

eCH-0118 Règles de codification GML pour INTERLIS

Titre	Règles de codification GML pour INTERLIS
Code	eCH-0118
Type	norme de procédure
Stade	expérimentale
Version	1.00
Statut	Remplacé
Validation	2011-06-08
Date de publication	2016-09-09
Remplace	
Langues	Français, Allemand
Auteur(s)	Groupe spécialisé eCH INTERLIS <i>Claude Eisenhut, Eisenhut Informatik AG</i> <i>Michael Germann, infoGrips GmbH</i> <i>Dr. Peter Staub, swisstopo</i>
Éditeur / Distributeur	Association eCH, Mainaustrasse 30, Case postale, 8034 Zürich T 044 388 74 64, F 044 388 71 80 www.ech.ch / info@ech.ch

Condensé

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 de codification pour les données géographiques qui se base sur du XML.

En premier lieu, seront définis les concepts fondamentaux des langues INTERLIS et GML utilisées, ainsi que les règles de codification et les cas d'application XML (chapitre 4). Après les aspects fondamentaux sur la codification d'objets –XML, s'ensuit une présentation des différences entre la codification GML et la codification du XML spécifique à INTERLIS (Chapitre 5 et 6).

La partie principale de cette norme se compose de la définition détaillée et complète des règles de codification de schéma (chapitre 7) ainsi que des règles de codification d'instance (chapitre 8).

L'annexe A comprend le schéma XML de base prédéfini pour la codification GML d'INTERLIS. En plus de la présente norme, ce document est disponible à part, sous forme de fichier – schéma XML. La présente spécification se termine par un exemple simple d'application (Annexe B).

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	8
4	Notions, concepts	9
4.1	INTERLIS	9
4.2	Geography Markup Language (GML)	9
4.3	Règles de codification.....	10
4.4	Cas d'application du XML	10
5	Principe de la codification d'objets dans XML	12
5.1	Caractéristiques d'objet: attributs dans modèle INTERLIS	12
5.2	Associations	12
5.3	Discriminateur de type	12
5.4	Héritage.....	12
5.5	Transformation sémantique	12
6	Différences sur les règles de codification XML INTERLIS	13
6.1	Règles de codification.....	13
6.2	Présentation d'un transfert.....	13
6.3	Lecture de modèles élargis / traduits	13
6.4	Jeu de caractères	13
6.5	Types de transferts.....	14
7	Règles de codification – schéma	15
7.1	Remarque préliminaire	15
7.2	Codification des caractères	15
7.3	Règles générales.....	15
7.4	Représentation des éléments de modèle INTERLIS sur des noms de schémas XML	15

7.5	Représentation de types de base	16
7.6	Présentation générale du schéma XML	16
7.7	Codification de domaines de valeurs	17
7.8	Codification de thèmes	18
7.9	Codification de catégories, de structures et d'associations	20
7.10	Codification des vues.....	21
7.11	Codification des rapports	21
7.11.1	Rapports intégrés	22
7.11.2	Rapports non intégrés	24
7.12	Codification des définitions de graphique	26
7.13	Codification des attributs	26
7.14	Représentation des types INTERLIS sur des types de schéma XML.....	29
7.14.1	Aperçu.....	29
7.14.2	Codification des chaînes de caractères	29
7.14.3	Codification d'énumérations	30
7.14.4	Codification de types de données numériques	33
7.14.5	Codification de domaines de valeurs formatés	34
7.14.6	Codification des contenants	35
7.14.7	Codification de types de catégories.....	35
7.14.8	Codification de types de chemin d'attribut	36
7.14.9	Codification des coordonnées	36
7.14.10	Codification des polylignes.....	37
7.14.11	Codification des surfaces simples.....	37
7.14.12	Codification des zonages	38
7.14.13	Codification de types pour l'identification d'objet	40
7.14.14	Codification d'autres formes de pièces courbes.....	41
8	Règles de codification d'instance	42
8.1	Codification de plusieurs contenants (TOPIC) dans un fichier de transfert	42
8.2	Codification des objets et éléments de structure.....	42
8.3	Codification d'identifications de contenant et d'objet.....	42
8.4	Codification de rapports.....	43
8.5	Codification de géométries	43

9	Exclusion de responsabilité – Droits de tiers	43
10	Droits d’auteur	44
	Annexe A – Schéma de base pré-défini	45
	Annexe B - Exemple	47
	INTERLIS 2-Modèle	47
	INTERLIS 1-Modèle	48
	Schéma d’application GML.....	49
	INTERLIS 1-Données de transfert.....	51
	INTERLIS 2-Données de transfert («INTERLIS-XML»).....	52
	Données GML	53

1 Statut du document

Remplacé: Le document a été remplacé par une nouvelle version plus récente. Son utilisation reste encore possible, mais il est recommandé d'appliquer la dernière version.

2 Introduction

2.1 Aperçu

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

2.2 Conditions préalables

Ce document suppose la possession de connaissances sur INTERLIS et le GML. En vue d'un exposé approfondi, se reporter aux sources [1], [2]. 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 de modèle la plus précise possible.
- Il ne doit y avoir aucune perte de donnée face à un transfert de données 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 en aucun cas remplacer le modèle INTERLIS.
- Le schéma d'application GML obtenu se destine aussi bien au transfert de données de masse qu'aux géoservices Web (notamment WFS).
- Ce document utilise INTERLIS 2.3 et GML 3.2.x. Étant donné qu'INTERLIS 1 peut plus ou moins être traduit à 1:1 d'après INTERLIS 2.3, cela est également valable pour INTERLIS 1.

Remarque: Il n'y a aucune perte de donnée, 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 en résultant ne représente donc 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 en résultant est conforme au GML Simple Feature (GML-SF), 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 selon GML-SF (par ex. arc de cercle), alors le schéma d'application GML obtenu n'est pas conforme à GML-SF.

3 Références normatives

- [1] Groupe spécialisé eCH Geoinformation (2006): *eCH-0031: Geoinformation: INTERLIS 2 – Manuel de référence*, Norme eCH.
Site <http://www.ech.ch> → Normes → eCH-0031 (2010-09-06)
- [2] 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 (2010-09-06)
- [3] World Wide Web Consortium W3C (2004): *XML Schema 1.1*. Norme.
Site <http://www.w3.org/XML/Schema> (2010-09-06)
- [4] ISO/IEC 14977 (1996): *Information technology – Syntactic metalanguage – Extended BNF*. Norme internationale.
- [5] ISO/IEC 10646:2003 (2003): *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*. Norme internationale.
- [6] Groupe spécialisé eCH INTERLIS (2010): *eCH-0117: Méta-attributs pour modèles INTERLIS*. Norme eCH [en cours d'élaboration].

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].

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

- Un langage de description des données et
- Des règles de codification XML.

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 B 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 [2] est un ensemble de schémas XML de base [3] 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 de 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 catégories et quels éléments sont les attributs de ces catégories. Cette description est moins précise que celle effectuée au moyen d'INTERLIS, mais ceci 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 B présente un exemple de schéma d'application avec un fichier de transfert conforme à ce schéma.

Remarque: Les schémas GML 3.1 ne valident pas tout! Il ne s'agit pas seulement d'un problème provenant de GML; se reporter également à <http://schemas.opengis.net/gml/3.1.1/readme.txt>

Dans la spécification GML 3.2 [2], on trouve ce qui suit 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, il est possible d'obtenir le format de transfert (transfert basé sur un modèle) avec l'application de certaines règles. Ce faisant, les règles sont donc formulées de sorte à pouvoir être appliquées sur n'importe quel modèle de données.

4.4 Cas d'application du XML

En principe, les cas d'application qui influencent considérablement le design d'une structure XML sont les suivants:

- *Textes/documents* (par ex. XHTML, DocBook). Dans ce cas, c'est la lisibilité pour les personnes et la structuration/l'ordre correct des éléments XML qui sont au premier plan. Les types de données présentent moins d'importance.
- *Annonces* (par ex. mutation de bien foncier ou dans le contexte AJAX et des services Web). Dans ce cas, c'est la facilité du traitement qui se trouve au premier plan ; souvent, on accepte volontairement des redondances des données.
- *Transfert de données de masse* (par ex. Interface de mensuration officielle). Dans ce cas, c'est l'efficacité du traitement et la taille des fichiers XML qui figurent en premier plan.
- *Enormes quantités de données* (par ex. images, séries de mesures). Dans ce cas, la taille des fichiers XML est tellement dominante que le marquage des différentes informations est en partie ou intégralement supprimé.
- *«Données» éditées par des personnes* (par ex. documents de schémas XML, scripts «ant build»). Dans ce cas, c'est la lisibilité pour les personnes qui est au premier plan. Contrairement au cas « Texte/ documents », les types de données sont également importants ici.

Autres aspects:

- *Utilisation interne aux applications.* La structure XML peut être orientée précisément en fonction des besoins de cette application.
- *Différents rapports d'utilisation,* par ex. différentes restrictions « sévères », car au début, toutes les informations ne sont pas encore présentes dans une chaîne de processus de traitement.
- *Utilisation inter-applications.* On ne peut pas orienter la structure XML selon les besoins d'une application; cependant, les différentes applications possèdent habituellement une représentation pourtant analogue à la réalité.
- *Utilisation inter-domaines.* Les différentes applications présentent typiquement une représentation différente de la réalité. Exemple: dans l'administration fiscale, les personnes se

trouvent au premier plan, les terrains et bâtiments sont également présents. Dans le registre foncier, les droits réels sont au premier plan et les terrains et les personnes sont également présents. Dans la mensuration officielle, les géométries des terrains et les bâtiments figurent au premier plan et les personnes ne sont pas présentes.

- Schéma existant ou style de design de schéma existant.
- Méta données, notamment schémas, en tant que partie des données
- Structure générique au lieu de structure spécifique, par ex:

```
<object class="Person">  
  <attr name="Name" value="Muster"/>  
  <attr name="Vorname" value="Peter"/>  
</object>
```
- *Extensibilité*: dans les structures existantes, peut-on facilement intégrer ou renvoyer à d'autres caractéristiques (par ex. pour la gestion des versions de documents)?
- *Tolérance de modification*: Peut-on facilement modifier les structures de données *sans avoir* à re-développer entièrement le programme?

5 Principe de la codification d'objets dans XML

Quelques décisions de projets fondamentales doivent être prises, la plupart d'entre elles sont toutefois déjà concernées par la spécification GML.

5.1 Caractéristiques d'objet: attributs dans modèle INTERLIS

Les caractéristiques d'objet peuvent être codées en tant qu'attributs XML ou en tant que sous-éléments XML. Via GML, la variante des sous-éléments XML est demandée.

5.2 Associations

Les associations peuvent être codées en tant qu'objets autonomes («tableau intermédiaire»), en tant que référence unilatérale, référence réciproque ou en tant qu'intégration d'objet.

Aucune variante particulière n'est demandée par GML.

5.3 Discriminateur de type

Le type d'un objet peut être codé en tant qu'attribut XML (en règle générale `xsi:type`) ou bien sous forme d'élément XML.

Via GML, la variante d'élément XML est demandée. GML définit cela en tant que «Object-Property-Object Pattern».

5.4 Héritage

La hiérarchie d'héritage peut être exprimée dans le schéma XML ou être représentée sous forme de structures de contenu identique selon différents procédés (les structures n'entraînent plus l'expression de l'héritage, comme dans le cas du Mapping objet-relationnel (l'O/R-Mapping¹)).

5.5 Transformation sémantique

En principe, on peut également effectuer des transformations sémantiques, tant que celles-ci ne modifient pas le contenu, mais qu'elles se contentent seulement d'en modifier la structure. Par exemple, on pourrait réunir deux catégories reliées à travers une association 1:1 en un seul élément XML.

¹ *dt.* Représentation d'objet relationnel; procédé pour la représentation d'objets dans des bases de données relationnelles.

6 Différences sur les règles de codification XML INTERLIS

6.1 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.

6.2 Présentation d'un transfert

Le manuel de référence INTERLIS divise un transfert en générique et en domaine de données. Ce document ne traite pas de 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).

Variante: transférer les indications du générique en tant que métadonnées GML.

6.3 Lecture de modèles élargis / traduits

Le manuel de référence INTERLIS 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 les modèles de données INTERLIS traduits, aucun schéma d'application GML ne sera créé, c.à.d. que le format de transfert ne se trouve pas modifié lors d'une traduction.

Remarque: La manipulation d'une extension de modèle inconnue avec GML (sans logiciel spécial) est-elle en fait possible?

Inconnu dans le sens de «schéma non disponible»: non.

Inconnu dans le sens de «schéma disponible, mais pas implémenté»: oui.

6.4 Jeu de caractères

Le manuel de référence INTERLIS définit un jeu de caractères restreint (annexe B 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.

6.5 Types de transferts

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

Le présent document décrit comment, dans le cadre de prestations Web, les différents objets sont transférés. Le transfert classique basé sur un fichier est défini en option (correspond dans INTERLIS à: «FULL» et «COMPLETE»).

Remarque: le transfert incrémentiel ne peut pas être réalisé avec des logiciels GML standard car le GML ne définit aucune sémantique de ce type. Les indications de consistance peuvent être codées en tant que métadonnées GML.

+

7 Règles de codification – schéma

7.1 Remarque préliminaire

Pour la formalisation des règles de dérivation, on utilise la notation EBNF dans le chapitre 2.1 du manuel de référence INTERLIS [4].

7.2 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 [5].

7.3 Règles générales

Le schéma XML créé [3] peut contenir autant de commentaires (<!-- ... -->) ou d'éléments supplémentaires `xsd:annotation` que l'on souhaite.

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

Chaque modèle de données INTERLIS contient un propre espace de nom 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 2.3; MODEL ModelA (de) AT "mailto:ceis@localhost" VERSION "2010-02-31" = 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>

7.5 Représentation de types de base

Pour les types de base indépendant du modèle et provenant d'INTERLIS (par ex. «HALIGNMENT», «SurfacePropertyType»), un schéma de base (Annexe A) est défini.

Variante: dans chaque schéma créé, il faut toujours re-définir (c.à.d. moins de fichiers – schéma).

7.6 Présentation 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-1.0/' %ModelName%.

```

L'identificateur de l'espace de nom XML (règle «Identificateur d'espace de nom») résulte du nom de modèle ou peut être défini au moyen du méta-attribut «ili2.iligml10.namespaceName» [6].

Exemple	
INTERLIS	<pre> INTERLIS 2.3; MODEL ModelA (de) AT "mailto:ceis@localhost" VERSION "2010-02-31" = IMPORTS Units; END ModelA. </pre>
GML	<pre> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.interlis.ch/ILIGML-1.0/ModelA" targetNamespace= "http://www.interlis.ch/ILIGML-1.0/ModelA" elementFormDefault="qualified" attributeFormDefault="unqualified" xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:ns1= "http://www.interlis.ch/ILIGML-1.0/Units"> <xsd:annotation> <xsd:appinfo source="http://www.interlis.ch/ili2c"> <ili2:model>ModelA</ili2:model> <ili2:modelVersion>2010-0-31</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-1.0/Units"/> </xsd:schema> </pre>

7.7 Codification de domaines de valeurs

Les définitions des domaines de valeurs sont représentées comme suit:

```

DomainDef = DomainDefComplex | DomainDefSimple.

DomainDefComplex = '<xsd:complexType name="%DomainName%">
  <xsd:complexContent>
    <xsd:restriction base="%BaseType%">
      <!-- any applicable xsd restrictions -->
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>'.

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

```

Selon la règle de représentation du type de données INTERLIS correspondant (voir chapitre 7.14), c'est la règle «DomainDefComplex» ou la règle «DomainDefSimple» qui sera appliquée.

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>
Instance GML	<pre> <Bodenbedeckung gml:id="b1"> <member> <BoFlaechen gml:id="o1"> ... </BoFlaechen> </member> <member> <BoFlaechen gml:id="o2"> ... </BoFlaechen> </member> </Bodenbedeckung> </pre>

Variante: Représenter chaque thème en tant que schéma XML.

7.9 Codification de catégories, de structures et d'associations

Les catégories et structures 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>'.

```

Dans le schéma XML, on ne peut ainsi plus reconnaître s'il s'agit d'une catégorie «CLASS» ou d'une structure «STRUCTURE».

Les associations (voir également chapitre 7.11) sont représentées comme suit:

```

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, des rôles intégrés au sein de la catégorie / association, s'applique ce qui suit: on procède d'abord au codage de tous les rôles, puis à celui de tous les attributs, et enfin à celui de tous les rôles intégrés. Les attributs et les rôles sont codés conformément à leur ordre de définition dans le fichier de modèle. Les rôles intégrés sont triés par ordre alphabétique croissant. Il n'y a pas de création d'un attribut spécialisé ou d'un rôle spécialisé (avec le mot clé «EXTENDED»).

Si la catégorie ne représente aucune extension d'une autre catégorie, alors on utilise `gml:AbstractFeatureType` en tant que type de base ou `gml:AbstractFeature` en tant qu'élément de base. Si la catégorie correspond à une extension, on utilise le type de base ou l'élément de base créé correspondant. La structure d'héritage des catégories est ainsi reproduite dans le schéma XML.

Dans le cas d'associations intégrées, seuls les attributs sont codés en tant que partie du type pour l'association, c.à.d. aucune «RoleDef» (voir également chapitre **Fehler! Verweisquelle konnte nicht gefunden werden.**).

Les paramètres ne sont pas retransmis, à une exception près, comme cela est indiqué dans le chapitre **Fehler! Verweisquelle konnte nicht gefunden werden.** .

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>
Instance GML	<pre><Horizont gml:id="o1"> <Horizontbezeichnung>Ah</Horizontbezeichnung> </Horizont></pre>

Variante: «STRUCTURE» basée sur `gml:AbstractObject`

7.10 Codification des vues

Pour la codification des vues, se reporter au chapitre 3.2.4 dans le manuel de référence INTERLIS [1]. Seuls les attributs qui ont été indiqués dans la Vue, soit de façon explicite sous «ATTRIBUTE», soit de façon implicite avec «ALL OF», sont transmis en tant qu'attributs de l'objet de vue.

7.11 Codification des rapports

Les rapports sont codés selon deux types: intégré ou non intégré. Un rapport intégré est codé en tant que sous-élément d'une catégorie concernée par l'association. L'instance d'un rapport non intégré (lien) est codée comme une instance de catégorie.

Les rapports sont toujours intégrés, sauf:

- lorsqu'ils possèdent plus de deux rôles
- lorsque, pour les deux rôles (de base), la cardinalité est supérieure à 1
- lorsqu'un OID est demandé pour le rapport
- pour certains rapport interthématiques (cf. ci-dessous).

Dans le cas où l'un des deux rôles (de base) présente une cardinalité maximale supérieure à 1, ce rôle est intégré dans la catégorie-objectif. Si cette catégorie-objectif est définie dans un autre Topic que l'association (de base), alors l'intégration ne peut pas être effectuée.

Dans le cas où les deux rôles (de base) présentent une cardinalité maximale inférieure ou égale à 1, le deuxième rôle est intégré dans la catégorie-objectif. Si la catégorie-objectif est définie dans un autre Topic que l'association (de base) et que la catégorie-objectif du premier rôle est définie dans le même Topic que l'association (de base), alors le premier rôle est intégré dans la catégorie-objectif. Ceci signifie qu'il n'est pas possible d'effectuer d'intégration lorsque les catégories-objectif des deux rôles sont définies dans un autre Topic que l'association (de base).

Variante: Intégrer la référence de sorte que la cardinalité dans le schéma XML soit visible.

Variante: Dans le cas de composition, intégrer la catégorie-objectif.

Variante: Représenter les rôles de sorte à pouvoir effectuer la codification de la référence ou de l'objet intégré en tant qu'instance.

7.11.1 Rapports intégrés

Les rapports intégrés sont transmis en tant que référence dans la catégorie pour laquelle le rapport est intégré. Si le rapport est ordonné, un attribut XML supplémentaire («ORDER_POS»; défini dans l'annexe A) s'avère alors nécessaire.

```

EmbeddedRoleDef = ( EmbeddedUnorderedRoleDef
| EmbeddedOrderedRoleDef)
'<xsd:element name="%RoleName%.LINK_DATA"
  'minOccurs="0">'
  '<xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="%AssociationName%"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>'.

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

```

```

</xsd:element>'.
EmbeddedOrderedRoleDef =
'<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>'.

```

Pour %RoleName%, le nom du rôle renvoyant à l'objet opposé doit être indiqué (l'autre rôle n'est pas codé).

Si le rapport intégré possède des attributs, ceux-ci sont codés en tant qu'éléments supplémentaires %RoleName%.LINK_DATA. Dans le schéma, cet élément est toujours défini, sauf si le rapport est «FINAL» et ne possède aucun attribut. Si le rapport n'est pas «FINAL», les attributs ne peuvent qu'être éventuellement ajoutés dans une extension; l'élément pour l'intégration doit donc être défini sans attributs dans le cas de rapports sans « FINAL ».

Exemple	
INTERLIS	<pre> CLASS Liegenschaft = END Liegenschaft; CLASS Person = END Person; ASSOCIATION Eigentum = Grundsteuck -- {0..*} Liegenschaft; Eigentuemmer -- {1} Person; END Eigentum; </pre>
GML	<pre> <xsd:element name="Liegenschaft" type="LiegenschaftType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="LiegenschaftType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> <xsd:element name="Eigentuemmer" type="gml:ReferenceType"> <xsd:annotation> <xsd:appinfo> <gml:targetElement>Person</gml:targetElement> </xsd:appinfo> </xsd:annotation> </xsd:element> <xsd:element name="Eigentuemmer.LINK_DATA" minOccurs="0"> <xsd:complexType> <xsd:sequence> <xsd:element ref="Eigentum"/> </xsd:sequence> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>

	<pre> </xsd:sequence> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> <xsd:element name="Person" type="PersonType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="PersonType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> <xsd:element name="Eigentum" type="EigentumType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="EigentumType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>
Instance	<pre> <Person gml:id="o1"/> <Liegenschaft gml:id="o2"> <Eigentuemmer xlink:href="#o1"/> </Liegenschaft> </pre>
GML	

7.11.2 Rapports non intégrés

Les rapports non intégrés sont transmis comme des instances d'objet de catégories.

Remarque: pour les rapports sans noms explicites, le nom (de catégorie) est formé en reliant les différents noms de rôle (par ex. %RoleName1RoleName2%).

Les rôles sont considérés comme des attributs. Les rôles en eux-mêmes sont codés comme suit:

```

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>'.

```

Si le rapport est ordonné, un attribut XML supplémentaire («ORDER_POS»; défini dans l'annexe A) s'avère nécessaire par rapport à gml:ReferenceType.

Il n'est pas toujours possible de reconnaître si la référence indique un objet dans le même fichier de transfert (c'est uniquement possible lorsque la référence est un identificateur Fragment-XML relatif au document).

Exemple	
INTERLIS	<pre> CLASS Person = END Person; CLASS Verein = END Verein; ASSOCIATION DauerhafteVereinigungMitglied = DauerhafteVereinigung -- {0..*} Verein; Mitglied -- {1..*} Person; END DauerhafteVereinigungMitglied; </pre>
GML	<pre> <xsd:element name="Verein" type="VereinType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="VereinType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> <xsd:element name="Person" type="PersonType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="PersonType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> <xsd:element name="DauerhafteVereinigungMitglied" type="DauerhafteVereinigungMitgliedType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="DauerhafteVereinigungMitgliedType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> <xsd:element name="DauerhafteVereinigung" type="gml:ReferenceType"> </pre>

	<pre> <xsd:annotation> <xsd:appinfo> <gml:targetElement>Verein</gml:targetElement> </xsd:appinfo> </xsd:annotation> </xsd:element> <xsd:element name="Mitglied" type="gml:ReferenceType"> <xsd:annotation> <xsd:appinfo> <gml:targetElement>Person</gml:targetElement> </xsd:appinfo> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>
Instance	<pre> <Person gml:id="o1"/> <Verein gml:id="o2"/> </pre>
GML	<pre> <DauerhafteVereinigungMitglied gml:id=""> <DauerhafteVereinigung xlink:href="#o2"/> <Mitglied xlink:href="#o1"/> </DauerhafteVereinigungMitglied> </pre>

7.12 Codification des définitions de graphique

Pour chaque définition de graphique, les catégories de signature («Sign-ClassRef») référencées par la définition de graphique sont transmises lors du transfert. Les instances d'objet des catégories 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.

7.13 Codification des attributs

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

```

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	
INTERLIS	<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>
GML	<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:AbstractFeature"/> <xsd:complexType name="KoernungType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <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:extension> </xsd:complexContent> </xsd:complexType> </pre>

	<pre> </xsd:sequence> </xsd:extension> </xsd:complexContent> </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: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</p> <p>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>

7.14 Représentation des types INTERLIS sur des types de schéma XML

7.14.1 Aperçu

INTERLIS	Représentation	Type simple/complexe
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	Complex
OIDType	xsd:int / xsd:token	Simple
BlackboxType	xsd:anyType / xsd:base64Binary	Complex/Simple
ClassType	xsd:string	Simple
AttributePathType	xsd:string	Simple
LineType	gml:CurveType / gml:PolygonType	Complex

7.14.2 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:string` et «URI» en tant que `xsd:anyURI`. Dans le cas où ils sont dérivables du modèle, on code aussi `xsd:restriction`.

Exemple

INTERLIS	<pre> DOMAIN 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>

7.14.3 Codification d'énumérations

Dans INTERLIS, les énumérations peuvent être affinées et/ou élargies. 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 seulement 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:Code`.

Concernant l'identificateur des listes de codes (`codeSpace`) des différentes valeurs d'énumération, l'identificateur de l'espace de nom du modèle (`%StdModelNamespaceIdentifier%`; voir chapitre 7.6), complété par le nom représenté de la définition de domaine de valeur, peut être utilisé (exemple: «[Verein eCH, Mainaustrasse 30, Postfach, 8034 Zürich
T 044 388 74 64, F 044 388 71 80](http://www.interlis.ch/ILIGML-</p>
</div>
<div data-bbox=)

1.0/ModelA/Farbe»). Si l'énumération est définie dans le cadre d'un attribut, alors l'identificateur de l'espace de nom du modèle est complété par le nom de catégorie représenté et le nom d'attribut.

(Exemple: «http://www.interlis.ch/ILIGML-1.0/ModelA/Person/HaarFarbe»).

Le méta-attribut `ili2.iligml10.identifizierCodeSpace` 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.

Exemple	
INTERLIS	<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>
GML	<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" </pre>

	<pre><gml:Definition gml:id="o9"> <gml:identifrier codeSpace= "http://www.interlis.ch/ILIGML-1.0/ModelA/FarbePlus" >gruen</gml:identifrier> </gml:Definition> </gml:dictionaryEntry> </gml:Dictionary></pre>
Instance GML	<pre><Fahrzeug gml:id="o1"> <CarroserieFarbe>rot<CarroserieFarbe> </Fahrzeug> <Auto gml:id="o2"> <CarroserieFarbe>rot.dunkel<CarroserieFarbe> </Auto></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 aussi les noeuds), on applique la syntaxe pour les constantes d'énumération (règle «EnumValue»). Ce faisant, le caractère «#» est supprimé.

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

Variante: Représenter DomainDef/AttributeDef sans «FINAL» en tant que `xsd:string` sans `xsd:enumeration`

Variante: Représenter DomainDef/AttributeDef sans «FINAL» en tant que `xsd:union` de `xsd:string` et `xsd:enumeration`

Variante: Représenter DomainDef/AttributeDef sans «FINAL» en tant que `xsd:string` et `xsd:pattern` complexe

7.14.4 Codification de types de données numériques

Les types de données numériques sont codés en fonction de la valeur inférieur et de la valeur supérieur, en tant que `xsd:integer`, `xsd:decimal` ou `xsd:double` kodiert. 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> </xsd:simpleType> <xsd:simpleType name="TestDec"> <xsd:restriction base="xsd:decimal"> <xsd:minInclusive value="1.0"/> <xsd:maxInclusive value="10.0"/> </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>
-----	---

Variante: Codification en tant que `gml:MeasureType` dans le cas où une indication sur l'unité est présente dans le modèle.

Remarque: Il n'est pas possible d'indiquer l'unité dans le schéma. Avec «MeasureType», on peut seulement coder l'unité dans les données.

7.14.5 Codification de domaines de valeurs formatés

Les domaines de valeurs formatés, qui référencent «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 de valeurs formatés 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"> </pre>

	<pre> <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>
--	--

7.14.6 Codification des contenants

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

7.14.7 Codification de types de catégories

Le type de catégorie 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 intégral qualifié d'association, de structure ou de catégorie (par ex. «DM01AVCH24D.pointfixeCatégorie1.LFP1»).

	LKoord WITHOUT OVERLAPS>0.10;
GML	<pre> <xsd:complexType name="LKoord"> <xsd:complexContent> <xsd:restriction base="gml:PointPropertyType"> <xsd:sequence> <xsd:element ref="gml:Point"/> </xsd:sequence> </xsd:restriction> </xsd:complexContent> </xsd:complexType> <xsd:complexType name="GebaeudeFlaeche"> <xsd:complexContent> <xsd:restriction base="gml:SurfacePropertyType"> <xsd:sequence> <xsd:element ref="gml:Polygon"/> </xsd:sequence> </xsd:restriction> </xsd:complexContent> </xsd:complexType> </pre>

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

Dans le cas où des attributs de lignes apparaissent, les extensions correspondantes pour `gml:Curve` doivent être créées.

7.14.12 Codification des zonages

Pour chaque attribut avec un domaine de valeurs «AREA», un `gml:PointPropertyType` est codé: le point de référence de zone.

Pour chaque attribut avec un domaine de valeurs «AREA», un type de caractéristiques (Feature-Typ), est généré en supplément: le «tableau de ligne de bordure». Ce type de caractéristiques reçoit deux attributs: «geometry» du type `gml:CurvePropertyType` pour les morceaux de lignes séparés; et «lineattr» pour les attributs de ligne, dans le cas où le domaine de valeurs INTERLIS en contient.

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 = Grenze : MANDATORY AREA WITH (ARCS,STRAIGHTS) VERTEX LKoord WITHOUT OVERLAPS>0.10 LINE ATTRIBUTES GrenzlinieEigenschaften; END Grundstueck; </pre>
GML	<pre> <xsd:complexType name="LKoord"> <xsd:complexContent> </pre>

	<pre> <xsd:restriction base="gml:PointPropertyType"> <xsd:sequence> <xsd:element ref="gml:Point"/> </xsd:sequence> </xsd:restriction> </xsd:complexContent> </xsd:complexType> <xsd:element name="GrenzlinieEigenschaft" type="GrenzlinieEigenschaftType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="GrenzlinieEigenschaftType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> <xsd:element name="streitig" type="xsd:boolean" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> <xsd:element name="Grundstueck.Grenze" type="Grundstueck.GrenzeType" substitutionGroup="gml:AbstractFeature"/> <xsd:complexType name="Grundstueck.GrenzeType"> <xsd:complexContent> <xsd:extension base="gml:AbstractFeatureType"> <xsd:sequence> <xsd:element name="geometry" type="gml:CurvePropertyType"/> <xsd:element name="lineattr"> <xsd:complexType> <xsd:sequence> <xsd:element ref="GrenzlinieEigenschaft"/> </xsd:sequence> <xsd:attributeGroup ref="gml:OwnershipAttributeGroup"/> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> <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="Grenze" type="LKoord"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>
<p>Instance GML</p>	<pre> <Grundstueck.Grenze gml:id="o0"> <geometry> <gml:LineString gml:id="g0" srsName="urn:ogc:def:crs:EPSG::21781" srsDimension="2"> <gml:pos>146.92 174.98 </gml:pos> <gml:pos>163.64 185.96 </gml:pos> <gml:pos>158.15 194.31 </gml:pos> <gml:pos>149.79 188.82 </gml:pos> </gml:LineString> </geometry> </Grundstueck.Grenze> </pre>

	<pre> <gml:pos>147.04 193 </gml:pos> <gml:pos>138.68 187.51 </gml:pos> <gml:pos>146.92 174.98</gml:pos> </gml:LineString> </geometry> <lineattr> <GrenzlinieEigenschaft gml:id="s0"> <streitig>false</streitig> <GrenzlinieEigenschaft> </lineattr> </Grundstueck.Grenze> <Grundstueck gml:id="o1"> <Grenze> <gml:Point gml:id="g1" srsName="urn:ogc:def:crs:EPSG::21781" srsDimension="2"> <gml:pos>147.00 173.00 </gml:pos> </gml:Point> </Grenze> </Grundstueck> </pre>
--	--

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

7.14.13 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.iligml10.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.iligml10.identifieurPattern`» permet de définir le formatage de la valeur de l'identificateur (valeur pour l'élément XML `gml:identifieur` dans l'instance GML).

Le méta-attribut «`ili2.iligml10.hrefPattern`» permet de définir le formatage de la valeur de référence (valeur pour l'attribut XML `xlink:href` dans l'instance GML).

Dans les définitions-modèle, il est possible d'utiliser «`#{value}`» en tant qu'espace réservé pour la valeur OID.

Exemple	
INTERLIS	<pre> DOMAIN BFS_EGID = OID 1 .. 9999999999; UUID = OID TEXT*36; !!@ ili2.iligml10.identifieurCodeSpace= "http://inspire.jrc.ec.europa.eu/" !!@ ili2.iligml10.identifieurPattern= "urn:inspire:object:id:#{value}" !!@ ili2.iligml10.hrefPattern= "urn:inspire:object:id:#{value}" InspiredId = OID TEXT; !! z.B. "DENW1000:A00012ab" </pre>

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>
-----	---

Variante: coder en tant que `gml:CodeWithAuthorityType`

Variante: coder en tant que `xsd:anyURI`

7.14.14 Codification d'autres formes de pièces 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.

8 Règles de codification d'instance

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

8.1 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 l'élément XML «TRANSFER» en tant que racine. Cet élément et les types correspondants sont définis dans l'annexe A.

8.2 Codification des objets et éléments de structure

Les objets sont toujours codés en ligne pour un contenant. Les objets ne sont jamais codés en ligne pour un autre objet.

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`.

8.3 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.iligml10.identifierCodeSpace`», alors on utilise l'identificateur de l'espace de nom du modèle (`%StdModelNamespaceIdentifier%`; voir chapitre 7.6), en tant que «`codeSpace`», et 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-1.0/INTERLIS/UUIDOID`»).

Si, lors de la définition du domaine de valeurs de l'OID, une valeur a été définie pour le méta-attribut «`ili2.iligml10.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.iligml10.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.iligml10.identifierPattern`» alors celle-ci est utilisée.

8.4 Codification de rapports

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

8.5 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.

9 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'actua-

lité, 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.

10 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 pré-défini

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.interlis.ch/ILIGML-1.0/INTERLIS"
  xmlns:ili2="http://www.interlis.ch/ili2"
  targetNamespace=
    "http://www.interlis.ch/ILIGML-1.0/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>2005-06-16</ili2:modelVersion>
      <ili2:modelAt>http://www.interlis.ch</ili2:modelAt>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:import namespace="http://www.opengis.net/gml/3.2"/>

  <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="I320ID">
    <xsd:restriction base="xsd:int">
      <xsd:minInclusive value="0"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="STANDARD0ID">
    <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="UUI0ID">
    <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 - Exemple

INTERLIS 2-Modèle

```

INTERLIS 2.3;

MODEL Beispiel (de)
AT "mailto:ceis@localhost"
VERSION "2008-03-31" =

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 GebaeudeFlaeche =
        Gebaeude -- {0..*} Gebaeude;
        Flaeche -- {1} BoFlaechen;
    END GebaeudeFlaeche;

END Bodenbedeckung;

END Beispiel.

```

INTERLIS 1-Modèle

```

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-1.0/Beispiel"
  targetNamespace="http://www.interlis.ch/ILIGML-1.0/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>2008-03-31</ili2:modelVersion>
      <ili2:modelAt>mailto:ceis@localhost</ili2:modelAt>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:import namespace="http://www.opengis.net/gml/3.2"/>
  <xsd:complexType name="LKoord">
    <xsd:complexContent>
      <xsd:restriction base="gml:PointPropertyType">
        <xsd:sequence>
          <xsd:element ref="gml:Point"/>
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:element name="BoFlaechen.Form" type="BoFlaechen.FormType"
    substitutionGroup="gml:AbstractFeature"/>
  <xsd:complexType name="BoFlaechen.FormType">
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractFeatureType">
        <xsd:sequence>
          <xsd:element name="geometry"
            type="gml:CurvePropertyType"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <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="LKoord"/>
        </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="BoFlaechen.Form"/>
          <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/INTERLIS2.3">
<HEADERSECTION SENDER="ceis" VERSION="2.3">
  <MODELS>
    <MODEL NAME="Beispiel" VERSION="2008-03-31"
      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
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:ili="http://www.interlis.ch/ILIGML-1.0/INTERLIS"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  gml:id="tid">
  <ili:baskets>
  <ceis:Bodenbedeckung
    xmlns:ceis="http://www.interlis.ch/ILIGML-1.0/Beispiel"
    gml:id="bid">
    <ceis:member>
      <ceis:Gebaeude gml:id="x50">
        <ceis: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>
        </ceis:PositionHauseingang>
        <ceis:AssNr>958</ceis:AssNr>
        <ceis:Flaeche xlink:href="#x10"/>
      </ceis:Gebaeude>
    </ceis:member>
    <ceis:member>
      <ceis:Strasse gml:id="x40">
        <ceis: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>
  </ceis:Achse>
</ceis:Strasse>
</ceis:member>
<ceis:member>
  <ceis:BoFlaechen.Form gml:id="g1">
    <ceis:geometry>
      <gml:Curve gml:id="g1.1" srsName="urn:ogc:def:crs:EPSG::21781"
srsDimension="2">
        <gml:segments>
          <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>
    </ceis:geometry>
  </ceis:BoFlaechen.Form>
</ceis:member>
<ceis:member>
  <ceis:BoFlaechen.Form gml:id="g2">
    <ceis:geometry>
      <gml:Curve gml:id="g2.1" srsName="urn:ogc:def:crs:EPSG::21781"
srsDimension="2">
        <gml:segments>
          <gml:LineStringSegment interpolation="linear">
            <gml:pos>146.92 174.98 </gml:pos>
            <gml:pos>163.64 185.96 </gml:pos>
            <gml:pos>158.15 194.31 </gml:pos>
            <gml:pos>149.79 188.82 </gml:pos>
            <gml:pos>147.04 193 </gml:pos>
            <gml:pos>138.68 187.51 </gml:pos>
            <gml:pos>146.92 174.98</gml:pos>
          </gml:LineStringSegment>
        </gml:segments>
      </gml:Curve>
    </ceis:geometry>
  </ceis:BoFlaechen.Form>
</ceis:member>
<ceis:member>
  <ceis:BoFlaechen.Form gml:id="g3">
    <ceis:geometry>
      <gml:Curve gml:id="g3.1">
        <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:segments>
      </gml:Curve>
    </ceis:geometry>
  </ceis:BoFlaechen.Form>
</ceis:member>
</ceis:member>

```

```

        <gml:posList>194.26 208.19 186.38 206.82</gml:posList>
      </gml:LineStringSegment>
    </gml:segments>
  </gml:Curve>
</ceis:geometry>
</ceis:BoFlaechen.Form>
</ceis:member>
<ceis:member>
  <ceis:BoFlaechen.Form gml:id="g4">
    <ceis:geometry>
      <gml:Curve gml:id="g4.1">
        <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:segments>
      </gml:Curve>
    </ceis:geometry>
  </ceis:BoFlaechen.Form>
</ceis:member>
<ceis:member>
  <ceis:BoFlaechen gml:id="x30">
    <ceis:Art>humusiert</ceis:Art>
    <ceis:Form>
      <gml:Point gml:id="g8.1" srsName="urn:ogc:def:crs:EPSG::21781"
srsDimension="2">
        <gml:pos>148.20 183.48</gml:pos>
      </gml:Point>
    </ceis:Form>
  </ceis:BoFlaechen>
</ceis:member>
<ceis:member>
  <ceis:BoFlaechen gml:id="x20">
    <ceis:Art>befestigt</ceis:Art>
    <ceis:Form>
      <gml:Point gml:id="g20.1"
srsName="urn:ogc:def:crs:EPSG::21781" srsDimension="2">
        <gml:pos>168.27 170.85</gml:pos>
      </gml:Point>
    </ceis:Form>
  </ceis:BoFlaechen>
</ceis:member>
<ceis:member>
  <ceis:BoFlaechen gml:id="x10">
    <ceis:Art>Gebaeude</ceis:Art>
    <ceis:Form>
      <gml:Point gml:id="g10.1"
srsName="urn:ogc:def:crs:EPSG::21781" srsDimension="2">
        <gml:pos>133.95 206.06</gml:pos>
      </gml:Point>
    </ceis:Form>
  </ceis:BoFlaechen>
</ceis:member>
</ceis:Bodenbedeckung>
</ili:baskets>
</ili:TRANSFER>

```