max- Occurs="unbounded" processContents="lax"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt; </pre>

Geometrische Wertebereiche werden nicht kodiert. Einschränkungen geometrischer Typen lassen sich im GML-Applikationsschema tatsächlich nicht festhalten.

## 6.8 Kodierung von Themen

Themen haben in INTERLIS zwei Funktionen: einerseits definieren sie einen eigenen Namensraum (getrennt vom Namensraum des Modells), andererseits definieren sie einen Behälter. Die Eigenschaft als Namensraum wird nicht abgebildet. Die Eigenschaft als Behälterbeschreibung wird wie folgt abgebildet.

```

TopicDef = '<xsd:complexType name="%TopicName%MemberType">
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractFeatureMemberType">
      <xsd:sequence>
        <xsd:choice>'
          { '<xsd:element ref="%ClassName%"/>' }
        '</xsd:choice>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="%TopicName%" type="%TopicName%Type"
  substitutionGroup="gml:AbstractFeature"/>
<xsd:complexType name="%TopicName%Type">
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractFeatureType">
      <xsd:sequence>
        <xsd:element name="member"
          type="%TopicName%MemberType"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attributeGroup
        ref="gml:AggregationAttributeGroup"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>'.

```



Beispiel	
<b>INTERLIS</b>	<pre> TOPIC Bodenbedeckung =    CLASS BoFlaechen =   END BoFlaechen;    CLASS Gebaeude =   END Gebaeude;  END Bodenbedeckung; </pre>
<b>GML</b>	<pre> &lt;xsd:complexType name="BodenbedeckungMemberType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureMemberType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:choice&gt;           &lt;xsd:element ref="BoFlaechen"/&gt;           &lt;xsd:element ref="Gebaeude"/&gt;         &lt;/xsd:choice&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; &lt;xsd:element name="Bodenbedeckung"   type="BodenbedeckungType"   substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="BodenbedeckungType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="member"           type="BodenbedeckungMemberType"           minOccurs="0" maxOccurs="unbounded"/&gt;       &lt;/xsd:sequence&gt;       &lt;xsd:attributeGroup         ref="gml:AggregationAttributeGroup"/&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; </pre>
<b>GML-Instanz</b>	<pre> &lt;Bodenbedeckung gml:id="b1"&gt;   &lt;member&gt;     &lt;BoFlaechen gml:id="o1"&gt;       ...     &lt;/BoFlaechen&gt;   &lt;/member&gt;   &lt;member&gt;     &lt;BoFlaechen gml:id="o2"&gt;       ...     &lt;/BoFlaechen&gt;   &lt;/member&gt; &lt;/Bodenbedeckung&gt; </pre>

## 6.9 Kodierung von Klassen

Klassen werden wie folgt abgebildet:

```

ClassDef = '<xsd:element name="%ClassName%"
              type="%ClassName%Type"
              substitutionGroup="gml:AbstractFeature
              | %Base-ClassName%"/>
<xsd:complexType name="%ClassName%Type">
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractFeatureType
    | %Base-ClassName%Type">
      <xsd:sequence>
        { AttributeDef | EmbeddedRoleDef }
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>'

```

Beispiel	
INTERLIS	<pre> CLASS Horizont =   Horizontbezeichnung : MANDATORY TEXT*10; END Horizont; </pre>
GML	<pre> &lt;xsd:element name="Horizont" type="HorizontType"               substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="HorizontType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="Horizontbezeichnung"&gt;           &lt;xsd:simpleType&gt;             &lt;xsd:restriction base="xsd:normalizedString"&gt;               &lt;xsd:maxLength value="10"/&gt;             &lt;/xsd:restriction&gt;           &lt;/xsd:simpleType&gt;         &lt;/xsd:element&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; </pre>
GML-Instanz	<pre> &lt;Horizont gml:id="o1"&gt;   &lt;Horizontbezeichnung&gt;Ah&lt;/Horizontbezeichnung&gt; &lt;/Horizont&gt; </pre>

## 6.10 Kodierung von Strukturen

Strukturen werden wie folgt abgebildet:

```

ClassDef = '<xsd:element name="%StructureName%"
              type="%ClassName%Type"
              substitutionGroup="gml:AbstractObject
              | %Base-ClassName%"/>
              <xsd:complexType name="%StructureName%Type">
                <xsd:sequence>
                  { AttributeDef | EmbeddedRoleDef }
                </xsd:sequence>
              </xsd:complexType>'
    
```

### Exemple

<p><b>INTERLIS</b></p>	<pre> DOMAIN ... STRUCTURE Koernung =   Ton : Prozent100;   Schluff : Prozent100;   Sand : Prozent100; END Koernung;         </pre>
<p><b>GML</b></p>	<pre> &lt;xsd:element name="Koernung" type="KoernungType"               substitutionGroup="gml:AbstractObject"/&gt; &lt;xsd:complexType name="KoernungType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="Ton" type="Prozent100"                   minOccurs="0"/&gt;     &lt;xsd:element name="Schluff" type="Prozent100"                   minOccurs="0"/&gt;     &lt;xsd:element name="Sand" type="Prozent100"                   minOccurs="0"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt;         </pre>

## 6.11 Kodierung von Beziehungen

Für die Kodierung von Beziehungen gibt es zwei Methoden:

- Die Kodierung mittels eingebetteter Referenzen
- Die Kodierung mittels Zwischentabellen

Je nach Definition der ASSOCIATION im Modell wird nur eine von diesen Methoden, oder eine Kombination beider Methoden, verwendet.

Folgende Elemente einer Beziehung beeinflussen einzeln oder in Kombination die Wahl der Methode:

- EXTERNAL: Externe Referenzen (EXTERNAL bei RoleDef)
- ATTRIBUTE: sind Attribute vorhanden oder nicht
- FINAL: Ist die Beziehung als FINAL definiert, oder nicht

- ORDERED: geordnete Beziehung
- OID: Beziehungsinstanz hat eine Objektidentifikation

### 6.11.1 Eingebettete Referenzen

Die Kodierung mittels einer eingebetteten Referenz erfolgt für Rollen (RoleDef) ohne die Eigenschaft EXTERNAL. Ist die gegenüberliegende Rolle geordnet (ORDERED), ist die Reihenfolge der Referenzen in der Instanz signifikant (das ist im XSD nicht ersichtlich).

```
EmbeddedRoleDef =
  '<xsd:element name="%RoleName%" type="gml:ReferenceType"
    [ 'minOccurs="%minCardinality%" ]
    [ 'maxOccurs="%maxCardinality%" ] '>'
  '<xsd:annotation>
    <xsd:appinfo>
      <gml:targetElement>%ClassName%</gml:targetElement>
    </xsd:appinfo>
  </xsd:annotation>
</xsd:element>'
```

%RoleName% ergibt sich aus dem Namen der gegenüberliegenden Rolle. %ClassName% ist die Zielklasse der gegenüberliegenden Rolle.

%minCardinality% und %maxCardinality% ergeben sich aus der Kardinalität der gegenüberliegenden Rolle.

Die Kodierung mittels einer eingebetteten Referenz erfolgt auch dann, wenn es zusätzlich eine Zwischentabelle gibt.

### 6.11.2 Zwischentabelle

Die Kodierung mittels Zwischentabelle erfolgt, falls mindestens eine der folgenden Bedingungen erfüllt ist:

- Die Assoziation hat Attribute
- Die Assoziation ist nicht FINAL
- Die Assoziation ist eine Erweiterung (EXTENDED oder EXTENDS)
- Die Beziehungsinstanzen haben eine Objektidentifikation
- Alle Rollen sind EXTERNAL
- Eine Rolle ist EXTERNAL und die gegenüberliegende Rolle ist ORDERED
- Die Beziehung hat mehr als zwei Rollen

**Sonderregel:** Ist die Beziehung nicht FINAL, keine Erweiterung, hat keine Attribute, hat nur zwei Rollen, alle Rollen sind nicht EXTERNAL und es ist keine Objektidentifikation erforderlich, werden zwar im XSD die Definitionen für eine Zwischentabelle erzeugt (als Kopf für eine evtl. in einem anderen Modell vorhandene Erweiterung), aber in der XML-Instanz wird kein Element für das Zwischenobjekt kodiert.

```
AssociationDef = '<xsd:element name="%AssociationName%"
  type="%AssociationName%Type"
  substitutionGroup="gml:AbstractFeature |
  | %Base-AssociationName%"/>'
```

```
<xsd:complexType name="%AssociationName%Type">
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractFeatureType |
      | %Base-AssociationName%Type">
      <xsd:sequence>
        { RoleDef } { AttributeDef | EmbeddedRoleDef }
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>'.
```

Für die Reihenfolge der Attribute, Rollen, Eingebetteten-Rolle innerhalb der Assoziation gilt: Zuerst werden alle Rollen, dann alle Attribute und dann alle Eingebetteten-Rollen kodiert. Die Attribute und Rollen werden gemäss ihrer Definitionsreihenfolge in der Modelldatei kodiert. Die Eingebetteten-Rollen werden alphabetisch aufsteigend sortiert. Ein spezialisiertes Attribut oder eine spezialisierte Rolle (mit dem Schlüsselwort «EXTENDED»), wird nicht generiert.

Ist die Assoziation keine Erweiterung einer anderen Assoziation, wird `gml:AbstractFeatureType` als Basistyp, bzw. `gml:AbstractFeature` als Basis-Element verwendet. Ist die Assoziation eine Erweiterung, wird der entsprechend generierte Basistyp bzw. das Basis-Element verwendet. Die Vererbungsstruktur der Assoziation ist somit im XML-Schema wiedergegeben.

Hinweis: Für Beziehungen ohne expliziten Namen wird der (Klassen)Name durch zusammenhängen der einzelnen Rollennamen gebildet (z.B. `%RoleName1RoleName2%`).

```
RoleDef = UnorderedRoleDef | OrderedRoleDef.

UnorderedRoleDef =
  '<xsd:element name="%RoleName%" type="gml:ReferenceType">
    <xsd:annotation>
      <xsd:appinfo>
        <gml:targetElement>%ClassName%</gml:targetElement>
      </xsd:appinfo>
    </xsd:annotation>
  </xsd:element>'.
```

```
OrderedRoleDef =
  '<xsd:element name="%RoleName%">
    <xsd:annotation>
      <xsd:appinfo>
        <gml:targetElement>%ClassName%</gml:targetElement>
      </xsd:appinfo>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence/>
      <xsd:attributeGroup ref="gml:OwnershipAttributeGroup"/>
      <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      <xsd:attribute ref="ili:ORDER POS"/>
    </xsd:complexType>
  </xsd:element>'.
```

Ist die Beziehung geordnet, ist gegenüber `gml:ReferenceType` ein zusätzliches XML-Attribut («ORDER\_POS»; in Anhang A definiert) erforderlich.

Ob die Referenz auf ein Objekt in der gleichen Transferdatei zeigt, lässt sich nicht immer erkennen (nur wenn die Referenz ein Dokument relativer XML-Fragment-Identifikator ist).

## 6.11.3 Beispiele

### 6.11.3.1 Nur eingebettete Referenzen, keine Zwischentabelle

<p>INTERLIS</p>	<pre> CLASS ClassA = END ClassA;  CLASS ClassB = END ClassB;  ASSOCIATION a2b (FINAL)=   a -- {1..*} ClassA;   b -- {1..*} ClassB; END a2b;         </pre>
<p>GML</p>	<pre> &lt;xsd:element name="ClassA" type="ClassAType"   substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="ClassAType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="b" maxOccurs="unbounded"           type="gml:ReferenceType"&gt;           &lt;xsd:annotation&gt;             &lt;xsd:appinfo&gt;               &lt;gml:targetElement&gt;ClassB&lt;/gml:targetElement&gt;             &lt;/xsd:appinfo&gt;           &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; &lt;xsd:element name="ClassB" type="ClassBType"   substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="ClassBType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="a" maxOccurs="unbounded"           type="gml:ReferenceType"&gt;           &lt;xsd:annotation&gt;             &lt;xsd:appinfo&gt;               &lt;gml:targetElement&gt;ClassA&lt;/gml:targetElement&gt;             &lt;/xsd:appinfo&gt;           &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt;         </pre>
<p>GML-Instanz</p>	<pre> &lt;ClassA gml:id="o1"&gt;   &lt;b xlink:href="#o2"/&gt; &lt;/ClassA&gt; &lt;ClassB gml:id="o2"&gt;   &lt;a xlink:href="#o1"/&gt; &lt;/ClassB&gt;         </pre>

### 6.11.3.2 Eingebettete Referenzen und Zwischentabelle

<p>INTERLIS</p>	<pre> CLASS ClassA = END ClassA;  CLASS ClassB = END ClassB;  ASSOCIATION a2b (OID)=   a -- {1..*} ClassA;   b -- {1..*} ClassB; END a2b; </pre>
<p>GML</p>	<pre> &lt;xsd:element name="ClassA" type="ClassAType"   substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="ClassAType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="b" maxOccurs="unbounded"           type="gml:ReferenceType"&gt;           &lt;xsd:annotation&gt;             &lt;xsd:appinfo&gt;               &lt;gml:targetElement&gt;ClassB&lt;/gml:targetElement&gt;             &lt;/xsd:appinfo&gt;           &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; &lt;xsd:element name="ClassB" type="ClassBType"   substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="ClassBType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="a" maxOccurs="unbounded"           type="gml:ReferenceType"&gt;           &lt;xsd:annotation&gt;             &lt;xsd:appinfo&gt;               &lt;gml:targetElement&gt;ClassA&lt;/gml:targetElement&gt;             &lt;/xsd:appinfo&gt;           &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; &lt;xsd:element name="a2b"   type="a2bType"   substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="a2bType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="a"           type="gml:ReferenceType"&gt;           &lt;xsd:annotation&gt;             &lt;xsd:appinfo&gt;               &lt;gml:targetElement&gt;ClassA&lt;/gml:targetElement&gt;             &lt;/xsd:appinfo&gt;           &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; </pre>

	<pre> &lt;/xsd:element&gt; &lt;xsd:element name="b"   type="gml:ReferenceType"&gt;   &lt;xsd:annotation&gt;     &lt;xsd:appinfo&gt;       &lt;gml:targetElement&gt;ClassB&lt;/gml:targetElement&gt;     &lt;/xsd:appinfo&gt;   &lt;/xsd:annotation&gt; &lt;/xsd:element&gt; &lt;/xsd:sequence&gt; &lt;/xsd:extension&gt; &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; </pre>
GML-Instanz	<pre> &lt;ClassA gml:id="o1"&gt;   &lt;b xlink:href="#o2"/&gt; &lt;/ClassA&gt; &lt;ClassB gml:id="o2"&gt;   &lt;a xlink:href="#o1"/&gt; &lt;/ClassB&gt; &lt;a2b gml:id="x1"&gt;   &lt;a xlink:href="#o1"/&gt;   &lt;b xlink:href="#o2"/&gt; &lt;/a2b&gt; </pre>

### 6.11.3.3 Keine eingebettete Referenzen, nur Zwischentabelle

INTERLIS	<pre> CLASS ClassA = END ClassA;  CLASS ClassB = END ClassB;  ASSOCIATION a2b =   a (EXTERNAL)-- {1..*} ClassA;   b (EXTERNAL)-- {1..*} ClassB; END a2b; </pre>
GML	<pre> &lt;xsd:element name="ClassA" type="ClassAType"   substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="ClassAType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; &lt;xsd:element name="ClassB" type="ClassBType"   substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="ClassBType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; &lt;xsd:element name="a2b"   type="a2bType" </pre>



	<pre> substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="a2bType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="a"           type="gml:ReferenceType"&gt;           &lt;xsd:annotation&gt;             &lt;xsd:appinfo&gt;               &lt;gml:targetElement&gt;ClassA&lt;/gml:targetElement&gt;             &lt;/xsd:appinfo&gt;           &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;         &lt;xsd:element name="b"           type="gml:ReferenceType"&gt;           &lt;xsd:annotation&gt;             &lt;xsd:appinfo&gt;               &lt;gml:targetElement&gt;ClassB&lt;/gml:targetElement&gt;             &lt;/xsd:appinfo&gt;           &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; </pre>
GML-Instanz	<pre> &lt;ClassA gml:id="o1"&gt; &lt;/ClassA&gt; &lt;ClassB gml:id="o2"&gt; &lt;/ClassB&gt; &lt;a2b gml:id="x1"&gt;   &lt;a xlink:href="#o1"/&gt;   &lt;b xlink:href="#o2"/&gt; &lt;/a2b&gt; </pre>

#### 6.11.3.4 Teilweise eingebettete Referenzen und Zwischentabelle

INTERLIS	<pre> CLASS ClassA = END ClassA;  CLASS ClassB = END ClassB;  ASSOCIATION a2b =   a (EXTERNAL) -- {1..*} ClassA;   b (ORDERED) -- {1..*} ClassB; END a2b; </pre>
GML	<pre> &lt;xsd:element name="ClassA" type="ClassAType"   substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="ClassAType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;/xsd:sequence&gt;       &lt;/xsd:extension&gt;     &lt;/xsd:complexContent&gt;   &lt;/xsd:complexType&gt; &lt;xsd:element name="ClassB" type="ClassBType"   substitutionGroup="gml:AbstractFeature"/&gt; </pre>

	<pre> &lt;xsd:complexType name="ClassBType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="a" maxOccurs="unbounded"           type="gml:ReferenceType"&gt;           &lt;xsd:annotation&gt;             &lt;xsd:appinfo&gt;               &lt;gml:targetElement&gt;ClassA&lt;/gml:targetElement&gt;             &lt;/xsd:appinfo&gt;           &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; &lt;xsd:element name="a2b"   type="a2bType"   substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="a2bType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="a"           type="gml:ReferenceType"&gt;           &lt;xsd:annotation&gt;             &lt;xsd:appinfo&gt;               &lt;gml:targetElement&gt;ClassA&lt;/gml:targetElement&gt;             &lt;/xsd:appinfo&gt;           &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;         &lt;xsd:element name="b"&gt;           &lt;xsd:annotation&gt;             &lt;xsd:appinfo&gt;               &lt;gml:targetElement&gt;ClassB&lt;/gml:targetElement&gt;             &lt;/xsd:appinfo&gt;           &lt;/xsd:annotation&gt;           &lt;xsd:complexType&gt;             &lt;xsd:sequence/&gt;             &lt;xsd:attributeGroup               ref="gml:OwnershipAttributeGroup"/&gt;             &lt;xsd:attributeGroup               ref="gml:AssociationAttributeGroup"/&gt;             &lt;xsd:attribute ref="ili:ORDER_POS"/&gt;           &lt;/xsd:complexType&gt;         &lt;/xsd:element&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; </pre>
<p><b>GML-Instanz</b></p>	<pre> &lt;ClassA gml:id="o1"&gt; &lt;/ClassA&gt; &lt;ClassB gml:id="o2"&gt;   &lt;a xlink:href="#o1"/&gt; &lt;/ClassB&gt; &lt;a2b gml:id="x1"&gt;   &lt;a xlink:href="#o1"/&gt;   &lt;b xlink:href="#o2" ORDER_POS="1"/&gt; &lt;/a2b&gt; </pre>

## 6.12 Kodierung von Sichten

Zur Kodierung von Sichten vgl. Kapitel 3.2.3 im INTERLIS-Referenzhandbuch [1]. Als Attribute des Sichtobjekts werden nur diejenigen Attribute übertragen, welche in der Sicht explizit unter «ATTRIBUTE» bzw. implizit mit «ALL OF» angegeben wurden.

## 6.13 Kodierung von Grafikdefinitionen

Für jede Grafikdefinition werden im Transfer die von der Grafikdefinition referenzierten Signaturklassen («Sign-ClassRef») übertragen. Die Objektinstanzen der Signaturklassen werden durch das Ausführen der Grafikdefinitionen auf einem konkreten Inputdatensatz erzeugt. Parameter werden dabei wie Attribute kodiert.

## 6.14 Kodierung von Attributen

Jedes Attribut einer Objektinstanz (einschliesslich komplexer Attribute wie «STRUCTURE», «LIST OF», «BAG OF» etc.) wird wie folgt kodiert.

Desweiteren gelten die im Kapitel 6.15.13 (LINE ATTRIBUTES) und im Kapitel 6.15.15 (mehrere Geometrieattribute in derselben Klasse) definierten speziellen Regeln.

```

AttributeDef = ExplicitTypeAttribute
                | ImplicitTypeAttribute
                | StructAttribute
                | ReferenceAttribute.

ExplicitTypeAttribute =
    '<xsd:element name="%AttributeName%" type="%DataType%"'
    [ 'minOccurs="0"' ] '/>'.

ImplicitTypeAttribute =
    '<xsd:element name="%AttributeName%"'
    [ 'minOccurs="0"' ] '>'
    '<xsd:simpleType>'
    '<xsd:restriction base="%BaseType%">'
    '<!-- any applicable xsd restrictions -->'
    '</xsd:restriction>'
    '</xsd:simpleType>'
    '</xsd:element>'.

StructAttribute = '<xsd:element name="%AttributeName%"'
    [ 'minOccurs="0"' ]
    [ 'maxOccurs="%maxCardinality%"' ] '>'
    '<xsd:complexType>'
    '<xsd:sequence>'
    '<xsd:element ref="%StructureName%"' '/>'
    '</xsd:sequence>'
    '</xsd:complexType>'
    '</xsd:element>'.

ReferenceAttribute =
    '<xsd:element name="%AttributeName%" type="gml:ReferenceType"'
    [ 'minOccurs="0"' ] '>'

```

```
<xsd:annotation>
  <xsd:appinfo>
    <gml:targetElement>%ClassName%</gml:targetElement>
  </xsd:appinfo>
</xsd:annotation>
</xsd:element>'.
```

Beispiel

<p><b>INTERLIS</b></p>	<pre>DOMAIN   Prozent100 = 0 .. 100;  STRUCTURE Koernung =   Ton : Prozent100;   Schluff : Prozent100;   Sand : Prozent100; END Koernung;  CLASS Profil =   Lage : COORD 480000.00 .. 850000.00,            60000.00 .. 320000.00;   KoernungsklasseOberboden : Koernung;   KoernungsklasseUnterboden : Koernung;   Bodenpunktzahl : 0 .. 100; END Profil;</pre>
<p><b>GML</b></p>	<pre>&lt;xsd:simpleType name="Prozent100"&gt;   &lt;xsd:restriction base="xsd:integer"&gt;     &lt;xsd:minInclusive value="0"/&gt;     &lt;xsd:maxInclusive value="100"/&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt; &lt;xsd:element name="Koernung" type="KoernungType"   substitutionGroup="gml:AbstractObject"/&gt; &lt;xsd:complexType name="KoernungType"&gt;    &lt;xsd:sequence&gt;     &lt;xsd:element name="Ton" type="Prozent100"       minOccurs="0"/&gt;     &lt;xsd:element name="Schluff" type="Prozent100"       minOccurs="0"/&gt;     &lt;xsd:element name="Sand" type="Prozent100"       minOccurs="0"/&gt;   &lt;/xsd:sequence&gt;  &lt;/xsd:complexType&gt; &lt;xsd:element name="Profil" type="ProfilType"   substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="ProfilType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="Lage" minOccurs="0"           type="gml:PointPropertyType"&gt;         &lt;/xsd:element&gt;         &lt;xsd:element name="KoernungsklasseOberboden"&gt;           &lt;xsd:complexType&gt;             &lt;xsd:sequence&gt;               &lt;xsd:element ref="Koernung"/&gt;             &lt;/xsd:sequence&gt;           &lt;/xsd:complexType&gt;         &lt;/xsd:element&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt;</pre>

	<pre> &lt;/xsd:element&gt; &lt;xsd:element name="KoernungsklasseUnterboden"&gt;   &lt;xsd:complexType&gt;     &lt;xsd:sequence&gt;       &lt;xsd:element ref="Koernung"/&gt;     &lt;/xsd:sequence&gt;   &lt;/xsd:complexType&gt; &lt;/xsd:element&gt; &lt;xsd:element name="Bodenpunktzahl" minOccurs="0"&gt;   &lt;xsd:simpleType&gt;     &lt;xsd:restriction base="xsd:integer"&gt;       &lt;xsd:minInclusive value="0"/&gt;       &lt;xsd:maxInclusive value="100"/&gt;     &lt;/xsd:restriction&gt;   &lt;/xsd:simpleType&gt; &lt;/xsd:element&gt; &lt;/xsd:sequence&gt; &lt;/xsd:extension&gt; &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; </pre>
GML-Instanz	<pre> &lt;Profil gml:id="o1"&gt;   &lt;Lage&gt;     &lt;gml:Point gml:id="o2" srsName="urn:ogc:def:crs:EPSG::21781"&gt;       &lt;pos&gt;754300.00 247600.00&lt;/pos&gt;     &lt;/gml:Point&gt;   &lt;/Lage&gt;   &lt;KoernungsklasseOberboden&gt;     &lt;Koernung gml:id="o3"&gt;       &lt;Ton&gt;30&lt;/Ton&gt;       &lt;Schluff&gt;20&lt;/Schluff&gt;       &lt;Sand&gt;50&lt;/Sand&gt;     &lt;/Koernung&gt;   &lt;/KoernungsklasseOberboden&gt;   &lt;KoernungsklasseUnterboden&gt;     &lt;Koernung gml:id="o4"&gt;       &lt;Ton&gt;30&lt;/Ton&gt;       &lt;Schluff&gt;10&lt;/Schluff&gt;       &lt;Sand&gt;60&lt;/Sand&gt;     &lt;/Koernung&gt;   &lt;/KoernungsklasseUnterboden&gt;   &lt;Bodenpunktzahl&gt;50&lt;/Bodenpunktzahl&gt; &lt;/Profil&gt; </pre>

## 6.15 Abbildung der INTERLIS-Typen auf XML-Schema-Typen

### 6.15.1 Übersicht

INTERLIS	Abbildung	Complex/Simple Type
TextType	xsd:normalizedString	Simple
EnumerationType	xsd:string / gml:CodeType	Simple/Complex
EnumTreeValueType	xsd:string / gml:CodeType	Simple/Complex

AlignmentType	xsd:string	Simple
BooleanType	xsd:boolean	Simple
NumericType	xsd:integer / xsd:decimal / xsd:double	Simple
FormattedType	xsd:string / xsd:date / xsd:time / xsd:dateTime	Simple
CoordinateType	gml:PointType	Simple
OIDType	xsd:int / xsd:token	Simple
BlackboxType	xsd:anyType / xsd:base64Binary	Complex/Simple
ClassType	xsd:string	Simple
AttributePathType	xsd:string	Simple
LineType	gml:CurveType	Simple
Surface/AreaType	gml:PolygonType	Simple
MulticoordType	gml:MultiPointType	Simple
MultilineType	gml:MultiCurveType	Simple
MultisurfaceType	gml:MultiSurfaceType	Simple
LINE ATTRIBUTES	In einer zusätzlichen Klasse	Complex

### 6.15.2 Kodierung von Zeichenketten

Attribute/Wertebereiche vom Basistyp «TEXT» werden als `xsd:normalizedString`; «MTEXT» als `xsd:string`; «NAME» als `xsd:token` und «URI» als `xsd:anyURI` kodiert. Falls aus dem Modell ableitbar, wird auch `xsd:restriction` kodiert.

#### Beispiel

INTERLIS	<pre>DOMAIN TestText = TEXT; TestTextLen = TEXT*10; TestMText = MTEXT; TestMTextLen = MTEXT*10; TestURI = URI; TestName = NAME;</pre>
GML	<pre>&lt;xsd:simpleType name="TestText"&gt;   &lt;xsd:restriction base="xsd:normalizedString"&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt; &lt;xsd:simpleType name="TestTextLen"&gt;   &lt;xsd:restriction base="xsd:normalizedString"&gt;     &lt;xsd:maxLength value="10"/&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt; &lt;xsd:simpleType name="TestMText"&gt;</pre>

	<pre> &lt;xsd:restriction base="xsd:string"&gt; &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt; &lt;xsd:simpleType name="TestMTextLen"&gt;   &lt;xsd:restriction base="xsd:string"&gt;     &lt;xsd:maxLength value="10"/&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt; &lt;xsd:simpleType name="TestURI"&gt;   &lt;xsd:restriction base="xsd:anyURI"/&gt; &lt;/xsd:simpleType&gt; &lt;xsd:simpleType name="TestName"&gt;   &lt;xsd:restriction base="xsd:token"&gt;     &lt;xsd:maxLength value="255"/&gt;     &lt;xsd:pattern value="[a-zA-Z][a-zA-Z0-9_]*"/&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt; </pre>
--	---

### 6.15.3 Kodierung von Aufzählungen

In INTERLIS können Aufzählungen verfeinert und/oder erweitert werden. Beispiele:

```

Farbe = (rot, gelb, gruen);
FarbeVerfeinert EXTENDS Farbe = (rot (dunkelrot, orange, karmin));
FarbeErweitert EXTENDS Farbe = (schwarz, blau);

```

Da sich mit XML-Schema-Aufzählungen weder verfeinern noch erweitern lassen, wird eine INTERLIS-Aufzählung nur dann in eine XML-Schema-Aufzählung abgebildet, wenn der Wertebereich oder das Attribut als «FINAL» markiert ist und keine Erweiterung eines anderen Wertebereichs ist. Ansonsten wird die Aufzählung als `gml:CodeType` abgebildet.

Für den Codelisten-Identifikator (`codeSpace`) der einzelnen Aufzählwerte kann der Namensraum-Identifikator des Modells (`%StdModelNamespaceIdentifizier%`; siehe Kapitel 6.6), ergänzt durch den abgebildeten Namen der Wertebereichsdefinition verwendet werden (Beispiel: «`http://www.interlis.ch/ILIGML-2.0/ModelA/Farbe`»). Wird die Aufzählung im Rahmen eines Attributs definiert, wird der Namensraum-Identifikator des Modells mit dem abgebildeten Klassennamen und Attributnamen ergänzt

(Beispiel: «`http://www.interlis.ch/ILIGML-2.0/ModelA/Person/HaarFarbe`»).

Mit dem Metaattribut `ili2.iligml20.identifizierCodeSpace` kann der Codelisten-Identifikator (der Werte für das XML-Attribut `gml:codeSpace` in der GML-Instanz) pro Wertebereich definiert werden.

Die Aufzählung (die Liste der möglichen Aufzählwerte) wird als `gml:Dictionary` in einem zusätzlichen GML-Instanz-Dokument pro Modelldatei erzeugt. Dieses GML-Instanz-Dokument wird gem. folgender Regel erstellt:

```

CodelistsInstanceDoc = '<?xml version='1.0' encoding='UTF-8'?>'
  '<gml:Dictionary xmlns:gml="http://www.opengis.net/gml/3.2"
gml:id="%AnyGmlId%">'
  '<gml:identifizier codeSpace=
    "%AnyURI%"
  >' %AnyCode% '</gml:identifizier>'
  '{ '<gml:dictionaryEntry>'
    CodelistInstance

```

```
'</gml:dictionaryEntry>'}
'</gml:Dictionary>'.
CodelistInstance = '<gml:Dictionary
gml:id="%QualifiedEnumTypeDefName%">'
  '<gml:identifier codeSpace=
    "%ModelNamespaceIdentifier%"
    >' %QualifiedEnumTypeDefName% '</gml:identifier>'
  { CodelistEntry }
'</gml:Dictionary>'.
CodelistEntry = '<gml:dictionaryEntry>'
  '<gml:Definition gml:id="%AnyGmlId%">'
  '<gml:identifier codeSpace=
    "%CodespaceIdentifier%"
    >' %EnumValue% '</gml:identifier>'
  '</gml:Definition>'
'</gml:dictionaryEntry>'.
```

AnyURI ist eine gültige URI nach freier Wahl des Instanz-Erstellers.

AnyCode ist ein Codewert nach freier Wahl des Instanz-Erstellers.

AnyGmlId ist ein gültiger gml:id Wert nach freier Wahl des Instanz-Erstellers.

ModelNamespaceIdentifier ist die URI des Modells in dem die Aufzählung definiert wurde (gem. Kap. 6.6).

QualifiedEnumTypeDefName ist der qualifizierte Interlis-Name der Domain, oder des Attributs, welches die Aufzählung definiert.

CodespaceIdentifier ist die URI der Aufzählung wie oben beschrieben hergeleitet oder gem. Metaattribut ili2.iligml20.identifierCodeSpace.

### Beispiel

<p><b>INTERLIS</b></p>	<pre>MODEL ModelA ...  DOMAIN   FarbeFinal (FINAL) = (rot, gelb, gruen);   Farbe = (rot, gelb, gruen);   FarbePlus EXTENDS Farbe=(rot(dunkel,hell),gelb,gruen);  CLASS Fahrzeug =   CarrosserieFarbe : Farbe; END Fahrzeug;  CLASS Auto EXTENDS Fahrzeug =   CarrosserieFarbe (EXTENDED) : FarbePlus; END Auto;</pre>
<p><b>GML</b></p>	<pre>&lt;xsd:simpleType name="FarbeFinal"&gt;   &lt;xsd:restriction base="xsd:string"&gt;     &lt;xsd:enumeration value="rot"/&gt;     &lt;xsd:enumeration value="gelb"/&gt;     &lt;xsd:enumeration value="gruen"/&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt; &lt;xsd:complexType name="Farbe"&gt;   &lt;xsd:restriction     base="gml:CodeType"/&gt;</pre>



	<pre> &lt;/xsd:complexType&gt; &lt;xsd:complexType name="FarbePlus"&gt;   &lt;xsd:restriction     base="gml:CodeType"/&gt; &lt;/xsd:complexType&gt; &lt;xsd:element name="Fahrzeug" type="FahrzeugType"   substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="FahrzeugType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="CarrosserieFarbe" type="Farbe"           minOccurs="0"/&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; &lt;xsd:element name="Auto" type="AutoType"   substitutionGroup="Fahrzeug"/&gt; &lt;xsd:complexType name="AutoType"&gt;   &lt;xsd:annotation&gt;     &lt;xsd:appinfo&gt;       &lt;ili2c:extendedCodelistAttr&gt;         &lt;ili2c:extendedAttribute&gt;CarrosserieFarbe       &lt;/ili2c:extendedAttribute&gt;       &lt;ili2c:codelist&gt;FarbePlus&lt;/ili2c:codelist&gt;     &lt;/ili2c:extendedCodelistAttr&gt;     &lt;/xsd:appinfo&gt;   &lt;/xsd:annotation&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="FahrzeugType"&gt;       &lt;xsd:sequence&gt;         &lt;!-- CarrosserieFarbe ist schon in FahrzeugType           definiert --&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; </pre>
<p><b>GML-Instanz</b></p>	<pre> &lt;Fahrzeug gml:id="o1"&gt;   &lt;CarrosserieFarbe&gt;rot&lt;CarrosserieFarbe&gt; &lt;/Fahrzeug&gt; &lt;Auto gml:id="o2"&gt;   &lt;CarrosserieFarbe&gt;rot.dunkel&lt;CarrosserieFarbe&gt; &lt;/Auto&gt; </pre>
<p><b>GML-Instanz der Aufzählung</b></p>	<p><b>Beispiel: Farbe und FarbePlus</b></p> <pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;gml:Dictionary   xmlns="http://www.opengis.net/gml/3.2"   xmlns:gml="http://www.opengis.net/gml/3.2"   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"   gml:id="f0"&gt;   &lt;identifier codeSpace="http://www.interlis.ch"&gt;ModelA.ili&lt;/identifier&gt;   &lt;dictionaryEntry&gt;     &lt;gml:Dictionary gml:id="ModelA.Farbe"&gt;       &lt;identifier codeSpace="http://www.interlis.ch/ILIGML- 2.0/ModelA"&gt;ModelA.Farbe&lt;/identifier&gt;       &lt;dictionaryEntry&gt;         &lt;gml:Definition gml:id="f1"&gt; </pre>

	<pre>         &lt;gml:identifizier         codeSpace="http://www.interlis.ch/ILIGML-         2.0/ModelA/Farbe"&gt;rot&lt;/gml:identifizier&gt;         &lt;/gml:Definition&gt;         &lt;/dictionaryEntry&gt;         &lt;dictionaryEntry&gt;         &lt;gml:Definition gml:id="f2"&gt;         &lt;gml:identifizier         codeSpace="http://www.interlis.ch/ILIGML-         2.0/ModelA/Farbe"&gt;gelb&lt;/gml:identifizier&gt;         &lt;/gml:Definition&gt;         &lt;/dictionaryEntry&gt;         &lt;dictionaryEntry&gt;         &lt;gml:Definition gml:id="f3"&gt;         &lt;gml:identifizier         codeSpace="http://www.interlis.ch/ILIGML-         2.0/ModelA/Farbe"&gt;gruen&lt;/gml:identifizier&gt;         &lt;/gml:Definition&gt;         &lt;/dictionaryEntry&gt;         &lt;/gml:Dictionary&gt;         &lt;/dictionaryEntry&gt;         &lt;dictionaryEntry&gt;         &lt;gml:Dictionary gml:id="ModelA.FarbePlus"&gt;         &lt;identifizier codeSpace="http://www.interlis.ch/ILIGML-         2.0/ModelA"&gt;ModelA.FarbePlus&lt;/identifizier&gt;         &lt;dictionaryEntry&gt;         &lt;gml:Definition gml:id="f10"&gt;         &lt;gml:identifizier         codeSpace="http://www.interlis.ch/ILIGML-         2.0/ModelA/FarbePlus"&gt;rot.dunkel&lt;/gml:identifizier&gt;         &lt;/gml:Definition&gt;         &lt;/dictionaryEntry&gt;         &lt;dictionaryEntry&gt;         &lt;gml:Definition gml:id="f11"&gt;         &lt;gml:identifizier         codeSpace="http://www.interlis.ch/ILIGML-         2.0/ModelA/FarbePlus"&gt;rot.hell&lt;/gml:identifizier&gt;         &lt;/gml:Definition&gt;         &lt;/dictionaryEntry&gt;         &lt;dictionaryEntry&gt;         &lt;gml:Definition gml:id="f12"&gt;         &lt;gml:identifizier         codeSpace="http://www.interlis.ch/ILIGML-         2.0/ModelA/FarbePlus"&gt;gelb&lt;/gml:identifizier&gt;         &lt;/gml:Definition&gt;         &lt;/dictionaryEntry&gt;         &lt;dictionaryEntry&gt;         &lt;gml:Definition gml:id="f13"&gt;         &lt;gml:identifizier         codeSpace="http://www.interlis.ch/ILIGML-         2.0/ModelA/FarbePlus"&gt;gruen&lt;/gml:identifizier&gt;         &lt;/gml:Definition&gt;         &lt;/dictionaryEntry&gt;         &lt;/gml:Dictionary&gt;         &lt;/dictionaryEntry&gt;         &lt;/gml:Dictionary&gt;     </pre>
--	---

Die Aufzählungswerte werden in beiden Fällen wie folgt kodiert:

```
EnumValue = ( EnumElement-Name { '.' EnumElement-Name } ) .
```

Für die Kodierung von Aufzählungswerten (unabhängig davon, ob der Wertebereich nur die Blätter oder auch die Knoten umfasst) wird die Syntax für Aufzählungskonstanten angewendet (Regel «EnumValue»). Das Zeichen «#» wird dabei weggelassen.

Die vordefinierten Textausrichtungstypen «HALIGNMENT» und «VALIGNMENT» werden im Anhang A definiert. Für den Typ «BOOLEAN» wird der entsprechende XML-Schema Typ `xsd:boolean` verwendet.

### 6.15.4 Kodierung von numerischen Datentypen

Numerische Datentypen werden je nach Unter- und Obergrenzwert als `xsd:integer`, `xsd:decimal` oder `xsd:double` kodiert. Falls aus dem Modell ableitbar, wird auch `xsd:restriction` kodiert.

Beispiel	
INTERLIS	<pre>DOMAIN TestInt = 1 .. 10; TestDec = 1.0 .. 10.0; TestDouble = 0.123e1 .. 0.234e1;</pre>
GML	<pre>&lt;xsd:simpleType name=" TestInt"&gt;   &lt;xsd:restriction base="xsd:integer"&gt;     &lt;xsd:minInclusive value="1"/&gt;     &lt;xsd:maxInclusive value="10"/&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt; &lt;xsd:simpleType name="TestDec"&gt;   &lt;xsd:restriction base="xsd:decimal"&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt; &lt;xsd:simpleType name="TestDouble"&gt;   &lt;xsd:restriction base="xsd:double"&gt;     &lt;xsd:minInclusive value="1.23"/&gt;     &lt;xsd:maxInclusive value="2.34"/&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt;</pre>

### 6.15.5 Kodierung von formatierten Wertebereichen

Formatierte Wertebereiche, die «INTERLIS.XMLDate», «INTERLIS.XMLTime» oder «INTERLIS.XMLDateTime» referenzieren, werden in die entsprechenden XML-Schema-Datumstypen (`xsd:date`, `xsd:time`, `xsd:dateTime`) abgebildet. Alle anderen formatierten Wertebereiche werden in ein `xsd:string` abgebildet.

Beispiel	
INTERLIS	<pre>STRUCTURE GregorianDate = Year: 1582 .. 2999; SUBDIVISION Month: 1 .. 12; SUBDIVISION Day: 1 .. 31; END GregorianDate;</pre>

	<pre> DOMAIN BuchungsDatum = FORMAT INTERLIS.XMLDate                 "2002-01-01" ..                 "2007-12-31"; StartZeit = FORMAT INTERLIS.XMLTime             "00:00:00.000" ..             "23:59:59.999"; MessZeitpunkt = FORMAT INTERLIS.XMLDateTime               "2002-01-01T00:00:00.000" ..               "2007-12-31T23:59:59.999"; EigenesDatum = FORMAT BASED ON GregorianDate               ( Year "Y" Month "M" Day "D"); </pre>
GML	<pre> &lt;xsd:simpleType name="BuchungsDatum"&gt;   &lt;xsd:restriction base="xsd:date"/&gt; &lt;/xsd:simpleType&gt; &lt;xsd:simpleType name="StartZeit"&gt;   &lt;xsd:restriction base="xsd:time"/&gt; &lt;/xsd:simpleType&gt; &lt;xsd:simpleType name="MessZeitpunkt"&gt;   &lt;xsd:restriction base="xsd:dateTime"/&gt; &lt;/xsd:simpleType&gt; &lt;xsd:simpleType name="EigenesDatum"&gt;   &lt;xsd:restriction base="xsd:string"/&gt; &lt;/xsd:simpleType&gt; </pre>

### 6.15.6 Kodierung von Gefässen

Die XML-Variante des Typs «BLACKBOX» wird als `xsd:any` kodiert, die binäre Variante als `xsd:base64Binary`.

Beispiel	
INTERLIS	<pre> DOMAIN BlackboxXml = BLACKBOX XML; BlackboxBinary = BLACKBOX BINARY; </pre>
GML	<pre> &lt;xsd:complexType name="BlackboxXml"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:any namespace="##any"               minOccurs="0" maxOccurs="unbounded"               processContents="lax"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt; &lt;xsd:simpleType name="BlackboxBinary"&gt;   &lt;xsd:restriction base="xsd:base64Binary"&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt; </pre>

### 6.15.7 Kodierung von Klassentypen

Der Klassentyp wird als `xsd:normalizedString` kodiert.

Beispiel	
INTERLIS	<pre> DOMAIN InterlisClassRef = CLASS; </pre>

GML	<pre>&lt;xsd:simpleType name="InterlisClassRef"&gt;   &lt;xsd:restriction base="xsd:normalizedString"&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt;</pre>
-----	--

Der Wert enthält den vollständig qualifizierten Klassen-, Struktur- oder Assoziationsnamen (z.B. «DM01AVCH24D.FixpunkteKategorie1.LFP1»).

### 6.15.8 Kodierung von Attributpfadtypen

Der Attributpfadtyp wird als `xsd:normalizedString` kodiert.

Beispiel	
INTERLIS	<pre>DOMAIN   InterlisAttributeRef = ATTRIBUTE;</pre>
GML	<pre>&lt;xsd:simpleType name="InterlisAttributeRef"&gt;   &lt;xsd:restriction base="xsd:normalizedString"&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt;</pre>

Der Wert enthält den vollständig qualifizierten Klassennamen gefolgt vom durch einen Punkt abgetrennten Attributnamen (z.B. «Grunddatensatz.Fixpunkte.LFP.Nummer»).

### 6.15.9 Kodierung von Koordinaten

Koordinaten werden als `gml:PointPropertyType` kodiert.

Beispiel	
INTERLIS	<pre>CLASS Profil =   Lage : COORD 480000.00 .. 850000.00,               60000.00 .. 320000.00; END Profil;</pre>
GML	<pre>&lt;xsd:element name="Profil" type="ProfilType" substitution- Group="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="ProfilType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="Lage" minOccurs="0" ty- pe="gml:PointPropertyType"/&gt;         ...       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt;</pre>

Angaben zu Achsen-Wertebereich, Einheit, Referenzsystem und Richtungssinn lassen sich im GML-Applikationsschema nicht festhalten.

Falls im INTERLIS-Modell kein Referenzsystem definiert wurde, soll in der GML-Instanz ein werkzeugspezifischer Default-Wert für das XML-Attribut srsName verwendet werden. Falls im INTERLIS-Modell ein Referenzsystem definiert wurde, soll in der GML-Instanz ein Wert der diesem Referenzsystem entspricht, verwendet werden. Der Wert kann bei der Deklaration des Referenzsystemobjektes mit Hilfe des Metaattributes «ili2.iligml20.srsName» definiert werden. Falls kein Metaattribut definiert ist, ist die Abbildung werkzeugspezifisch.

Beispiel	
INTERLIS	<pre>REFSYSTEM BASKET BCoordSys ~ CoordSys.CoordsysTopic OBJECTS OF GeoCartesian2D : !!@ ili2.iligml20.srsName="urn:ogc:def:crs:EPSG::21781" CHLV03;</pre>

### 6.15.10 Kodierung von Linienzügen

Linienzüge werden als `gml:CurvePropertyType` kodiert.

Beispiel	
INTERLIS	<pre>DOMAIN Point2D = COORD 0.000 .. 200.000 [INTERLIS.m], !! Min_East Max_East 0.000 .. 200.000 [INTERLIS.m], !! Min_North Max_North ROTATION 2 -&gt; 1;  CLASS StreetAxis = Geometry: MANDATORY POLYLINE WITH (STRAIGHTS) VERTEX Point2D; END StreetAxis;</pre>
GML	<pre>&lt;xsd:element name="StreetAxis" type="StreetAxisType" substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="StreetAxisType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="Geometry" minOccurs="0" type="gml:CurvePropertyType"/&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt;</pre>

Angaben zu Stützpunkt-Wertebereich und zulässigen Kurvenstück-Formen lassen sich im GML-Applikationsschema nicht festhalten.

### 6.15.11 Kodierung von Einzelflächen

Einzelflächen werden als `gml:SurfacePropertyType` kodiert. In der Instanz ist dann aber nur `gml:Polygon` zulässig.

Beispiel	
INTERLIS	<pre>DOMAIN Point2D = COORD   0.000 .. 200.000 [INTERLIS.m], !! Min_East Max_East   0.000 .. 200.000 [INTERLIS.m], !! Min_North Max_North ROTATION 2 -&gt; 1;  CLASS LandCover =   Geometry: MANDATORY SURFACE WITH (STRAIGHTS)   VERTEX Point2D WITHOUT OVERLAPS &gt; 0.100; END LandCover;</pre>
GML	<pre>&lt;xsd:element name="LandCover" type="LandCoverType" substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="LandCoverType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="Geometry" minOccurs="0" type="gml:SurfacePropertyType"/&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt;</pre>
GML-Instanz	<pre>&lt;LandCover gml:id="o0"&gt;   &lt;Geometry&gt;     &lt;gml:Polygon gml:id=="g0" srsDimension="2" srsName="http://www.opengis.net/gml/srs/epsg.xml#21781"&gt;       &lt;gml:exterior&gt;         &lt;gml:LinearRing&gt;           &lt;gml:posList&gt;...&lt;/gmlposList&gt;         &lt;/gml:LinearRing&gt;       &lt;/gml:exterior&gt;       &lt;gml:interior&gt;         &lt;gml:LinearRing&gt;           &lt;gml:posList&gt;...&lt;/gmlposList&gt;         &lt;/gml:LinearRing&gt;       &lt;/gml:interior&gt;     &lt;/gml:Polygon&gt;   &lt;/Geometry&gt; &lt;/LandCover&gt;</pre>

Angaben zu Stützpunkt-Wertebereich und zulässigen Kurvenstück-Formen lassen sich im GML-Applikationsschema nicht festhalten.

### 6.15.12 Kodierung von Gebietseinteilungen

Gebietseinteilungen werden als `gml:SurfacePropertyType` kodiert. In der Instanz ist dann aber nur `gml:Polygon` zulässig.

Beispiel	
INTERLIS	<pre>DOMAIN ... CLASS Grundstueck =   Geometrie : MANDATORY AREA WITH (ARCS,STRAIGHTS)   VERTEX LKoord WITHOUT OVERLAPS&gt;0.10 END Grundstueck;</pre>
GML	<pre>&lt;xsd:element name="Grundstueck" type="GrundstueckType" substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="GrundstueckType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="Geometrie" minOccurs="0" type="gml:SurfacePropertyType"/&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt;</pre>

Angaben zu Stützpunkt-Wertebereich und zulässigen Kurvenstück-Formen lassen sich im GML-Applikationsschema nicht festhalten, ebenso wenig, dass es sich um eine Gebietseinteilung handelt.

### 6.15.13 Kodierung von LINE ATTRIBUTES

Falls bei Einzelflächen und Gebietseinteilungen Linienattribute vorkommen (INTERLIS 2.3), müssen zwei entsprechende Klassen generiert werden:

- Eine Hauptklasse, die die Eigenschaften der INTERLIS-Klasse (ausser den LINE ATTRIBUTES) enthält. Die Geometrie dieser Klasse wird als `gml:SurfacePropertyType` kodiert.
- Eine zusätzliche Klasse, die die Randlinien mit den LINE ATTRIBUTES enthält. Ein Objekt dieser Klasse enthält eine Beziehung zu dem/den entsprechenden Objekt/ten (min. 1, max. 2) der Hauptklasse.

Lineattribute sind im INTERLIS 2.4 nicht mehr vorhanden.

Beispiel	
INTERLIS	<pre>DOMAIN LKoord = COORD 480000.00 .. 850000.00 [m] {CHLV03[1]},           60000.00 .. 320000.00 [m] {CHLV03[2]},           ROTATION 2 -&gt; 1; STRUCTURE GrenzlinieEigenschaften =   streitig : BOOLEAN; END GrenzlinieEigenschaften;</pre>



	<pre> CLASS Grundstueck =   Geometrie : MANDATORY AREA WITH (ARCS,STRAIGHTS)   VERTEX LKoord   WITHOUT OVERLAPS&gt;0.10   LINE ATTRIBUTES GrenzlinieEigenschaften; END Grundstueck; </pre>
<p><b>GML</b></p>	<pre> &lt;xsd:element name="GrenzlinieEigenschaften" ty- pe="GrenzlinieEigenschaftenType" substitution- Group="gml:AbstractObject"/&gt; &lt;xsd:complexType name="GrenzlinieEigenschaftenType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="streitig" type="xsd:boolean" minOccurs="0"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt;  &lt;xsd:element name="Grundstueck.Grenze" type="Grundstueck.GrenzeType" substitu- tionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="Grundstueck.GrenzeType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="Geometry" minOccurs="0" ty- pe="gml:CurvePropertyType"/&gt;         &lt;xsd:element name="lineattr" minOccurs="0" ty- pe="gml:GrenzlinieEigenschaften"/&gt;         &lt;xsd:element name="Grundstueck" minOccurs="1" max- Occurs="2" type="gml:ReferenceType"/&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt;  &lt;xsd:element name="Grundstueck" type="GrundstueckType" sub- stitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="GrundstueckType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="Geometrie" minOccurs="0" type="gml:SurfacePropertyType"/&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; </pre>

### 6.15.14 Kodierung von Multigeometrien

Multigeometrien werden als GML-Multigeometrien kodiert, bzw. MULTICOORD als gml:MultiPointPropertyType, MULTIPOLYLINE als gml:MultiCurvePropertyType und MULTISURFACE als gml:MultiSurfacePropertyType.

Die Kodierung von INTERLIS-2.3-Multigeometrien erfolgt über Strukturen.

Beispiel	
INTERLIS	DOMAIN

	<pre> ... CLASS Multipunkte =   Geometrie : MULTICOORD2 END Multipunkte; </pre>
GML	<pre> &lt;xsd:element name="Multipunkte" type="MultipunkteType" substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="MultipunkteType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="Geometrie" minOccurs="0" type="gml:MultiPointPropertyType"/&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; ... </pre>

### 6.15.15 Kodierung unterschiedlicher Geometrieattribute innerhalb derselben Klasse

Falls eine Klasse mehrere Geometrieattribute hat und wegen einer Konsistenzbedingung nur eines der Attribute einen Wert haben darf, werden diese als `xsd:choice` kodiert. Die Konsistenzbedingung muss dabei folgendes Muster haben:

```

DEFINED(attr1) AND NOT(DEFINED(attr2)) OR (NOT(DEFINED(attr1)) AND
DEFINED(attr2))

```

#### Beispiel INTERLIS 2.3

INTERLIS	<pre> DOMAIN ... CLASS Landcover =   Punkt:LKoord;   Linie:POLYLINE WITH (ARCS,STRAIGHTS) VERTEX LKoord;   Flaechе:SURFACE WITH (ARCS,STRAIGHTS) VERTEX LKoord ...   MANDATORY CONSTRAINT (DEFINED(Punkt) AND NOT(DEFINED(Linie)) AND NOT(DEFINED(Flaechе))) OR (NOT (DEFINED(Punkt)) AND DEFINED(Linie) AND NOT( DEFINED(Flaechе))) OR (NOT( DEFINED(Punkt)) AND NOT( DEFINED(Linie)) AND DEFINED(Flaechе)); END Landcover; </pre>
GML	<pre> &lt;xsd:element name="Landcover" type="LandcoverType" substitutionGroup="gml:AbstractFeature"/&gt; &lt;xsd:complexType name="LandcoverType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="gml:AbstractFeatureType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:choice&gt;           &lt;xsd:element name="Punkt" type="gml:PointPropertyType"/&gt;           &lt;xsd:element name="Linie" type="gml:CurvePropertyType"/&gt;           &lt;xsd:element name="Flaechе" type="gml:SurfacePropertyType"/&gt;         &lt;/xsd:choice&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; </pre>

	<pre> &lt;/xsd:sequence&gt; &lt;/xsd:extension&gt; &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; ... </pre>
--	--

### 6.15.16 Kodierung von Typen zur Objektidentifikation

Typen für Objektidentifikationen werden als `xsd:int` («OID NumericType») oder `xsd:token` («OID» als «ANY» oder «TextType») abgebildet.

Mit dem Metaattribut «`ili2.iligml20.identifizierCodeSpace`» kann die Formatierung des Identifikatorwertes (der Werte für das XML-Attribut `gml:codeSpace` in der GML-Instanz) definiert werden.

Mit dem Metaattribut «`ili2.iligml20.identifizierPattern`» kann die Formatierung des Identifikatorwertes (der Werte für das XML-Element `gml:identifizier` in der GML-Instanz) definiert werden.

Mit dem Metaattribut «`ili2.iligml20.hrefPattern`» kann die Formatierung des Referenzwertes (der Werte für das XML-Attribut `xlink:href` in der GML-Instanz) definiert werden.

In den Musterdefinitionen kann «`#{value}`» als Platzhalter für den OID-Wert verwendet werden.

Beispiel	
<b>INTERLIS</b>	<pre> DOMAIN BFS_EGID = OID 1 .. 9999999999; UUID = OID TEXT*36; !!@ ili2.iligml20.identifizierCodeSpace= "http://inspire.jrc.ec.europa.eu/" !!@ ili2.iligml20.identifizierPattern= "urn:inspire:object:id:#{value}" !!@ ili2.iligml20.hrefPattern= "urn:inspire:object:id:#{value}" InspiredId = OID TEXT; !! z.B. "DENW1000:A00012ab" </pre>
<b>GML</b>	<pre> &lt;xsd:simpleType name="BFS_EGID"&gt;   &lt;xsd:restriction base="xsd:int"&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt; &lt;xsd:simpleType name="UUID"&gt;   &lt;xsd:restriction base="xsd:token"&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt; &lt;xsd:simpleType name="InspiredId"&gt;   &lt;xsd:restriction base="xsd:token"&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt; </pre>

### 6.15.17 **Kodierung von weiteren Kurvenstück-Formen**

Wird in dieser Version dieser Spezifikation nicht unterstützt.

*Anmerkung:* Es muss zuerst ein INTERLIS-Basismodell erstellt werden, so dass eine Abbildung auf in GML vordefinierte Kurventypen möglich ist.

## 7 Instanz-Kodierungsregeln

Zusätzlich zu den Schema-Codierungsregeln gelten die in diesem Abschnitt definierten Instanz-Regeln.

### 7.1 Kodierung mehrerer Behälter (TOPICs) in einer Transferdatei

Werden in einer Transferdatei mehrere Behälter transferiert (typischerweise der Fall) wird als Wurzel-XML-Element «TRANSFER» verwendet. Dieses Element und die zugehörigen Typen sind im Anhang A definiert.

### 7.2 Kodierung von Objekten und Strukturelementen

Objekte werden immer inline zu einem Behälter kodiert. Objekte werden nie inline zu einem anderen Objekt kodiert.

Strukturelemente werden immer inline zu einem Objekt oder anderen Strukturelement kodiert. Strukturelemente werden nie referenziert, obwohl sie eine `gml:id` erhalten.

### 7.3 Kodierung von Objekt- und Behälteridentifikationen

Die Transferidentifikation eines Objektes oder Behälters wird als `gml:id` kodiert.

Die stabile Identifikation des Objektes oder Behälters wird als `gml:identifizier` kodiert.

Falls bei der Wertebereichsdefinition des OID für das Metaattribut «`ili2.iligml20.identifizierCodeSpace`» kein Wert definiert wurde, wird als «codeSpace» der Namensraum-Identifikator des Modells (`%StdModelNamespaceIdentifizier%`; siehe Kapitel 6.6), ergänzt durch den abgebildeten Namen der Wertebereichsdefinition für die OIDs verwendet (Beispiel für den Wertebereich «UUIDOID» aus dem Modell INTERLIS (Anhang A im Referenzhandbuch [1]): «`http://www.interlis.ch/ILIGML-2.0/INTERLIS/UUIDOID`»).

Falls bei der Wertebereichsdefinition des OID für das Metaattribut «`ili2.iligml20.identifizierCodeSpace`» ein Wert definiert wurde, wird dieser verwendet.

Falls bei der Wertebereichsdefinition des OID für das Metaattribut «`ili2.iligml20.identifizierPattern`» kein Wert definiert wurde, wird der Wert unverändert verwendet (das entspricht dem Muster «`#{value}`»).

Falls bei der Wertebereichsdefinition des OID für das Metaattribut «`ili2.iligml20.identifizierPattern`» ein Wert definiert wurde, wird dieser verwendet.

## 7.4 Kodierung von Beziehungen

Als Referenzwert wird immer die `gml:id` verwendet (z.B. `xlink:href="#o1` als Verweis auf ein Objekt mit `gml:id="o1"`), ausser wenn das referenzierte Objekt eine stabile OID hat.

Wenn das referenzierte Objekt eine stabile OID hat, wird diese für die Referenz verwendet.

Falls bei der Wertebereichsdefinition für den OID kein Muster zur Bildung des `xlink:href`-Werts definiert wurde, wird «urn:x-ili:\${value}» als Muster verwendet (z.B. `xlink:href="urn:x-ili:o1` als Verweis auf ein Objekt mit `gml:identifier="o1"`).

Falls bei der Wertebereichsdefinition für den OID ein Muster zur Bildung des `xlink:href`-Werts definiert wurde, wird dieses Muster verwendet.

## 7.5 Kodierung von Geometrien

Alle Geometrien müssen inline kodiert werden, auch wenn das GML-Basischema Referenzen zulassen würde.

Bei «SURFACE» und «AREA» ist nur `gml:Polygon` zulässig (`gml:Surface` ist also *nicht* zulässig).

Bei `gml:LineStringSegment` ist in GML-SF nur `gml:posList` zulässig.

Bei `gml:exterior/gml:interior` ist in GML-SF nur `gml:LinearRing` (mit `gml:posList`) zulässig.

Bei `gml:Arc` ist in GML-SF nur `gml:posList` zulässig.

## 7.6 Kodierung von Code Lists

Die Instanz-Kodierung von Aufzählungen ist im Kapitel 6.15.3 definiert.

## 8 Haftungsausschluss/Hinweise auf Rechte Dritter

**eCH**-Standards, welche der Verein **eCH** dem Benutzer zur unentgeltlichen Nutzung zur Verfügung stellt, oder welche **eCH** referenziert, haben nur den Status von Empfehlungen. Der Verein **eCH** haftet in keinem Fall für Entscheidungen oder Massnahmen, welche der Benutzer auf Grund dieser Dokumente trifft und / oder ergreift. Der Benutzer ist verpflichtet, die Dokumente vor deren Nutzung selbst zu überprüfen und sich gegebenenfalls beraten zu lassen. **eCH**-Standards können und sollen die technische, organisatorische oder juristische Beratung im konkreten Einzelfall nicht ersetzen.

In **eCH**-Standards referenzierte Dokumente, Verfahren, Methoden, Produkte und Standards sind unter Umständen markenrechtlich, urheberrechtlich oder patentrechtlich geschützt. Es liegt in der ausschliesslichen Verantwortlichkeit des Benutzers, sich die allenfalls erforderlichen Rechte bei den jeweils berechtigten Personen und/oder Organisationen zu beschaffen.

Obwohl der Verein **eCH** all seine Sorgfalt darauf verwendet, die **eCH**-Standards sorgfältig auszuarbeiten, kann keine Zusicherung oder Garantie auf Aktualität, Vollständigkeit, Richtigkeit bzw. Fehlerfreiheit der zur Verfügung gestellten Informationen und Dokumente gegeben werden. Der Inhalt von **eCH**-Standards kann jederzeit und ohne Ankündigung geändert werden.

Jede Haftung für Schäden, welche dem Benutzer aus dem Gebrauch der **eCH**-Standards entstehen ist, soweit gesetzlich zulässig, wegbedungen.

## 9 Urheberrechte

Wer **eCH**-Standards erarbeitet, behält das geistige Eigentum an diesen. Allerdings verpflichtet sich der Erarbeitende sein betreffendes geistiges Eigentum oder seine Rechte an geistigem Eigentum anderer, sofern möglich, den jeweiligen Fachgruppen und dem Verein **eCH** kostenlos zur uneingeschränkten Nutzung und Weiterentwicklung im Rahmen des Vereinszweckes zur Verfügung zu stellen.

Die von den Fachgruppen erarbeiteten Standards können unter Nennung der jeweiligen Urheber von **eCH** unentgeltlich und uneingeschränkt genutzt, weiterverbreitet und weiterentwickelt werden.

**eCH**-Standards sind vollständig dokumentiert und frei von lizenz- und/oder patentrechtlichen Einschränkungen. Die dazugehörige Dokumentation kann unentgeltlich bezogen werden.

Diese Bestimmungen gelten ausschliesslich für die von **eCH** erarbeiteten Standards, nicht jedoch für Standards oder Produkte Dritter, auf welche in den **eCH**-Standards Bezug genommen wird. Die Standards enthalten die entsprechenden Hinweise auf die Rechte Dritter.

## Anhang A – Vordefiniertes Basis-Schema

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.interlis.ch/ILIGML-2.0/INTERLIS"
  xmlns:ili2="http://www.interlis.ch/ili2"
  targetNamespace=
    "http://www.interlis.ch/ILIGML-2.0/INTERLIS"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  xmlns:gml="http://www.opengis.net/gml/3.2">

  <xsd:annotation>
    <xsd:appinfo source="http://www.interlis.ch/ili2c">
      <ili2:model>INTERLIS</ili2:model>
      <ili2:modelVersion>2016-01-31</ili2:modelVersion>
      <ili2:modelAt>http://www.interlis.ch</ili2:modelAt>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:import namespace="http://www.opengis.net/gml/3.2" schemaLoca-
    tion="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>

  <xsd:attribute name="ORDER_POS" type="xsd:positiveInteger"/>

  <xsd:simpleType name="HALIGNMENT">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Left"/>
      <xsd:enumeration value="Center"/>
      <xsd:enumeration value="Right"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="VALIGNMENT">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Top"/>
      <xsd:enumeration value="Cap"/>
      <xsd:enumeration value="Half"/>
      <xsd:enumeration value="Base"/>
      <xsd:enumeration value="Bottom"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="I32OID">
    <xsd:restriction base="xsd:int">
      <xsd:minInclusive value="0"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="STANDARDROID">
    <xsd:restriction base="xsd:token">
      <xsd:length value="16"/>
      <xsd:pattern value="[a-zA-Z][a-zA-Z0-9]*"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="UUIDOID">
    <xsd:restriction base="xsd:token">
      <xsd:length value="36"/>
      <xsd:pattern value=
        "[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}"/>
    </xsd:restriction>
  </xsd:simpleType>
<xsd:complexType name="TRANSFERMemberType">

```



```
<xsd:complexContent>
  <xsd:extension base="gml:AbstractFeatureMemberType">
    <xsd:sequence>
      <xsd:choice>
        <xsd:element ref="gml:AbstractFeature"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:element name="TRANSFER" type="TRANSFERType"
  substitutionGroup="gml:AbstractFeature"/>

<xsd:complexType name="TRANSFERType">
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractFeatureType">
      <xsd:sequence>
        <xsd:element name="baskets"
          type="TRANSFERMemberType"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="gml:AggregationAttributeGroup"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
</xsd:schema>
```

## Anhang B – ili2c-Basis-Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.interlis.ch/ili2c/ILIGML-2.0"
  targetNamespace=
    "http://www.interlis.ch/ili2c/ILIGML-2.0"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  >

  <xsd:element name="Model" type="ModelType"/>
  <xsd:complexType name="ModelType">
    <xsd:sequence>
      <xsd:element name="modelName" type="xsd:token"/>
      <xsd:element name="modelVersion"
type="xsd:normalizedString" minOccurs="0"/>
      <xsd:element name="modelIssuer" type="xsd:anyURI"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="ExtendedCodelistAttr"
type="ExtendedCodelistAttrType"/>
  <xsd:complexType name="ExtendedCodelistAttrType">
    <xsd:sequence>
      <xsd:element name="extendedAttribute" type="xsd:Name"/>
      <xsd:element name="codelist" type="xsd:QName"/>
    </xsd:sequence>
  </xsd:complexType>

</xsd:schema>
```

## Anhang C – Beispiel

### INTERLIS 2-Modell

```
INTERLIS 2.4;

MODEL Beispiel (de)
AT "mailto:ceis@localhost"
VERSION "2016-01-31" =

  DOMAIN

    LKoord = COORD 100.00 .. 300.00, 100.00 .. 300.00;

  TOPIC Bodenbedeckung =

    CLASS BoFlaechen =
      Art (FINAL): MANDATORY (
        Gebaeude,
        befestigt,
        humusiert,
        Gewaesser,
        bestockt,
        vegetationslos);
      Form : MANDATORY AREA WITH (ARCS,STRAIGHTS)
            VERTEX Beispiel.LKoord
            WITHOUT OVERLAPS>0.10;
    END BoFlaechen;

    CLASS Strasse =
      Achse : MANDATORY POLYLINE WITH (ARCS,STRAIGHTS)
            VERTEX LKoord;
    END Strasse;

    CLASS Gebaeude =
      PositionHauseingang : LKoord;
      AssNr : MANDATORY TEXT*6;
      UNIQUE AssNr;
    END Gebaeude;

    ASSOCIATION GebaeudeFlaechen (FINAL)=
      Gebaeude -- {0..*} Gebaeude;
      Flaechen -- {1} BoFlaechen;
    END GebaeudeFlaechen;

  END Bodenbedeckung;

END Beispiel.
```

## INTERLIS 1-Modell

```
TRANSFER Beispiel;

DOMAIN
  LKoord = COORD2 100.00 100.00
              300.00 300.00;

MODEL Beispiel

TOPIC Bodenbedeckung =
  TABLE BoFlaechen =
    Art: (Gebaeude,
          befestigt,
          humusiert,
          Gewaesser,
          bestockt,
          vegetationslos);
    Form: AREA WITH (STRAIGHTS, ARCS) VERTEX LKoord
           WITHOUT OVERLAPS > 0.10;
  NO IDENT
END BoFlaechen;

TABLE Strasse =
  Achse : POLYLINE WITH (ARCS,STRAIGHTS)
           VERTEX LKoord;
  NO IDENT
END Strasse;

TABLE Gebaeude =
  PositionHauseingang : LKoord;
  AssNr: TEXT*6;
  Flaechen: -> BoFlaechen // Art = Gebaeude //;
IDENT
  AssNr; !! Annahme AssNr sei eindeutig.
  Flaechen; !! Dem Gebaeude ist genau eine
             !! Flaechen zugeordnet
END Gebaeude;

END Bodenbedeckung.

END Beispiel.

FORMAT FREE;
CODE BLANK = DEFAULT, UNDEFINED = DEFAULT, CONTINUE = DEFAULT;
TID = ANY;

END.
```

## GML-Applikationsschema

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.interlis.ch/ILIGML-2.0/Beispiel"
  targetNamespace="http://www.interlis.ch/ILIGML-2.0/Beispiel"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  xmlns:gml="http://www.opengis.net/gml/3.2">
  <xsd:annotation>
    <xsd:appinfo source="http://www.interlis.ch/ili2c"
      xmlns:ili2="http://www.interlis.ch/ili2">
      <ili2:model>Beispiel</ili2:model>
      <ili2:modelVersion>2016-01-31</ili2:modelVersion>
      <ili2:modelAt>mailto:ceis@localhost</ili2:modelAt>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:import namespace="http://www.opengis.net/gml/3.2" schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>

  <xsd:element name="BoFlaechen" type="BoFlaechenType"
    substitutionGroup="gml:AbstractFeature"/>
  <xsd:complexType name="BoFlaechenType">
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractFeatureType">
        <xsd:sequence>
          <xsd:element name="Art">
            <xsd:simpleType>
              <xsd:restriction base="xsd:normalizedString">
                <xsd:enumeration value="Gebaeude"/>
                <xsd:enumeration value="befestigt"/>
                <xsd:enumeration value="humusiert"/>
                <xsd:enumeration value="Gewaesser"/>
                <xsd:enumeration value="bestockt"/>
                <xsd:enumeration value="vegetationslos"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="Form" type="gml:SurfacePropertyType"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:element name="Strasse" type="StrasseType"
    substitutionGroup="gml:AbstractFeature"/>
  <xsd:complexType name="StrasseType">
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractFeatureType">
        <xsd:sequence>
          <xsd:element name="Achse" type="gml:CurvePropertyType"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:element name="Gebaeude" type="GebaeudeType"
    substitutionGroup="gml:AbstractFeature"/>
  <xsd:complexType name="GebaeudeType">
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractFeatureType">
        <xsd:sequence>
          <xsd:element name="PositionHauseingang"
            type="LKoord"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

```

```

        <xsd:element name="AssNr">
          <xsd:simpleType>
            <xsd:restriction base="xsd:normalizedString">
              <xsd:maxLength value="6"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Flaeche" type="gml:ReferenceType">
          <xsd:annotation>
            <xsd:appinfo>
              <gml:targetElement>BoFlaechen</gml:targetElement>
            </xsd:appinfo>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="BodenbedeckungMemberType">
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractFeatureMemberType">
      <xsd:sequence>
        <xsd:choice>
          <xsd:element ref="BoFlaechen"/>
          <xsd:element ref="Strasse"/>
          <xsd:element ref="Gebaeude"/>
        </xsd:choice>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="Bodenbedeckung" type="BodenbedeckungType"
  substitutionGroup="gml:AbstractFeature"/>
<xsd:complexType name="BodenbedeckungType">
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractFeatureType">
      <xsd:sequence>
        <xsd:element name="member" type="BodenbedeckungMemberType"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="gml:AggregationAttributeGroup"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```

## INTERLIS 1-Transferdaten

```
SCNT
Beispiel Transfer-File
////
MTID Beispiel
MODL Beispiel
TOPI Bodenbedeckung
TABL BoFlaechen_Form
OBJE 1
STPT 146.92 174.98
LIPT 138.68 187.51
LIPT 147.04 193.00
LIPT 149.79 188.82
LIPT 158.15 194.31
LIPT 163.64 185.96
LIPT 146.92 174.98
ELIN
OBJE 2
STPT 140.69 156.63
LIPT 118.19 179.82
LIPT 113.00 219.97
LIPT 148.30 228.97
LIPT 186.38 206.82
ELIN
OBJE 3
STPT 186.38 206.82
ARCP 183.26 188.19
LIPT 170.18 176.00
LIPT 140.69 156.63
ELIN
OBJE 4
STPT 186.38 206.82
LIPT 194.26 208.19
ARCP 190.75 185.21
LIPT 174.10 169.00
LIPT 145.08 149.94
LIPT 140.69 156.63
ELIN
ETAB
TABL BoFlaechen
OBJE 10 0 148.20 183.48
OBJE 20 1 168.27 170.85
OBJE 30 2 133.95 206.06
ETAB
TABL Strasse
OBJE 100
STPT 190.26 208.00
ARCP 187.00 186.00
LIPT 173.10 171.00
LIPT 141.08 152.94
ELIN
ETAB
TABL Gebaeude
OBJE 40 148.41 175.96 958 10
ETAB
ETOP
EMOD
ENDE
```

## INTERLIS 2-Transferdaten («INTERLIS-XML»)

```

<?xml version='1.0' encoding='UTF-8'?>
<TRANSFER xmlns="http://www.interlis.ch/INTERLIS2.4">
<HEADERSECTION SENDER="ceis" VERSION="2.4">
  <MODELS>
    <MODEL NAME="Beispiel" VERSION="2016-01-31"
      URI="mailto:ceis@localhost" />
  </MODELS>
</HEADERSECTION>
<DATASECTION>
<Beispiel.Bodenbedeckung BID="itf0">
<Beispiel.Bodenbedeckung.Gebaeude TID="40">
  <PositionHauseingang>
    <COORD><C1>148.41</C1><C2>175.96</C2></COORD>
  </PositionHauseingang>
  <AssNr>958</AssNr><Flaeche REF="10" />
</Beispiel.Bodenbedeckung.Gebaeude>
<Beispiel.Bodenbedeckung.BoFlaechen TID="30">
  <Art>humusiert</Art>
  <Form><SURFACE><BOUNDARY>
    <POLYLINE>
      <COORD><C1>140.69</C1><C2>156.63</C2></COORD>
      <COORD><C1>170.18</C1><C2>176.0</C2></COORD>
      <ARC>
        <C1>186.38</C1><C2>206.82</C2><A1>183.26</A1><A2>188.19</A2>
      </ARC>
      <COORD><C1>148.3</C1><C2>228.97</C2></COORD>
      <COORD><C1>113.0</C1><C2>219.97</C2></COORD>
      <COORD><C1>118.19</C1><C2>179.82</C2></COORD>
      <COORD><C1>140.69</C1><C2>156.63</C2></COORD>
    </POLYLINE>
  </BOUNDARY>
</BOUNDARY>
  <POLYLINE>
    <COORD><C1>146.92</C1><C2>174.98</C2></COORD>
    <COORD><C1>163.64</C1><C2>185.96</C2></COORD>
    <COORD><C1>158.15</C1><C2>194.31</C2></COORD>
    <COORD><C1>149.79</C1><C2>188.82</C2></COORD>
    <COORD><C1>147.04</C1><C2>193.0</C2></COORD>
    <COORD><C1>138.68</C1><C2>187.51</C2></COORD>
    <COORD><C1>146.92</C1><C2>174.98</C2></COORD>
  </POLYLINE>
</BOUNDARY></SURFACE></Form>
</Beispiel.Bodenbedeckung.BoFlaechen>
<Beispiel.Bodenbedeckung.BoFlaechen TID="20">
  <Art>befestigt</Art>
  <Form><SURFACE><BOUNDARY>
    <POLYLINE>
      <COORD><C1>140.69</C1><C2>156.63</C2></COORD>
      <COORD><C1>145.08</C1><C2>149.94</C2></COORD>
      <COORD><C1>174.1</C1><C2>169.0</C2></COORD>
      <ARC>
        <C1>194.26</C1><C2>208.19</C2><A1>190.75</A1><A2>185.21</A2>
      </ARC>
      <COORD><C1>186.38</C1><C2>206.82</C2></COORD>
      <ARC>
        <C1>170.18</C1><C2>176.0</C2><A1>183.26</A1><A2>188.19</A2>
      </ARC>
      <COORD><C1>140.69</C1><C2>156.63</C2></COORD>
    </POLYLINE>
  </BOUNDARY></SURFACE></Form>
</Beispiel.Bodenbedeckung.BoFlaechen>

```



```

<Beispiel.Bodenbedeckung.BoFlaechen TID="10">
  <Art>Gebaeude</Art>
  <Form><SURFACE><BOUNDARY>
    <POLYLINE>
      <COORD><C1>146.92</C1><C2>174.98</C2></COORD>
      <COORD><C1>163.64</C1><C2>185.96</C2></COORD>
      <COORD><C1>158.15</C1><C2>194.31</C2></COORD>
      <COORD><C1>149.79</C1><C2>188.82</C2></COORD>
      <COORD><C1>147.04</C1><C2>193.0</C2></COORD>
      <COORD><C1>138.68</C1><C2>187.51</C2></COORD>
      <COORD><C1>146.92</C1><C2>174.98</C2></COORD>
    </POLYLINE>
  </BOUNDARY></SURFACE></Form>
</Beispiel.Bodenbedeckung.BoFlaechen>
<Beispiel.Bodenbedeckung.Strasse TID="100">
  <Achse>
    <POLYLINE>
      <COORD><C1>190.26</C1><C2>208.00</C2></COORD>
      <ARC>
        <C1>173.10</C1><C2>171.00</C2><A1>187.00</A1><A2>186.00</A2>
      </ARC>
      <COORD><C1>141.08</C1><C2>152.94</C2></COORD>
    </POLYLINE>
  </Achse>
</Beispiel.Bodenbedeckung.Strasse>
</Beispiel.Bodenbedeckung>
</DATASECTION>
</TRANSFER>

```

## GML-Daten

```

<?xml version="1.0" encoding="UTF-8"?>
<ili:TRANSFER gml:id="tid"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:ili="http://www.interlis.ch/ILIGML-2.0/INTERLIS"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.interlis.ch/ILIGML-
2.0/INTERLIS/ili.xsd"
>
<ili:baskets>
<ili2gml:Bodenbedeckung gml:id="bid"
  xmlns:ili2gml="http://www.interlis.ch/ILIGML-2.0/Beispiel"
  xsi:schemaLocation="http://www.interlis.ch/ILIGML-
2.0/Beispiel/beispiel.xsd"
>
  <ili2gml:member>
    <ili2gml:Gebaeude gml:id="x50">
      <ili2gml:PositionHauseingang>
        <gml:Point gml:id="g0" srsName="urn:ogc:def:crs:EPSG::21781"
srsDimension="2">
          <gml:pos>148.41 175.96</gml:pos>
        </gml:Point>
      </ili2gml:PositionHauseingang>
      <ili2gml:AssNr>958</ili2gml:AssNr>
      <ili2gml:Flaeche xlink:href="#x10"/>
    </ili2gml:Gebaeude>
  </ili2gml:member>
  <ili2gml:member>
    <ili2gml:BoFlaechen gml:id="x30">
      <ili2gml:Art>Gebaeude</ili2gml:Art>
      <ili2gml:Form>
        <gml:Polygon gml:id="g2" srsName="urn:ogc:def:crs:EPSG::21781"
srsDimension="2">
          <gml:exterior>
            <gml:Ring>
              <gml:curveMember>
                <gml:Curve gml:id="g21">
                  <gml:segments>
                    <gml:LineStringSegment interpolation="linear">
                      <gml:posList>146.92 174.98 163.64 185.96 158.15
194.31 149.79 188.82 147.04 193 138.68 187.51 146.92
174.98</gml:posList>
                    </gml:LineStringSegment>
                  </gml:segments>
                </gml:Curve>
              </gml:curveMember>
            </gml:Ring>
          </gml:exterior>
        </gml:Polygon>
      </ili2gml:Form>
    </ili2gml:BoFlaechen>
  </ili2gml:member>
  <ili2gml:member>
    <ili2gml:BoFlaechen gml:id="x10">
      <ili2gml:Art>humusiert</ili2gml:Art>
      <ili2gml:Form>
        <gml:Polygon gml:id="g41" srsNa-
me="urn:ogc:def:crs:EPSG::21781" srsDimension="2">
          <gml:exterior>
            <gml:Ring>
              <gml:curveMember>

```



```

        </gml:segments>
      </gml:Curve>
    </gml:curveMember>
  </gml:Ring>
</gml:exterior>
</gml:Polygon>
</ili2gml:Form>
</ili2gml:BoFlaechen>
</ili2gml:member>
<ili2gml:member>
  <ili2gml:Strasse gml:id="x40">
    <ili2gml:Achse>
      <gml:Curve gml:id="g40.1">
        <gml:segments>
          <gml:LineStringSegment interpolation="linear">
            <gml:posList>190.26 208.00</gml:posList>
          </gml:LineStringSegment>
          <gml:Arc numArc="1" interpolation="circularArc3Points">
            <gml:posList>190.26 208.00 187.00 186.00 173.10
171.00</gml:posList>
          </gml:Arc>
          <gml:LineStringSegment interpolation="linear">
            <gml:posList>173.10 171.00 141.08 152.94</gml:posList>
          </gml:LineStringSegment>
        </gml:segments>
      </gml:Curve>
    </ili2gml:Achse>
  </ili2gml:Strasse>
</ili2gml:member>
</ili2gml:Bodenbedeckung>
</ili:baskets>
</ili:TRANSFER>

```

## Anhang D – Mitarbeit & Überprüfung

	Koordinationsorgans für Geoinformation des Bundes GKG
Alain Buogo	Bundesamt für Landestopografie swisstopo
Pasquale Di Donato	Bundesamt für Landestopografie swisstopo
Claude Eisenhut	Eisenhut Informatik AG
Olivier Ertz	Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud HEIG-VD
Michael Germann	infoGrips GmbH
Jens Ingensand	Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud HEIG-VD
Dominic Kottmann	Konferenz der Kantonalen Geoinformationsstellen KK GEO
Thorsten Reitz	wetransform GmbH
Rolf Zürcher	Bundesamt für Landestopografie swisstopo

## Anhang E – Abkürzungen und Glossar

EBNF	Erweiterte Backus-Naur-Form
FMT	INTERLIS 1 Transferformat
GML	Geographic Markup Language
GML-SF	Geographic Markup Language Simple Feature
INTERLIS	Geodatenbeschreibungssprache und Transferformat
ITF	INTERLIS 1 Transferdatei
UTF-16	16-bit Universal Character Set Transformation Format
UTF-8	8-bit Universal Character Set Transformation Format
XML	Extensible Markup Language
XSD	XML Schema Definition
XTF	INTERLIS 2 Transferformat

## Anhang F – Änderungen gegenüber Vorversion

Folgende Anpassungen wurden in dieser Version gegenüber der Vorgängerversion getätigt:

- Das ganze Dokument wurde an die neue Version von INTERLIS (2.4) angepasst
- Die Zusammenfassung wurde entsprechend der geänderten Inhalte angepasst.
- Die Verzeichnisse wurden auf den neusten Stand gebracht.
- Die normativen Referenzen wurden aktualisiert.
- Kapitel 4 wurde entfernt. Der Hauptanwendungsfall wurde im Kapitel 2.3 beschrieben.
- Kapitel 5 wurde entfernt. Wenn nötig, wurde der Text in anderen Kapiteln verbreitet.
- Die Nummerierung der Kapitel wurde auf den neuen Stand gebracht.
- Das ili2c-Modell wurde neu im Anhang B definiert.
- Die Abbildung von DomainDefComplex (Kapitel 6.7) wurde leicht angepasst.
- Das bisherige Kapitel 7.9 wurde in drei neue Kapitel unterteilt: Kap. 6.9 Kodierung von Klassen, Kap. 6.10 Kodierung von Strukturen, Kap. 6.11 Kodierung von Beziehungen.
- Die Kodierung von Strukturen wurde angepasst.
- Die Kodierung von Beziehungen wurde angepasst.
- Die Übersicht der Abbildung der INTERLIS-Typen auf XML-Schema-Typen wurde auf den neuen Stand gebracht.
- Die Kodierung von Aufzählungen wurde angepasst.
- Die Kodierung von Koordinaten, Linienzügen, Einzelflächen, Gebietseinteilungen, Multigeometrien wurde angepasst, um die GML-Basistypen zu verwenden.
- Das Kapitel Kodierung von Linienattributen (6.15.13) wurde neu eingefügt.
- Das Kapitel Kodierung von Multigeometrien (6.15.14) wurde neu eingefügt.
- Das Kapitel Kodierung unterschiedlicher Geometrienattribute innerhalb derselben Klasse (6.15.15) wurde neu eingefügt.
- Das Kapitel 7 (bisher Kapitel 8) wurde auf den neuen Stand gebracht.
- Die Kapitel 8 und 9 (bisher Kapitel 9 und 10) bleiben unverändert.
- Das ili2c-Schema (Anhang B) wurde neu definiert.
- Die Anhänge A und C (bisher B) wurden auf den neuen Stand gebracht.
- Die Anhänge D, E, und F wurden neu eingefügt.