

eCH-0118 Règles de codification GML pour INTERLIS

Nom	Règles de codification GML pour INTERLIS
eCH-nombre	eCH-0118
Catégorie	norme de procédure
Stade	implémenté
Version	2.0
Statut	Approuvé
Date de approbation	2016-09-07
Date de publication	2016-09-09
Remplace version	1.00
Conditions préa- lables	eCH-0118 1.00
Annexes	ili2.xsd, ili2c.xsd
Langues	Allemand (original), Français (traduction)
Auteur(s)	<p>Groupe spécialisé eCH INTERLIS</p> <p><i>Claude Eisenhut, Eisenhut Informatik AG</i></p> <p><i>Michael Germann, infoGrips GmbH</i></p> <p><i>Rolf Zürcher, swisstopo</i></p> <p><i>Pasquale Di Donato, swisstopo</i></p> <p><i>Olivier Ertz, HEIG-VD</i></p> <p><i>Jens Ingensand, HEIG-VD</i></p> <p><i>Dominic Kottmann, KK GEO</i></p> <p><i>Thorsten Reitz, wetransform GmbH</i></p>
Éditeur / Distribu- teur	<p>Association eCH, Mainaustrasse 30, Case postale, 8034 Zürich</p> <p>T 044 388 74 64, F 044 388 71 80</p> <p>www.ech.ch / info@ech.ch</p>

Résumé

Cette norme eCH définit les règles de codification afin d'obtenir un format de transfert GML à partir d'un modèle de données INTERLIS. INTERLIS est une norme suisse pour la description des modèles de données conceptionnels et pour l'échange de données basé sur un modèle. Le GML est une norme pour la codification des données géographiques basées sur XML.

En premier lieu, les concepts fondamentaux des langages INTERLIS et GML utilisées seront définis, ainsi que les règles de codification et les cas d'application de XML (chapitre 4).

Suivi d'une représentation des différences entre la codification GML et la codification par XML spécifique à INTERLIS (chapitre 5).

La partie principale de ce standard s'occupe de la définition détaillée et complète des règles de codification de schéma (chapitre 6) ainsi que des règles de codification d'instance (chapitre 7).

L'annexe A comporte le schéma XML de base prédéfini pour la codification GML d'INTERLIS. En annexe B figure le schéma ili2c. Les deux documents sont disponibles à part de la présente norme, sous forme de schéma XML.

La présente spécification se termine par un exemple simple d'application (annexe C).

Table des matières

1	Statut du document	6
2	Introduction	7
	2.1 Aperçu	7
	2.2 Conditions préalables	7
	2.3 Restrictions et objectifs généraux	7
3	Références normatives	9
4	Notions, concepts	10
	4.1 INTERLIS	10
	4.2 Geography Markup Language (GML)	10
	4.3 Règles de codification.....	11
5	Différences sur les règles de codification INTERLIS- XML	12
	5.1 Règles de codification.....	12
	5.2 Structure d'un transfert	12
	5.3 Lecture de modèles élargis / traduits	12
	5.4 Jeu de caractères	12
	5.5 Types de transferts.....	13
6	Règles de codification – schéma	14
	6.1 Remarque préliminaire	14
	6.2 Codification des caractères	14
	6.3 Règles générales.....	14
	6.4 Représentation des éléments de modèle INTERLIS sur des noms de schémas XML 14	
	6.5 Représentation de types de base	15
	6.6 Structure générale du schéma XML	15
	6.7 Codification de domaines de valeurs	16
	6.8 Codification de thèmes	17
	6.9 Codification de classes	19
	6.10 Codification de structures	19
	6.11 Codification des associations.....	20
	6.11.1 Références incorporées	21

6.11.2	Table intermédiaire.....	21
6.11.3	Exemples	23
6.11.3.1	Références incorporées, sans table intermédiaire.....	23
6.11.3.2	Références incorporées et table intermédiaire	24
6.11.3.3	Aucune référence incorporée, que table intermédiaire.....	25
6.11.3.4	Références incorporées partiellement et table intermédiaire	26
6.12	Codification des vues.....	28
6.13	Codification des définitions de graphique	28
6.14	Codification des attributs	28
6.15	Représentation des types INTERLIS sur des types de schéma XML.....	31
6.15.1	Aperçu.....	31
6.15.2	Codification des chaînes de caractères	31
6.15.3	Codification d'énumérations	32
6.15.4	Codification de types de données numériques	36
6.15.5	Codification de domaines formatés de valeurs	37
6.15.6	Codification des contenants	38
6.15.7	Codification de types de classes	38
6.15.8	Codification de types de chemin d'attributs	38
6.15.9	Codification des coordonnées	39
6.15.10	Codification des polylignes.....	39
6.15.11	Codification des surfaces indépendantes.....	40
6.15.12	Codification d'une partition de territoire.....	41
6.15.13	Codification de LINE ATTRIBUTES	41
6.15.14	Codification de géométries multiples.....	43
6.15.15	Codification de différents attributs de géométrie au sein de la même classe 43	
6.15.16	Codification de types pour l'identification d'objet.....	44
6.15.17	Codification d'autres formes de portions de courbes.....	45
7	Règles de codification d'instance.....	46
7.1	Codification de plusieurs contenants (TOPIC) dans un fichier de transfert	46
7.2	Codification des objets et éléments de structure.....	46
7.3	Codification d'identifications de contenant et d'objet.....	46
7.4	Codification d'associations.....	47

7.5 Codification de géométries	47
7.6 Codification de listes de code	47
8 Exclusion de responsabilité – Droits de tiers	48
9 Droits d’auteur.....	48
Annexe A – Schéma de base prédéfini	49
Annexe B - Schéma de base ili2c.....	51
Annexe C - Exemple.....	52
Modèle INTERLIS 2	52
Modèle INTERLIS 1	53
Schéma d’application GML.....	54
INTERLIS 1-Données de transfert.....	56
INTERLIS 2-Données de transfert («INTERLIS-XML»).....	57
Annexe D – Collaboration & vérification	62
Annexe E – Abréviations et glossaire.....	62
Annexe F – Modifications par rapport à la version précédente	63

1 Statut du document

Approuvé: Le document a été approuvé par le comité d'experts. Il a pouvoir normatif pour le domaine d'utilisation défini dans le domaine de validité donné.

2 Introduction

2.1 Aperçu

Ce document définit les règles pour obtenir un schéma d'application GML à partir d'un modèle INTERLIS.

2.2 Conditions préalables

Ce document suppose des connaissances préalables sur INTERLIS et GML. Lors d'une étude approfondie, il faut se reporter aux sources [1], [2] et [3]. Toutefois, le texte est complété par des exemples, de sorte que le contenu peut aussi être compris par des personnes non expertes dans ce domaine.

2.3 Restrictions et objectifs généraux

- Il faut viser une représentation du modèle la plus précise possible.
- Il ne doit y avoir aucune perte de données face à un transfert de celles-ci avec INTERLIS 1 ou 2 (ITF ou XTF).
- Le schéma d'application GML obtenu peut être utilisé par des logiciels conformes au GML.
- Le schéma d'application GML obtenu ne contient pas toutes les caractéristiques de description du modèle INTERLIS car GML ne connaît pas ces possibilités. Au niveau du modèle (description des données), on observe donc une perte, c.à.d. que le schéma d'application GML ne peut pas remplacer le modèle INTERLIS.
- Le schéma d'application GML obtenu se destine aux géoservices Web (notamment WFS), mais il est moins adapté pour un transfert de données de masse (à cause de la manière de codification des associations).
- Ce document utilise INTERLIS 2.3, INTERLIS 2.4 et GML 3.2.1. Étant donné qu'INTERLIS 1 peut plus ou moins être traduit à 1:1 en INTERLIS 2.3, ce document est également valable pour INTERLIS 1.

Remarque: Il n'y a aucune perte de données, mais les capacités du XTF (Format de transfert INTERLIS 2-XML) qui ne sont pas présentes dans GML (par ex. transfert incrémentiel) ne sont pas reconstituées. Le format de transfert résultant ne représente dans ce cas pas un substitut intégral pour XTF.

Dans la mesure du possible, les types GML de base ne sont pas élargis afin de permettre une utilisation des données la plus large possible.

Dans la mesure du possible, le schéma d'application GML résultant est conforme au GML Simple Feature (GML-SF) [8] afin de permettre une utilisation des données la plus large possible. Si le modèle INTERLIS comprend des éléments qui ne peuvent pas être représentés à 1:1 en GML-SF (par ex. arc de cercle), le schéma d'application GML obtenu n'est pas conforme à GML-SF.

3 Références normatives

- [1] eCH, 2016. *eCH-0031 Géoinformation: INTERLIS 2- Manuel de référence Version 2.0*,
<http://www.ech.ch> → eCH Documents → Normes → eCH-0031
- [2] eCH, 2006. *eCH-0031 Géoinformation: INTERLIS 2- Manuel de référence Version 1.0*,
<http://www.ech.ch> → eCH Documents → Normes → eCH-0031
- [3] Open Geospatial Consortium OGC, 2007. *OpenGIS Geography Markup Language (GML) Encoding Standard*. OGC 07-036, Version 3.2.1.
Site http://portal.opengeospatial.org/files/?artifact_id=20509
en ligne : 2016-01-25
- [4] World Wide Web Consortium W3C, 2004. *XML Schema 1.1*. Norme.
Site <http://www.w3.org/XML/Schema>
en ligne : 2016-01-25
- [5] ISO/IEC 14977, 1996. *Information technology – Syntactic metalanguage – Extended BNF*. Norme internationale.
- [6] ISO/IEC 10646:2003, 2003. *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*. Norme internationale.
- [7] eCH INTERLIS, 2010. *eCH-0117: Méta-attributs pour modèles INTERLIS version 1.0*
<http://www.ech.ch> → eCH Documents → Normes → 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
En ligne : 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
En ligne : 2016-01-25

4 Notions, concepts

4.1 INTERLIS

INTERLIS est un langage conceptionnel de description des données et est utilisé dans le domaine de la modélisation et de l'échange de géodonnées [1, 2].

Le manuel de référence INTERLIS [1, 2] se compose de deux parties:

- un langage de description des données
- des règles de codage.

Si les règles de codification XML sont appliquées sur un modèle de données défini à l'aide du langage de description des données, on obtient alors un format de transfert adapté au modèle de données.

En ce qui concerne les règles de codification XML, une alternative est définie dans le présent document.

L'annexe C présente un exemple de modèle de données dans INTERLIS avec un fichier de transfert conforme à ce modèle de données.

4.2 Geography Markup Language (GML)

Le GML [3] est un ensemble de schémas XML de base [4] et un ensemble de règles définissant la façon dont ces schémas doivent être utilisés dans des schémas propres (appelés *schémas d'application*) pour la définition de formats de transfert. À partir des connaissances des schémas de base et des règles pour les schémas d'application, il est ainsi possible d'obtenir aussi une description des données en découlant.

Remarque: Comme les règles pour les schémas d'application GML sont définies, on ne connaît pas seulement la suite autorisée des éléments XML (grâce au schéma XML) mais on sait également quels éléments XML représentent des classes et quels éléments sont les attributs de ces classes. Cette description est moins précise que celle effectuée au moyen d'INTERLIS, mais cela ne constitue pas un aspect primordial dans ce document.

Le fichier INTERLIS 1-FMT est également une description de format. Contrairement à un schéma XML, il ne s'agit aucunement d'une description formelle ici.

L'annexe C présente un exemple de schéma d'application avec un fichier de transfert conforme à ce schéma.

Remarque: Les schémas GML 3.1 ne valident partiellement pas! Il ne s'agit pas seulement d'un problème provenant de GML;

voir également <http://schemas.opengis.net/gml/3.1.1/readme.txt>

Dans la spécification GML 3.2 [3], on trouve à ce sujet:

«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, too.»

4.3 Règles de codification

Les règles de codification définissent comment, à partir d'un modèle de données technique et avec l'application de certaines règles, il est possible d'obtenir le format de transfert (transfert basé sur un modèle). Ce faisant, les règles sont donc formulées de sorte à pouvoir être appliquées sur n'importe quel modèle de données.

5 Différences sur les règles de codification INTERLIS- XML

5.1 Règles de codification

Le manuel de référence INTERLIS définit les règles de codification des instances. Dans le présent document, les règles de codification de schéma sont définies et les instances sont obtenues à partir du schéma ainsi créé.

5.2 Structure d'un transfert

Le manuel de référence INTERLIS divise un transfert en générique et en domaine de données. Ce document ne traite pas l'aspect générique, étant donné que cela serait interprété en tant que donnée par un logiciel conforme à GML (au lieu de métadonnée pour le transfert).

5.3 Lecture de modèles élargis / traduits

Le manuel de référence INTERLIS 2.3 [2] définit un tableau de représentation en tant que partie du générique, de sorte qu'un programme de lecture qui a été programmé ou configuré pour un certain modèle de données puisse lire les données des extensions (ou des traductions) de ce modèle de données sans connaître les définitions de modèle élargis. Dans ce document, on renoncera à un tel tableau de représentation.

Un programme de lecture doit donc connaître le modèle de données élargi afin de pouvoir ignorer correctement les extensions.

Pour INTERLIS 2.4 [1], aucune règles spéciales ne sont nécessaires car les tableaux de représentation ont été abrogés en INTERLIS 2.4.

Pour les modèles de données INTERLIS traduits (TRANSLATION OF), aucun schéma d'application GML ne sera créé, c.à.d. que lors d'une traduction, le format de transfert n'est pas modifié.

5.4 Jeu de caractères

Le manuel de référence INTERLIS définit un jeu de caractères restreint (annexe D dans le manuel de référence) [1].

Le présent document définit l'étendue conforme des caractères Unicode. Un jeu de caractères restreint doit être défini dans le cadre d'une communauté de transfert.

5.5 Types de transferts

Le manuel de référence INTERLIS définit trois types de transfert: «FULL», «INITIAL» ou «UPDATE». En outre, il est possible de transmettre des indications sur la consistance: «COMPLETE», «INCOMPLETE», «INCONSISTENT», «ADAPTED».

Les règles de codification définies dans le présent document, s'appliquent au transfert d'objets individuels ou des jeux de données complets.

Remarque: Le transfert incrémentiel ne peut pas être réalisé avec seulement des logiciels GML standard car GML ne définit aucune sémantique de ce type. Les indications de consistance pourraient être codées en tant que métadonnées GML. Dans ce document le transfert incrémentiel n'est pas reconstitué.

6 Règles de codification – schéma

6.1 Remarque préliminaire

Pour la formalisation des règles de dérivation de format de transfert, on utilise la notation EBNF [5] qui a été introduite dans le chapitre 2.1 du manuel de référence INTERLIS [1].

6.2 Codification des caractères

Les règles XML s'appliquent: jeu de caractères Unicode et UTF-8 (recommandé) ou UTF-16 en tant que codification de caractères [6].

6.3 Règles générales

Le schéma XML créé [4] peut contenir autant de commentaires (<!-- ... -->) ou d'éléments supplémentaires `xsd:annotation` que l'on souhaite.

6.4 Représentation des éléments de modèle INTERLIS sur des noms de schémas XML

Chaque modèle de données INTERLIS contient un propre espace nominal de schéma XML. Pour les définitions de type et d'élément XML, on utilise les noms INTERLIS non qualifiés. En cas de conflits, on utilise les noms qualifiés. Les noms du niveau MODEL sont prioritaires par rapport aux noms du niveau TOPIC. En cas de conflits entre les noms de différents TOPIC, le nom du premier TOPIC défini dans le modèle a la priorité.

Exemple INTERLIS	Nom d'élément GML
<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 Représentation de types de base

Pour les types de base indépendant du modèle et provenant d'INTERLIS, un schéma de base (annexe A) est défini.

6.6 Structure générale du schéma XML

Chaque modèle INTERLIS est représenté dans un schéma XML dans un document de schéma.

```

ModelDef = '<xsd:schema
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns="%ModelNamespaceIdentifieur%"
    targetNamespace="%ModelNamespaceIdentifieur%"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    xmlns:gml="http://www.opengis.net/gml/3.2"
    { 'xmlns:%nsprefix% namespace=
        "%Imported-ModelNamespaceIdentifieur%" ' }
    '>'
    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-ModelNamespaceIdentifieur%"/>'.

ModelNamespaceIdentifieur = StdModelNamespaceIdentifieur
    | %MetaAttributeValue%.

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

```

Le modèle ili2c est défini dans l'annexe B.

L'identificateur d'espace nominaux XML (règle «ModelNamespaceIdentifieur») résulte du nom de modèle ou peut être défini au moyen du méta-attribut «ili2.iligml20.namespaceName» [7].

Exemple	
INTERLIS	<pre> INTERLIS 2.4; MODEL ModelA (de) AT "mailto:ceis@localhost" VERSION "2016-01-31" = IMPORTS Units; END ModelA. </pre>
GML	<pre> <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"> <xsd:annotation> <xsd:appinfo source="http://www.interlis.ch/ili2c"> <ili2:model>ModelA</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"/> <xsd:import namespace ="http://www.interlis.ch/ILIGML-2.0/Units"/> </xsd:schema> </pre>

6.7 Codification de domaines de valeurs

Les définitions des domaines de valeurs sont représentées comme suit:

```

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>'.

```

Selon la règle de représentation du type de données INTERLIS correspondant (voir chapitre 6.15), c'est la règle «DomainDefComplex» ou la règle «DomainDefSimple» qui sera appliquée.

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

Les domaines de valeurs géométriques ne seront pas codifiés. Les restrictions des types géométriques ne peuvent effectivement pas être définies dans le schéma d'application GML.

6.8 Codification de thèmes

Dans INTERLIS, les thèmes présentent deux fonctions: d'un côté, ils définissent un propre espace nominal (distinct de l'espace nominal du modèle) et, de l'autre côté, ils définissent un contenant. La caractéristique en tant qu'espace nominal n'est pas représentée. La caractéristique en tant que description de contenant est représentée comme suit:

```

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>'.

```

Exemple	
INTERLIS	<pre> TOPIC Bodenbedeckung = CLASS BoFlaechen = END BoFlaechen; CLASS Gebaeude = END Gebaeude; END Bodenbedeckung; </pre>
GML	<pre> <xsd:complexType name="BodenbedeckungMemberType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureMemberType"> <xsd:sequence> <xsd:choice> <xsd:element ref="BoFlaechen"/> <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> </pre>
GML Instanz	<pre> <Bodenbedeckung gml:id="b1"> <member> <BoFlaechen gml:id="o1"> ... </BoFlaechen> </member> <member> <BoFlaechen gml:id="o2"> ... </BoFlaechen> </member> </Bodenbedeckung> </pre>

6.9 Codification de classes

Des classes sont représentées comme suit:

```
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>'
```

Exemple

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

6.10 Codification de structures

Les structures sont représentées comme suit:

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

6.11 Codification des associations

Pour la codification des associations on distingue deux méthodes :

- La codification par des références incorporées
- La codification par des tables intermédiaires

Selon la définition de l'ASSOCIATION du modèle, seulement une de ces méthodes ou une combinaison des deux méthodes est utilisé.

Les éléments suivants d'une relation influencent, individuellement ou en combinaison, le choix de la méthode :

- EXTERNAL: références externes (EXTERNAL à RoleDef)
- ATTRIBUTE: sont des attributs disponibles ou pas
- FINAL: l'association est définie comme FINAL, ou pas
- ORDERED: association ordonnée
- OID: l'instance d'association a une identification d'objet

6.11.1 Références incorporées

La codification par une référence incorporée est utilisée pour des rôles (RoleDef) qui ne possèdent pas la caractéristique EXTERNAL. Si le rôle opposé est ordonné (ORDERED), l'ordre des références dans l'instance est signifiant (ce n'est pas visible dans le XSD).

```
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% résulte du nom du rôle opposé. %ClassName% est la classe cible du rôle opposé.

%minCardinality% et %maxCardinality% résultent de la cardinalité du rôle opposé.

La codification par une référence incorporée se fait aussi quand une table intermédiaire supplémentaire existe.

6.11.2 Table intermédiaire

La codification par une table intermédiaire est faite si au moins une des conditions suivantes est remplie :

- L'association a des attributs
- L'association n'est pas FINAL
- L'association est une extension (EXTENDED ou EXTENDS)
- Les instances de relation ont une identification d'objet
- Tous les rôles sont EXTERNAL
- Un rôle est EXTERNAL et le rôle opposé est ORDERED
- La relation a plus que deux rôles

Règle spéciale : Si la relation n'est pas FINAL, n'a pas d'extension, n'a pas d'attributs, n'a que deux rôles, aucun des rôles n'est EXTERNAL et pas d'identification d'objet est nécessaire, les définitions pour une table intermédiaire seront alors générées dans le XSD (comme modèle pour une extension qui figure éventuellement dans un autre modèle), mais aucune instance de cette table ne sera créé au niveau XML.

```
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>'.

```

Concernant l'ordre des attributs, des rôles et des rôles incorporés au sein de l'association, les principes suivants sont applicables : ce sont d'abord les rôles qui sont codés, puis tous les attributs et ensuite tous les rôles incorporés. Les attributs et les rôles sont codés dans le fichier de modèle selon leur ordre de définition. Les rôles incorporés sont triés par ordre alphabétique croissant. Un attribut spécialisé ou un rôle spécialisé (avec le mot clé «EXTENDED») n'est pas généré.

Si l'association n'est pas une extension d'une autre association, `gml:AbstractFeatureType` est utilisé comme type de base ou `gml:AbstractFeature` est utilisé comme élément de base. Si l'association est une extension, le type de base ou l'élément de base qui a été généré spécifiquement est utilisé. La structure d'hérédité de l'association est par conséquent représentée dans le schéma XML.

Notice : Le nom (de classe) est créé en rattachant les noms de rôles séparés pour les associations n'ayant pas de nom explicite (par ex. `%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>'.

```

Quand la relation est ordonnée, un attribut XML supplémentaire («ORDER_POS»; défini en annexe A) est nécessaire au `gml:ReferenceType`.

Il n'est pas toujours clair si la référence indique un objet du même fichier de transfert (sauf si la référence est un document d'identificateurs de fragment XML relatifs).

6.11.3 Exemples

6.11.3.1 Références incorporées, sans table intermédiaire

<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> <xsd:element name="ClassA" type="ClassAType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="ClassAType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> <xsd:element name="b" maxOccurs="unbounded" type="gml:ReferenceType"> <xsd:annotation> <xsd:appinfo> <gml:targetElement>ClassB</gml:targetElement> </xsd:appinfo> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> <xsd:element name="ClassB" type="ClassBType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="ClassBType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> <xsd:element name="a" maxOccurs="unbounded" type="gml:ReferenceType"> <xsd:annotation> <xsd:appinfo> <gml:targetElement>ClassA</gml:targetElement> </xsd:appinfo> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>
<p>GML Instanz</p>	<pre> <ClassA gml:id="o1"> <b xlink:href="#o2"/> </ClassA> <ClassB gml:id="o2"> <a xlink:href="#o1"/> </ClassB> </pre>

6.11.3.2 Références incorporées et table intermédiaire

<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> <xsd:element name="ClassA" type="ClassAType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="ClassAType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> <xsd:element name="b" maxOccurs="unbounded" type="gml:ReferenceType"> <xsd:annotation> <xsd:appinfo> <gml:targetElement>ClassB</gml:targetElement> </xsd:appinfo> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> <xsd:element name="ClassB" type="ClassBType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="ClassBType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> <xsd:element name="a" maxOccurs="unbounded" type="gml:ReferenceType"> <xsd:annotation> <xsd:appinfo> <gml:targetElement>ClassA</gml:targetElement> </xsd:appinfo> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> <xsd:element name="a2b" type="a2bType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="a2bType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> <xsd:element name="a" type="gml:ReferenceType"> <xsd:annotation> <xsd:appinfo> <gml:targetElement>ClassA</gml:targetElement> </xsd:appinfo> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>

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

6.11.3.3 Aucune référence incorporée, que table intermédiaire

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> <xsd:element name="ClassA" type="ClassAType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="ClassAType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> <xsd:element name="ClassB" type="ClassBType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="ClassBType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> <xsd:element name="a2b" type="a2bType" </pre>

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

6.11.3.4 Références incorporées partiellement et table intermédiaire

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> <xsd:element name="ClassA" type="ClassAType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="ClassAType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> <xsd:element name="ClassB" type="ClassBType" substitutionGroup="gml:AbstractFeature"/> </pre>

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

6.12 Codification des vues

Pour la codification des vues, se reporter au chapitre 3.2.3 dans le manuel de référence INTERLIS [1]. Seuls les attributs qui sont indiqués dans la vue de manière explicite sous «ATTRIBUTE» ou de manière implicite avec «ALL OF», sont transmis en tant qu'attributs de l'objet de vue.

6.13 Codification des définitions de graphique

Pour chaque définition de graphique, les classes de signature («Sign-ClassRef») référencées par la définition de graphique sont transmises lors du transfert. Les instances d'objet des classes de signature sont créées par l'exécution des définitions de graphique sur un ensemble concret de données d'entrée. Ce faisant, les paramètres sont codés comme des attributs.

6.14 Codification des attributs

Chaque attribut d'une instance d'objet (incluant les attributs complexes tels que «STRUCTURE», «LIST OF», «BAG OF», etc.) est codé comme suit.

En outre, les règles spéciales définies dans le chapitre 6.15.13 (LINE ATTRIBUTES) et le chapitre 6.15.15 (plusieurs attributs de géométrie dans la même classe) s'appliquent.

```

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>'.
```

Exemple

<p>INTERLIS</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>GML</p>	<pre><xsd:simpleType name="Prozent100"> <xsd:restriction base="xsd:integer"> <xsd:minInclusive value="0"/> <xsd:maxInclusive value="100"/> </xsd:restriction> </xsd:simpleType> <xsd:element name="Koernung" type="KoernungType" substitutionGroup="gml:AbstractObject"/> <xsd:complexType name="KoernungType"> <xsd:sequence> <xsd:element name="Ton" type="Prozent100" minOccurs="0"/> <xsd:element name="Schluff" type="Prozent100" minOccurs="0"/> <xsd:element name="Sand" type="Prozent100" minOccurs="0"/> </xsd:sequence> </xsd:complexType> <xsd:element name="Profil" type="ProfilType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="ProfilType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> <xsd:element name="Lage" minOccurs="0" type="gml:PointPropertyType"> </xsd:element> <xsd:element name="KoernungsklasseOberboden"> <xsd:complexType> <xsd:sequence> <xsd:element ref="Koernung"/> </xsd:sequence> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre>

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

6.15 Représentation des types INTERLIS sur des types de schéma XML

6.15.1 Aperçu

INTERLIS	Codage	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	Classe supplémentaire	Complex

6.15.2 Codification des chaînes de caractères

Les domaines de valeurs / attributs de type de base «TEXT» sont codés en tant que `xsd:normalizedString`; «MTEXT» en tant que `xsd:string`; «NAME» en tant que `xsd:token` et «URI» en tant que `xsd:anyURI`. Dans le cas où ils sont dérivables du modèle, on code aussi `xsd:restriction`.

Exemple

INTERLIS	DOMAIN
----------	--------

	<pre> TestText = TEXT; TestTextLen = TEXT*10; TestMText = MTEXT; TestMTextLen = MTEXT*10; TestURI = URI; TestName = NAME; </pre>
GML	<pre> <xsd:simpleType name="TestText"> <xsd:restriction base="xsd:normalizedString"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="TestTextLen"> <xsd:restriction base="xsd:normalizedString"> <xsd:maxLength value="10"/> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="TestMText"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="TestMTextLen"> <xsd:restriction base="xsd:string"> <xsd:maxLength value="10"/> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="TestURI"> <xsd:restriction base="xsd:anyURI"/> </xsd:simpleType> <xsd:simpleType name="TestName"> <xsd:restriction base="xsd:token"> <xsd:maxLength value="255"/> <xsd:pattern value="[a-zA-Z][a-zA-Z0-9_]*"/> </xsd:restriction> </xsd:simpleType> </pre>

6.15.3 Codification d'énumérations

Les énumérations peuvent être affinées et/ou élargies dans INTERLIS. Exemples :

```

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

```

Étant donné que les schémas XML ne permettent ni d'affiner ni d'élargir les énumérations, une énumération INTERLIS est reproduite en énumération de schéma XML lorsque le domaine de valeurs ou l'attribut est marqué en tant que «FINAL» et n'est pas une extension d'un autre domaine de valeurs. Sinon, l'énumération est reproduite en tant que `gml:CodeType`.

Concernant l'identificateur de listes de codes (`codeSpace`) des différentes valeurs d'énumération, l'identificateur d'espace nominal du modèle (`%StdModelNamespaceIdentifier%`; voir chapitre 7.6), complété par le nom de la définition du domaine de valeurs en question, peut être utilisé (exemple: «`http://www.interlis.ch/ILIGML-2.0/ModelA/Farbe`»). Si l'énumération est définie dans le cadre d'un attribut, alors

l'identificateur de l'espace nominal du modèle est complété par le nom de classe en question et le nom d'attribut.

(Exemple: «<http://www.interlis.ch/ILIGML-2.0/ModelA/Person/HaarFarbe>»).

Le méta-attribut `ili2.iligml20.identifieurCodeSpace` permet de définir l'identificateur des listes de codes (la valeur pour l'attribut XML `gml:codeSpace` dans l'instance GML) par domaine de valeurs.

L'énumération (la liste des valeurs d'énumération possibles) est générée par fichier de modèle en tant que `gml:Dictionary` dans un document d'instance GML supplémentaire. Ce document d'instance GML est établi selon la règle suivante :

```

CodelistInstanceDoc = '<?xml version='1.0' encoding='UTF-8'?>'
  '<gml:Dictionary xmlns:gml="http://www.opengis.net/gml/3.2"
gml:id="%AnyGmlId%">'
  '<gml:identifieur codeSpace=
    "%AnyURI%"
    >' %AnyCode% '</gml:identifieur>'
  { '<gml:dictionaryEntry>'
    CodelistInstance
    '</gml:dictionaryEntry>' }
  '</gml:Dictionary>'.
CodelistInstance = '<gml:Dictionary
gml:id="%QualifiedEnumTypeDefName%">'
  '<gml:identifieur codeSpace=
    "%ModelNamespaceIdentifier%"
    >' %QualifiedEnumTypeDefName% '</gml:identifieur>'
  { CodelistEntry }
  '</gml:Dictionary>'.
CodelistEntry = '<gml:dictionaryEntry>'
  '<gml:Definition gml:id="%AnyGmlId%">'
  '<gml:identifieur codeSpace=
    "%CodespaceIdentifier%"
    >' %EnumValue% '</gml:identifieur>'
  '</gml:Definition>'
  '</gml:dictionaryEntry>'.
  
```

`AnyURI` est une URI valable, à choix par l'auteur de l'instance

`AnyCode` est une valeur de code, à choix par l'auteur de l'instance

`AnyGmlId` est une valeur `gml:id` valable, à choix par l'auteur de l'instance

`ModelNameSpaceIdentifier` est l'URI du modèle dans lequel l'énumération a été définie (selon le chapitre 6.6).

`QualifiedEnumTypeDefName` est le nom Interlis qualifié du domaine ou de l'attribut qui définit l'énumération.

`CodespaceIdentifier` est l'URI de l'énumération issu des explications ci-dessus ou selon le méta-attribut `ili2.iligml20.identifieurCodeSpace`.

Exemple	
<p>INTERLIS</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>GML</p>	<pre> <xsd:simpleType name="FarbeFinal"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="rot"/> <xsd:enumeration value="gelb"/> <xsd:enumeration value="gruen"/> </xsd:restriction> </xsd:simpleType> <xsd:complexType name="Farbe"> <xsd:restriction base="gml:CodeType"/> </xsd:complexType> <xsd:complexType name="FarbePlus"> <xsd:restriction base="gml:CodeType"/> </xsd:complexType> <xsd:element name="Fahrzeug" type="FahrzeugType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="FahrzeugType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> <xsd:element name="CarrosserieFarbe" type="Farbe" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> <xsd:element name="Auto" type="AutoType" substitutionGroup="Fahrzeug"/> <xsd:complexType name="AutoType"> <xsd:annotation> <xsd:appinfo> <ili2c:extendedCodelistAttr> <ili2c:extendedAttribute>CarrosserieFarbe </ili2c:extendedAttribute> <ili2c:codelist>FarbePlus</ili2c:codelist> </ili2c:extendedCodelistAttr> </xsd:appinfo> </xsd:annotation> <xsd:complexContent> <xsd:extension base="FahrzeugType"> <xsd:sequence> <!-- CarrosserieFarbe ist schon in FahrzeugType definiert --> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>

	<pre> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>
<p>Instance GML</p>	<pre> <Fahrzeug gml:id="o1"> <CarroserieFarbe>rot<CarroserieFarbe> </Fahrzeug> <Auto gml:id="o2"> <CarroserieFarbe>rot.dunkel<CarroserieFarbe> </Auto> </pre>
<p>Instance GML de l'énumération</p>	<p>Exemple: Farbe et FarbePlus</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <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"> <identifieur codeSpace="http://www.interlis.ch">ModelA.ili</identifieur> <dictionaryEntry> <gml:Dictionary gml:id="ModelA.Farbe"> <identifieur codeSpace="http://www.interlis.ch/ILIGML-2.0/ModelA">ModelA.Farbe</identifieur> <dictionaryEntry> <gml:Definition gml:id="f1"> <gml:identifieur codeSpace="http://www.interlis.ch/ILIGML-2.0/ModelA/Farbe">rot</gml:identifieur> </gml:Definition> </dictionaryEntry> <dictionaryEntry> <gml:Definition gml:id="f2"> <gml:identifieur codeSpace="http://www.interlis.ch/ILIGML-2.0/ModelA/Farbe">gelb</gml:identifieur> </gml:Definition> </dictionaryEntry> <dictionaryEntry> <gml:Definition gml:id="f3"> <gml:identifieur codeSpace="http://www.interlis.ch/ILIGML-2.0/ModelA/Farbe">gruen</gml:identifieur> </gml:Definition> </dictionaryEntry> </gml:Dictionary> </dictionaryEntry> <dictionaryEntry> <gml:Dictionary gml:id="ModelA.FarbePlus"> <identifieur codeSpace="http://www.interlis.ch/ILIGML-2.0/ModelA">ModelA.FarbePlus</identifieur> <dictionaryEntry> <gml:Definition gml:id="f10"> <gml:identifieur codeSpace="http://www.interlis.ch/ILIGML-2.0/ModelA/FarbePlus">rot.dunkel</gml:identifieur> </gml:Definition> </dictionaryEntry> <dictionaryEntry> <gml:Definition gml:id="f11"> <gml:identifieur </pre>

	<pre> codeSpace="http://www.interlis.ch/ILIGML- 2.0/ModelA/FarbePlus">rot.hell</gml:identifie r> </gml:Definition> </dictionaryEntry> <dictionaryEntry> <gml:Definition gml:id="f12"> <gml:identifie r codeSpace="http://www.interlis.ch/ILIGML- 2.0/ModelA/FarbePlus">gelb</gml:identifie r> </gml:Definition> </dictionaryEntry> <dictionaryEntry> <gml:Definition gml:id="f13"> <gml:identifie r codeSpace="http://www.interlis.ch/ILIGML- 2.0/ModelA/FarbePlus">gruen</gml:identifie r> </gml:Definition> </dictionaryEntry> </gml:Dictionary> </dictionaryEntry> </gml:Dictionary> </pre>
--	---

Dans les deux cas, les valeurs d'énumération sont codées comme suit :

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

Pour la codification des valeurs d'énumération (indépendamment du fait que le domaine de valeurs comprenne seulement les feuilles ou les nœuds également), la syntaxe pour les constantes d'énumération (règle «EnumValue») est appliquée. Ce faisant, le caractère «#» est supprimé.

Les types prédéfinis d'orientation de texte, «HALIGNMENT» et «VALIGNMENT», sont définis dans l'annexe A. Pour le type «BOOLEAN», le type de schéma XML correspondant `xsd:boolean` est utilisé.

6.15.4 Codification de types de données numériques

Les types de données numériques sont codés en fonction de la valeur inférieure et de la valeur supérieure, en tant que `xsd:integer`, `xsd:decimal` ou `xsd:double`. Dans le cas où l'on peut obtenir une dérivation à partir du modèle, on effectue également la codification `xsd:restriction`.

Exemple	
INTERLIS	<pre> DOMAIN TestInt = 1 .. 10; TestDec = 1.0 .. 10.0; TestDouble = 0.123e1 .. 0.234e1; </pre>
GML	<pre> <xsd:simpleType name=" TestInt"> <xsd:restriction base="xsd:integer"> <xsd:minInclusive value="1"/> <xsd:maxInclusive value="10"/> </xsd:restriction> </pre>

	<pre> </xsd:simpleType> <xsd:simpleType name="TestDec"> <xsd:restriction base="xsd:decimal"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="TestDouble"> <xsd:restriction base="xsd:double"> <xsd:minInclusive value="1.23"/> <xsd:maxInclusive value="2.34"/> </xsd:restriction> </xsd:simpleType> </pre>
--	--

6.15.5 Codification de domaines formatés de valeurs

Les domaines formatés de valeurs qui réfèrent «INTERLIS.XMLDate», «INTERLIS.XMLTime» ou «INTERLIS.XMLDateTime», sont représentés dans les types de date de schéma XML correspondants (xsd:date, xsd:time, xsd:dateTime). Tous les autres domaines formatés de valeurs sont représentés dans un xsd:string.

Exemple	
INTERLIS	<pre> STRUCTURE GregorianCalendar = Year: 1582 .. 2999; SUBDIVISION Month: 1 .. 12; SUBDIVISION Day: 1 .. 31; END GregorianCalendar; 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 GregorianCalendar (Year "Y" Month "M" Day "D"); </pre>
GML	<pre> <xsd:simpleType name="BuchungsDatum"> <xsd:restriction base="xsd:date"/> </xsd:simpleType> <xsd:simpleType name="StartZeit"> <xsd:restriction base="xsd:time"/> </xsd:simpleType> <xsd:simpleType name="MessZeitpunkt"> <xsd:restriction base="xsd:dateTime"/> </xsd:simpleType> <xsd:simpleType name="EigenesDatum"> <xsd:restriction base="xsd:string"/> </xsd:simpleType> </pre>

6.15.6 Codification des contenants

La variante XML du type «BLACKBOX» est codée en tant que `xsd:any` et la variante binaire en tant que `xsd:base64Binary`.

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

6.15.7 Codification de types de classes

Le type de classe est codé en tant que `xsd:normalizedString`.

Exemple	
INTERLIS	<pre>DOMAIN InterlisClassRef = CLASS;</pre>
GML	<pre><xsd:simpleType name="InterlisClassRef"> <xsd:restriction base="xsd:normalizedString"> </xsd:restriction> </xsd:simpleType></pre>

La valeur contient le nom entièrement qualifié de classe, de structure ou d'association (par ex. «MD01MOCH24F.Points_fixesCategorie1.PFP1»).

6.15.8 Codification de types de chemin d'attributs

Le type de chemin d'attribut est codé en tant que `xsd:normalizedString`.

Exemple	
INTERLIS	<pre>DOMAIN InterlisAttributeRef = ATTRIBUTE;</pre>
GML	<pre><xsd:simpleType name="InterlisAttributeRef"> <xsd:restriction base="xsd:normalizedString"> </xsd:restriction> </xsd:simpleType></pre>

La valeur contient le nom de classe entièrement qualifié, suivi du nom d'attribut qui est séparé par un point (par ex. «CatalogueDonneesBase.Points_fixes.PFP.Numero»).

6.15.9 Codification des coordonnées

Les coordonnées sont codées en tant que `gml:PointPropertyType`.

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

Les indications sur le domaine de valeurs d'axe, l'unité, le système de référence et le sens d'orientation ne peuvent pas être consignées dans le schéma d'application GML.

Dans le cas où aucun système de référence n'a été défini dans le modèle INTERLIS, il faut utiliser une valeur par défaut spécifique à l'outil pour l'attribut XML `srsName` dans l'instance GML. Si un système de référence a été défini dans le modèle INTERLIS, il faut utiliser dans l'instance GML, une valeur qui correspond à ce système de référence. La valeur peut être définie à l'aide du méta-attribut «`ili2.iligml10.srsName`», lors de la déclaration de l'objet du système de référence. Si aucun méta-attribut n'est défini, la représentation est spécifique à l'outil.

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

6.15.10 Codification des polygones

Les polygones sont codées en tant que `gml:CurvePropertyType`.

Exemple	
INTERLIS	<pre>DOMAIN Point2D = COORD</pre>

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

Les indications sur le domaine de valeurs de point d'appui et les formes de portions de courbes autorisées ne peuvent pas être consignées dans le schéma d'application GML.

6.15.11 Codification des surfaces indépendantes

Les surfaces indépendantes sont codées en tant que `gml:SurfacePropertyType`. Mais dans l'instance seul `gml:Polygon` est autorisé.

Exemple	
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 -> 1; CLASS LandCover = Geometry: MANDATORY SURFACE WITH (STRAIGHTS) VERTEX Point2D WITHOUT OVERLAPS > 0.100; END LandCover; </pre>
GML	<pre> <xsd:element name="LandCover" type="LandCoverType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="LandCoverType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> <xsd:element name="Geometry" minOccurs="0" type="gml:SurfacePropertyType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </pre>
Instance GML	<pre> <LandCover gml:id="o0"> <Geometry> <gml:Polygon gml:id="g0" srsDimension="2" srs-Name="http://www.opengis.net/gml/srs/epsg.xml#21781"> <gml:exterior> <gml:LinearRing> </pre>

	<pre> <gml:posList>...</gmlposList> </gml:LinearRing> </gml:exterior> <gml:interior> <gml:LinearRing> <gml:posList>...</gmlposList> </gml:LinearRing> </gml:interior> </gml:Polygon> </Geometry> </LandCover> </pre>
--	--

Les indications sur le domaine de valeurs du point d'appui et les formes de portions de courbes autorisées ne peuvent pas être consignées dans le schéma d'application GML.

6.15.12 Codification d'une partition de territoire

La partition de territoire est codée en tant que `gml:SurfacePropertyType`. Mais dans l'instance seul `gml:Polygon` est autorisé.

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

Les indications sur le domaine de valeurs du point d'appui et les formes de portions de courbes autorisées ne peuvent pas être consignées dans le schéma d'application GML, d'autant plus qu'il s'agit d'une partition de territoire.

6.15.13 Codification de LINE ATTRIBUTES

Si une surface indépendante ou une partition de territoire comporte des attributs linéaires (INTERLIS 2.3), il est nécessaire de générer deux classes correspondantes :

- Une classe principale qui comporte les propriétés de la classe INTERLIS (sauf LINE ATTRIBUTES). La géométrie de cette classe est codée en tant que gml:SurfacePropertyType.
- Une classe supplémentaire qui comporte des lignes de frontières avec LINE ATTRIBUTES. Un objet de cette classe comporte une association à (aux) l'objet(s) correspondant(s) (min. 1, max. 2) de la classe principale.

Des attributs linéaires n'existent plus sous INTERLIS 2.4.

Exemple	
INTERLIS	<pre> DOMAIN LKoord = COORD 480000.00 .. 850000.00 [m] {CHLV03[1]}, 60000.00 .. 320000.00 [m] {CHLV03[2]}, ROTATION 2 -> 1; STRUCTURE GrenzlinieEigenschaften = streitig : BOOLEAN; END GrenzlinieEigenschaften; CLASS Grundstueck = Geometrie : MANDATORY AREA WITH (ARCS,STRAIGHTS) VERTEX LKoord WITHOUT OVERLAPS>0.10 LINE ATTRIBUTES GrenzlinieEigenschaften; END Grundstueck; </pre>
GML	<pre> <xsd:element name="GrenzlinieEigenschaften" ty- pe="GrenzlinieEigenschaftenType" substitution- Group="gml:AbstractObject"/> <xsd:complexType name="GrenzlinieEigenschaftenType"> <xsd:sequence> <xsd:element name="streitig" type="xsd:boolean" minOccurs="0"/> </xsd:sequence> </xsd:complexType> <xsd:element name="Grundstueck.Grenze" type="Grundstueck.GrenzeType" substitu- tionGroup="gml:AbstractFeature"/> <xsd:complexType name="Grundstueck.GrenzeType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> <xsd:element name="Geometry" minOccurs="0" ty- pe="gml:CurvePropertyType"/> <xsd:element name="lineattr" minOccurs="0" ty- pe="gml:GrenzlinieEigenschaften"/> <xsd:element name="Grundstueck" minOccurs="1" max- Occurs="2" type="gml:ReferenceType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> <xsd:element name="Grundstueck" type="GrundstueckType" sub- stitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="GrundstueckType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> <xsd:element name="Geometrie" minOccurs="0" </pre>

	<pre> type="gml:SurfacePropertyType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>
--	--

6.15.14 Codification de géométries multiples

Des géométries multiples sont codées en tant que géométries multiples GML : MULTICOORD en tant que gml:MultiPointPropertyType, MULTIPOLYLINE en tant que gml:MultiCurvePropertyType et MULTISURFACE en tant que gml:MultiSurfacePropertyType.

La codification des géométries multiples INTERLIS 2.3 s'effectue par des structures.

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

6.15.15 Codification de différents attributs de géométrie au sein de la même classe

Si une classe comporte plusieurs attributs de géométrie et si seulement un des attributs peut avoir une valeur à cause d'une condition de cohérence, ces attributs sont à coder en tant que xsd:choice. La condition de cohérence doit avoir la structure suivante:

```

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

```

Exemmmple INTERLIS 2.3	
INTERLIS	<pre> DOMAIN ... CLASS Landcover = Punkt:LKoord; Linie:POLYLINE WITH (ARCS,STRAIGHTS) VERTEX LKoord; </pre>

	<pre> Flaeche: SURFACE WITH (ARCS, STRAIGHTS) VERTEX LKoord ... MANDATORY CONSTRAINT (DEFINED(Punkt) AND NOT(DEFINED(Linie)) AND NOT(DEFINED(Flaeche))) OR (NOT (DEFINED(Punkt)) AND DEFINED(Linie) AND NOT(DEFINED(Flaeche))) OR (NOT(DEFINED(Punkt)) AND NOT(DEFINED(Linie)) AND DEFINED(Flaeche)); END Landcover; </pre>
GML	<pre> <xsd:element name="Landcover" type="LandcoverType" substitu- tionGroup="gml:AbstractFeature"/> <xsd:complexType name="LandcoverType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> <xsd:choice> <xsd:element name="Punkt" type="gml:PointPropertyType"/> <xsd:element name="Linie" type="gml:CurvePropertyType"/> <xsd:element name="Flaeche" type="gml:SurfacePropertyType"/> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> ... </pre>

6.15.16 Codification de types pour l'identification d'objet

Les types pour les identifications d'objet sont représentés en tant que `xsd:int` («OID NumericType») ou `xsd:token` («OID» en tant que «ANY» ou «TextType»).

Le méta-attribut «`ili2.iligml20.identifieurCodeSpace`» permet de définir le formatage de la valeur de l'identificateur (la valeur pour l'attribut XML `gml:codeSpace` dans l'instance GML).

Le méta-attribut «`ili2.iligml20.identifieurPattern`» permet de définir le formatage de la valeur de l'identificateur (des valeurs pour l'élément XML `gml:identifieur` dans l'instance GML).

Le méta-attribut «`ili2.iligml20.hrefPattern`» permet de définir le formatage de la valeur de référence (des valeurs pour l'attribut XML `xlink:href` dans l'instance GML).

Il est possible d'utiliser «`#{value}`» en tant qu'espace réservé pour la valeur OID dans les définitions de modèle.

Exemple	
INTERLIS	<pre> DOMAIN BFS_EGID = OID 1 .. 9999999999; UUID = OID TEXT*36; !!@ ili2.iligml20.identifieurCodeSpace= "http://inspire.jrc.ec.europa.eu/" !!@ ili2.iligml20.identifieurPattern= "urn:inspire:object:id:#{value}" !!@ ili2.iligml20.hrefPattern= "urn:inspire:object:id:#{value}" </pre>

	InspiredId = OID TEXT; !! z.B. "DENW1000:A00012ab"
GML	<pre> <xsd:simpleType name="BFS_EGID"> <xsd:restriction base="xsd:int"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="UUID"> <xsd:restriction base="xsd:token"> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="InspiredId"> <xsd:restriction base="xsd:token"> </xsd:restriction> </xsd:simpleType> </pre>

6.15.17 Codification d'autres formes de portions de courbes

Cette spécification n'est pas compatible avec cette version.

Remarque: Il faut d'abord créer un modèle de base INTERLIS, de sorte à permettre une représentation sur des types de courbes prédéfinis dans GML.

7 Règles de codification d'instance

En plus des règles de codification de schéma, les règles d'instance définies dans cette section s'appliquent également.

7.1 Codification de plusieurs contenants (TOPIC) dans un fichier de transfert

Dans le cas où plusieurs contenants sont transférés dans un fichier de transfert (cas typique), on utilise «TRANSFER» en tant qu'élément racine XML. Cet élément et les types correspondants sont définis dans l'annexe A.

7.2 Codification des objets et éléments de structure

Pour un contenant, les objets sont toujours codés en ligne. Par contre, pour un autre objet, les objets ne sont jamais codés en ligne.

Les éléments de structure sont toujours codés en ligne pour un objet ou un autre élément de structure. Les éléments de structure ne sont jamais référencés, bien qu'ils contiennent un `gml:id`.

7.3 Codification d'identifications de contenant et d'objet

L'identification de transfert d'un objet ou d'un contenant est codée en tant que `gml:id`.

L'identification stable de l'objet ou du contenant est codée en tant que `gml:identifier`.

Si, lors de la définition du domaine de valeurs de l'OID, aucune valeur n'a été définie pour le méta-attribut «`ili2.iligml20.identifierCodeSpace`», alors on utilise l'identificateur de l'espace nominal du modèle (`%StdModelNamespaceIdentifier%`; voir chapitre 6.6), en tant que «codeSpace», complété par le nom de la définition du domaine de valeurs, pour les OID (exemple pour le domaine de valeurs «UUIDOID» à partir du modèle INTERLIS (annexe A dans le manuel de référence [1]) : «`http://www.interlis.ch/ILIGML-2.0/INTERLIS/UUIDOID`»).

Si, lors de la définition du domaine de valeurs de l'OID, une valeur a été définie pour le méta-attribut «`ili2.iligml20.identifierCodeSpace`», alors celle-ci est utilisée.

Si, lors de la définition du domaine de valeurs de l'OID, aucune valeur n'a été définie pour le méta-attribut «`ili2.iligml20.identifierPattern`», alors la valeur est utilisée sans être modifiée (ce qui correspond au modèle «`{value}`»).

Si, lors de la définition du domaine de valeurs de l'OID, une valeur a été définie pour le méta-attribut «`ili2.iligml20.identifierPattern`», alors celle-ci est utilisée.

7.4 Codification d'associations

`gml:id` est toujours utilisé comme valeur de référence (par ex, `xlink:href="#o1`" en tant que renvoi à un objet avec `gml:id="o1"`), sauf si l'objet référencé possède un OID stable.

Si l'objet référencé possède un OID stable, alors celui-ci est utilisé pour la référence.

Si lors de la définition du domaine de valeurs de l'OID, aucun modèle n'a été défini pour la formation de la valeur `xlink:href`, alors c'est «`urn:x-ili:${value}`» qui sera utilisé en tant que modèle (par ex. `xlink:href="urn:x-ili:o1"` en tant que renvoi à un objet avec `gml:identifiant="o1"`).

Si lors de la définition du domaine de valeurs de l'OID, un modèle a été défini pour la formation de la valeur `xlink:href`, alors c'est ce modèle qui sera utilisé.

7.5 Codification de géométries

Toutes les géométries doivent être codées en ligne, et ce, même si le schéma GML de base autorisait les références.

Pour «SURFACE» et «AREA», seul `gml:Polygon` est autorisé (par conséquent, `gml:Surface` n'est pas autorisé).

Pour `gml:LineStringSegment`, seul `gml:posList` est autorisé dans GML-SF.

Pour `gml:exterior/gml:interior`, seul `gml:LinearRing` (avec `gml:posList`) est autorisé dans GML-SF.

Pour `gml:Arc`, seul `gml:posList` est autorisé dans GML-SF.

7.6 Codification de listes de code

La codification d'instance des énumérations est définie dans le chapitre 6.15.3.

8 Exclusion de responsabilité – Droits de tiers

Les normes élaborées par l'Association **eCH** et mises gratuitement à la disposition des utilisateurs, ainsi que les normes de tiers adoptées, ont seulement valeur de recommandations. L'Association **eCH** ne peut en aucun cas être tenue pour responsable des décisions ou mesures prises par un utilisateur sur la base des documents qu'elle met à disposition. L'utilisateur est tenu d'étudier attentivement les documents avant de les mettre en application et au besoin de procéder aux consultations appropriées. Les normes **eCH** ne remplacent en aucun cas les consultations techniques, organisationnelles ou juridiques appropriées dans un cas concret.

Les documents, méthodes, normes, procédés ou produits référencés dans les normes **eCH** peuvent le cas échéant être protégés par des dispositions légales sur les marques, les droits d'auteur ou les brevets. L'obtention des autorisations nécessaires auprès des personnes ou organisations détentrices des droits relève de la seule responsabilité de l'utilisateur.

Bien que l'Association **eCH** mette tout en œuvre pour assurer la qualité des normes qu'elle publie, elle ne peut fournir aucune assurance ou garantie quant à l'absence d'erreur, l'actualité, l'exhaustivité et l'exactitude des documents et informations mis à disposition. La teneur des normes **eCH** peut être modifiée à tout moment sans préavis.

Toute responsabilité relative à des dommages que l'utilisateur pourrait subir par suite de l'utilisation des normes **eCH** est exclue dans les limites des réglementations applicables.

9 Droits d'auteur

Tout auteur de normes **eCH** en conserve la propriété intellectuelle. Il s'engage toutefois à mettre gratuitement, et pour autant que ce soit possible, la propriété intellectuelle en question ou ses droits à une propriété intellectuelle de tiers à la disposition des groupes de spécialistes respectifs ainsi qu'à l'association **eCH**, pour une utilisation et un développement sans restriction dans le cadre des buts de l'association.

Les normes élaborées par les groupes de spécialistes peuvent, moyennant mention des auteurs **eCH** respectifs, être utilisées, développées et déployées gratuitement et sans restriction.

Les normes **eCH** sont complètement documentées et libres de toute restriction relevant du droit des brevets ou de droits de licence. La documentation correspondante peut être obtenue gratuitement.

Les présentes dispositions s'appliquent exclusivement aux normes élaborées par **eCH**, non aux normes ou produits de tiers auxquels il est fait référence dans les normes **eCH**. Les normes incluront les références appropriées aux droits de tiers.

Annexe A – Schéma de base prédéfini

```

<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>

```

Annexe B - Schéma de base ili2c

```
<?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>
```

Annexe C - Exemple

Modèle INTERLIS 2

```
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.
```

Modèle INTERLIS 1

```

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.

```

Schéma d'application GML

```

<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" schemaLoca-
    tion="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: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-Données de transfert

```

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-Données de transfert («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>
```

Données GML

```

<?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:Curve gml:id="g41.1">

```

```

        <gml:segments>
          <gml:LineStringSegment interpolation="linear">
            <gml:posList>186.38 206.82 148.3 228.97 113 219.97
118.19 179.82 140.69 156.63</gml:posList>
          </gml:LineStringSegment>
          <gml:LineStringSegment interpolation="linear">
            <gml:posList>140.69 156.63 170.18 176</gml:posList>
          </gml:LineStringSegment>
          <gml:Arc numArc="1" interpolation="circularArc3Points">
            <gml:posList>170.18 176.0 183.26 188.19 186.38
206.82</gml:posList>
          </gml:Arc>
        </gml:segments>
      </gml:Curve>
    </gml:curveMember>
  </gml:Ring>
</gml:exterior>
<gml:interior>
  <gml:Ring>
    <gml:curveMember>
      <gml:Curve gml:id="g41.2">
        <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:interior>
</gml:Polygon>
</ili2gml:Form>
</ili2gml:BoFlaechen>
</ili2gml:member>
<ili2gml:member>
  <ili2gml:BoFlaechen gml:id="x20">
    <ili2gml:Art>befestigt</ili2gml:Art>
    <ili2gml:Form>
      <gml:Polygon gml:id="g31" srsName="urn:ogc:def:crs:EPSG::21781" srsDimension="2">
        <gml:exterior>
          <gml:Ring>
            <gml:curveMember>
              <gml:Curve gml:id="g31.1">
                <gml:segments>
                  <gml:LineStringSegment interpolation="linear">
                    <gml:posList>140.69 156.63 145.08 149.94 174.1
169.0</gml:posList>
                  </gml:LineStringSegment>
                  <gml:Arc numArc="1" interpolation="circularArc3Points">
                    <gml:posList>174.1 169.0 190.75 185.21 194.26
208.19</gml:posList>
                  </gml:Arc>
                  <gml:LineStringSegment interpolation="linear">
                    <gml:posList>194.26 208.19 186.38 206.82</gml:posList>
                  </gml:LineStringSegment>
                  <gml:Arc numArc="1" interpolation="circularArc3Points">
                    <gml:posList>186.38 206.82 183.26 188.19 170.18
176.0</gml:posList>
                  </gml:Arc>
                </gml:segments>
              </gml:Curve>
            </gml:curveMember>
          </gml:Ring>
        </gml:exterior>
      </gml:Polygon>
    </ili2gml:Form>
  </ili2gml:BoFlaechen>
</ili2gml:member>
</ili2gml:member>

```

```

        </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>

```

Annexe D – Collaboration & vérification

	Organe de coordination de la géoinformation au niveau fédéral GCS
Alain Buogo	Office fédéral de topographie swisstopo
Pasquale Di Donato	Office fédéral de topographie 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	Conférence des services cantonaux de géoinformation CCGEO
Thorsten Reitz	wetransform GmbH
Rolf Zürcher	Office fédéral de topographie swisstopo

Annexe E – Abréviations et glossaire

EBNF	Notation étendue Backus-Naur
FMT	Format de transfert INTERLIS 1
GML	Geographic Markup Language
GML-SF	Geographic Markup Language Simple Feature
INTERLIS	Langage de modélisation de géodonnées et format de transfert
ITF	Fichier de transfert INTERLIS 1
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	Format de Transfert INTERLIS 2

Annexe F – Modifications par rapport à la version précédente

Les modifications suivantes ont été élaborées dans cette version par rapport à la version précédente :

- Tout le document a été adapté à la nouvelle version d'INTERLIS (2.4)
- Le résumé a été adapté selon le contenu modifié correspondant
- Les tables de matières ont été mises à jour
- Les références normatives ont été actualisées
- Le chapitre 4 a été supprimé. Le cas d'application principal est décrit dans le chapitre 2.3
- Le chapitre 5 a été supprimé. Si besoin, le texte a été repris dans d'autres chapitres.
- L'énumération des chapitres a été mise à jour
- Le modèle ili2c est défini en annexe B
- La représentation de DomainDefComplex (chapitre 6.7) a été modifiée légèrement
- Le chapitre 7.9 a été divisé en trois nouveaux chapitres: ch. 6.9 codification de classes, ch. 6.10 codification de structures, ch. 6.11 codification d'associations
- La codification de structures a été adaptée
- La codification d'associations a été adaptée
- La représentation de l'illustration des types INTERLIS sur des types de schéma XML a été mise à jour
- La codification d'énumérations a été adaptée
- La codification de coordonnées, des polygones, des surfaces indépendantes, des partitions de territoire, des géométries multiples a été adaptée afin d'utiliser les types de base GML
- Le chapitre codification de LINE ATTRIBUTES (6.15.13) est ajouté
- Le chapitre codification de géométries multiples (6.15.14) est ajouté
- Le chapitre Codification de différents attributs de géométrie au sein de la même classe (6.15.15) est ajouté
- Le chapitre 7 (avant chapitre 8) a été actualisé
- Les chapitres 8 et 9 (avant chapitres 9 et 10) restent inchangés
- Le schéma ili2c (annexe B) a été actualisé
- Les annexes A et C (avant B) ont été actualisés
- Les annexes D, E et F ont été rajoutés