DGIWG 208
Defence Geospatial Information Framework Encoding Specification: GML

Document type: Standard
Document date: 25 July 2024
Edition: 3.0.0
Responsible Party: Defence Geospatial Information Working Group (DGIWG)
Audience: This document is approved for public release and is available on the DGIWG website, http://www.dgiwg.org/dgiwg/.

Abstract: This document describes a schema using the Geography Markup Language for exchanging data for application schemas of the Defence Geospatial Information Model.

Table of Contents

List of Tables

List of Figures

i. Executive Summary

This standard is part of the Defence Geospatial Information Framework (DGIF), and works in concert with the Defence Geospatial Information Model (DGIM), the component of the DGIF specifying the logical data model. This standard described the use of Geography Markup Language (GML) a specialized encoding of XML schema supporting the encoding and exchange of geospatial information. This document provides detailed information concerning application schemas in the context of GML encoding and decoding. Examples encodings are provided in Annex A of this document.

ii. Submitting organizations

Nation	Organisation
United States	National Geospatial-Intelligence Agency

iii. Document point of contact

All inquiries are to be sent to secretariat@dgiwg.org.

iv. Revision history

Date	Release	Editors	Primary clauses modified	Description
2023-10-31	0.1	USA		Initial Revision
2023-11-01	0.2	USA, NLD	All	Updated Draft
2023-11-21	0.3	USA	All	Update for DGIWG Style Guide

v. Future work

As the DGIF programme of work continues and the business and technical process mature, the specification is expected to evolve and to be reviewed and updated accordingly.

1 Introduction

This specification is a supplement to the Defence Geospatial Information Framework (DGIF) and describes encoding for vector data in Geography Markup Language (GML) 3.2.1. It is the mandatory standard for XML encodings of datasets conforming to data product specifications (DPS) derived from DGIWG 205 - Defence Geospatial Information Model (DGIM).

2 Scope

This document defines and explains the GML application schema(s) for vector data in the DGIF.

3 Conformance

This specification conforms to the standards listed in section 4.

4 References

The documents listed in **Error! Not a valid bookmark self-reference.** are indispensable to understanding and using this standard. For dated references, only the cited edition or version applies. For undated references, the latest edition or version of the referenced document (including any amendments) applies.

Table 1: Normative References

Standard or Specification
ISO 19107:2003 – *Geographic information – Spatial schema*
ISO 19109:2005 – *Geographic information – Rules for application schema*
ISO 19125-1:2004 – *Geographic Information – Simple feature access – Part 1: Common architecture*
DGIWG 100 - *DGIWG 2D Spatial Schema Profile*
DGIWG 205 – *Defence Geospatial Information Model (DGIM) 3.0*
OGC 07-036 – *OpenGIS Geography Markup Language (GML) Encoding Standard*
OGC 09-025 - *Web Feature Service (WFS)*

The informative (non-normative) documents listed in Table 2 are useful to understanding and using this standard. For dated references, only the cited edition or version applies.

Table 2: Informative References

Standard or Specification
ISO/TS 19103:2005 – *Geographic information – Conceptual schema language*
ISO 19111:2003 – *Geographic information – Spatial referencing by coordinates*
ISO 19112:2003 – *Geographic information – Spatial referencing by geographic identifiers*
ISO 19115:2003 – *Geographic information – Metadata*
ISO 19123:2005 – *Geographic information – Schema for coverage geometry and functions*

Standard or Specification
ISO 19135:2005 – *Geographic information – Procedures for item registration*
ISO 19136:2007 – *Geographic information – Geography Markup Language (GML)*
ISO/IEC 19505-1:2012 – *Information technology – Object Management Group Unified Modeling Language (OMG UML) – Part 1: Infrastructure"*
ISO/IEC 19505-2:2012 – *Information technology – Object Management Group Unified Modeling Language (OMG UML) – Part 2: Superstructure*
ISO/IEC 19507:2012 – *Information technology – Object Management Group Object Constraint Language (OCL)*
DGIWG 200 – *Defence Geospatial Information Framework (DGIF) 3.0*
DGIWG 206 – *Defence Geospatial Feature Concept Dictionary (DGFCD) 3.0*
DGIWG 207 – *Defence Geospatial Real World Object Index (DGRWI) 3.0*
DGIWG 114 – *DGIWG Metadata Foundation 2.0.0*

5 Terms, definitions, and abbreviations

5.1 Definitions

Terms and definitions specific to this standard are given in Table 3.

Table 3: Definitions Applicable to this Standard

Term	Definition
Geospatial Information Model	A structured collection of feature information (features, attributes, associations, and ancillary data) whose metamodel conforms to the general feature model as specified in ISO 19109. EXAMPLE: Defence Geospatial Information Model
Application Schema	Conceptual schema for data required by one or more applications [ISO 19101]
Conceptual Model	Model that defines concepts of a universe of discourse [ISO 19101]
Feature	Abstraction of real world phenomena [ISO 19101] NOTE: A feature may occur as a type or an instance. Feature type or feature instance should be used when only one is meant.
Feature Attribute	Characteristic of a feature [ISO 19101]. NOTE 1: A feature attribute may occur as a type or an instance. Feature attribute type or feature attribute instance is used when only one is meant. NOTE 2: A feature attribute type has a name, a data type and a domain associated to it. A feature attribute instance has an attribute value taken from the domain of the feature attribute type.
Feature Concept / Data Dictionary	Dictionary that contains definitions of concepts that may be specified in detail in a feature catalogue [ISO 19126]
Feature Catalogue	Catalogue containing definitions of the feature types occurring in one or more sets of geographic data [ISO 19110]
Geographic Data	Data with implicit or explicit reference to a location relative to the Earth

Term	Definition
	NOTE: Geographic information is also used as a term for information concerning phenomena implicitly or explicitly associated with a location relative to the Earth.
Platform Independent Model	A model that is independent of the specific technological platform used to implement it.
Real World Object	An existing geographic (or geospatial) occurrence whose characteristics can be described/identified. EXAMPLE: A Wooden Bridge, A Mosque, A Divided Highway
Real World Object Tuple	A three element Feature Type-Attribute-Value combination used to describe a Real World Object
Universe of Discourse	View of the real or hypothetical world that includes everything of interest [ISO 19101]

5.2 Acronyms & Abbreviations

The acronyms and abbreviations used in this standard are specified in Table 4.

Table 4: Acronyms and Abbreviations Applicable to this Standard.

Acronym	Description
531 Code	A code consisting of 5 characters (2 letters and 3 numbers) for feature types, 3 characters for attributes, and 1 character for an attribute value. EXAMPLE; AA010, WID, 5
AlphaCode	A code consisting of the short name: upper camel case for feature types (e.g. GeneralBuilding) and lower camel case for attributes and values EXAMPLE: transportationSystemType.
AS	Application Schema
DGFCD	Defence Geospatial Feature Concept Dictionary
DGIM	Defence Geospatial Information Model
DGIWG	Defence Geospatial Information Working Group
DPS	Data Product Specification
EPSG	European Petroleum Survey Group Geodesy
GIS	Geographic Information System
GGDM	Groundwarfighter Geospatial Data Model
GML	Geography Markup Language
HTML	Hypertext Markup Language
ISO	International Organization for Standardization
OCL	Object Constraint Language
OGC	Open Geospatial Consortium
ONINA	Other, No Information, Not Applicable (reserved standard values)
UCUM	Unified Code for Units of Measure
UML	Unified Modeling Language
UoM	Unit of Measure
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

Acronym	Description
WFS	Web Feature Service
XML	eXtensible Markup Language
XSD	XML Schema Document
XSLT	Extensible Stylesheet Language Transformation

6 Application Schema

The following sections describe the use of an application schema (6.2), its role for data exchange (6.3), and the relationship between an application schema and the data captured on real world objects (6.4).

6.1 What is an Application Schema?

Per ISO 19109, an application schema represents a conceptual schema for data that is used by one or more applications. It defines data content, data structure, and operations for manipulating and processing the data[1].

6.2 Use of an Application Schema

An application schema defines a machine-readable format of the data structures that can be utilized by automated data management mechanisms. It can be used to create database structures and XML Schema (XSDs) to enables the transfer of data. It can also be used to document information relevant for an application domain providing for a common understanding of the data.

6.3 Application Schema for Data Exchange

An application schema describes the data that is relevant within an application domain. In this case, a Data Product Specification (DPS). Data publishers and users apply the DPS as the basis for a uniform exchange of this data. The publisher- and user-internal data format directly corresponds to a given application schema; however, the internal format can also differ. For example, a publisher may choose to store additional, internal information not represented by the application schema. Additionally, users might only have access to a subset of the data covered by the application schema.

It is important that a mapping exists between an internally-used application schema and the application schema that is used for data exchange. If such cases a user may request data from a publisher (e.g. via a web service interface[2]) and a publisher may provide data to the user. The data can then be transformed. Figure 1 depicts a notional example.

[1] The definition of operations in an application schema that has been created for data exchange is uncommon, and usually does not play a role for data exchange.

[2] Usually a web service interface based on the OGC Web Feature Service (WFS) standard is used for the management of vector data in spatial data infrastructures.

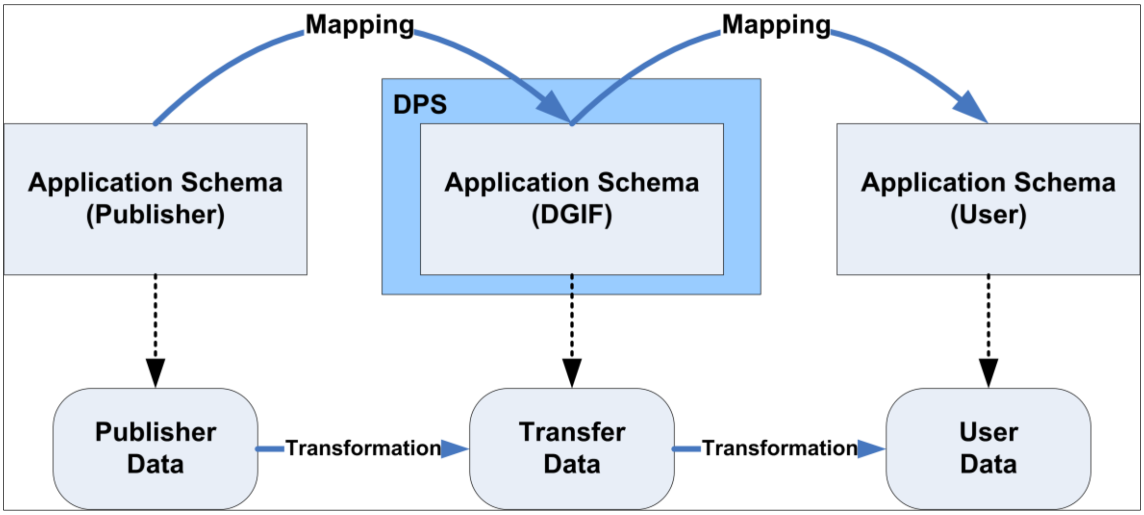

Figure 1: Data exchange via transformed data

Figure 2 depicts a notional data flow from a Ground-warfighter Geospatial Data Model (GGDM) product to the MGCP TRD4.

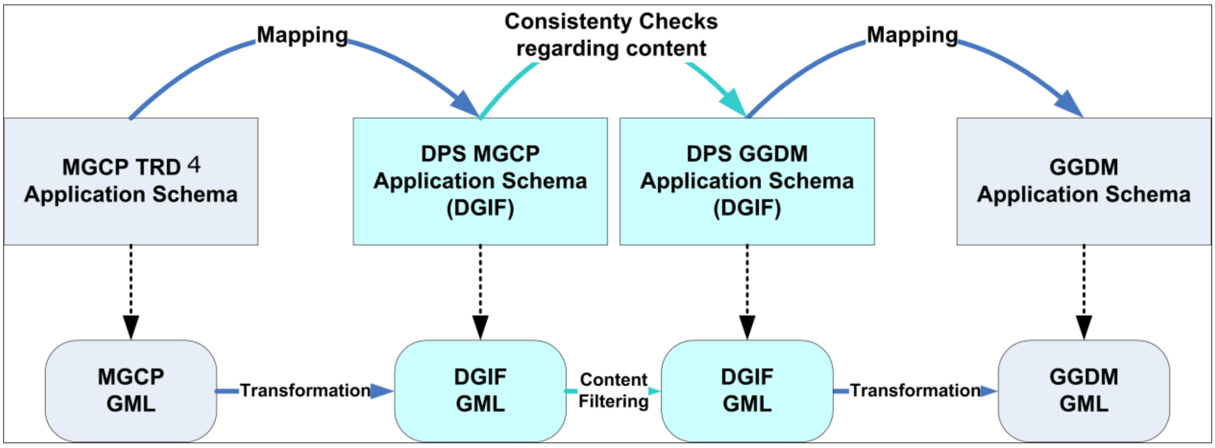

Figure 2: Example for data exchange from MGCP to GGDM.

As depicted above, a mapping defines how the content of an MGCP TRD4-dervided application schema is mapped to the content of the DGIF-derived DPS for MGCP. This is depicted as a conceptual process in the top portion of Figure 2.

As shown, content consistency checks are a required process and include a detailed comparison of the DGIF-derived MGCP DPS and a DGIF-derived GGDM DPS. These checks reveal what information and elements must be filtered[3].

A second mapping then enables a consistent transformation to the GGDM Application Schema as an end target.

The lower portion of Figure 2 depicts this same process as a GML transformation, content filtering, and a second GML transformation that would be implemented in a practical process.

[3] There may be content from other data sources necessary to provide all data for a full GGDM data set. This example is simplified.

MGCP information is encoded to GML according to an MGCP specification. An XSLT script implementing the mapping table is used to transform the encoded information into the DPS-MGCP GML format. This format can then be read by a GIS and the contained information can be stored as records in an Oracle database, which contains the information.

6.4 Features and Application Schema

To process information on objects of the real world, the relevant properties of these objects need to be identified first, and digitized via a conceptual model[4], the Defence Geospatial Information Model (DGIM). Hence the DGIM represents the universe of discourse for the military geospatial domain.

There are two basic types of digital representations of geodata: vector data and raster data. For this document only vector data is relevant. Vector data enables the combination of spatial, temporal, and thematic information of individual objects. Objects of the same type belong to a specific *class* of objects, which, in the context of application schema, is also called a f*eature type* (or other entities).

An application schema defines the relevant information of one or more feature types. The logical structure of data collected for an object follows the structure defined by the application schema for the according feature type. The actual physical data representation follows an implementation schema (and artefacts derived from it, for example XML Schema) that has been derived from the conceptual application schema.

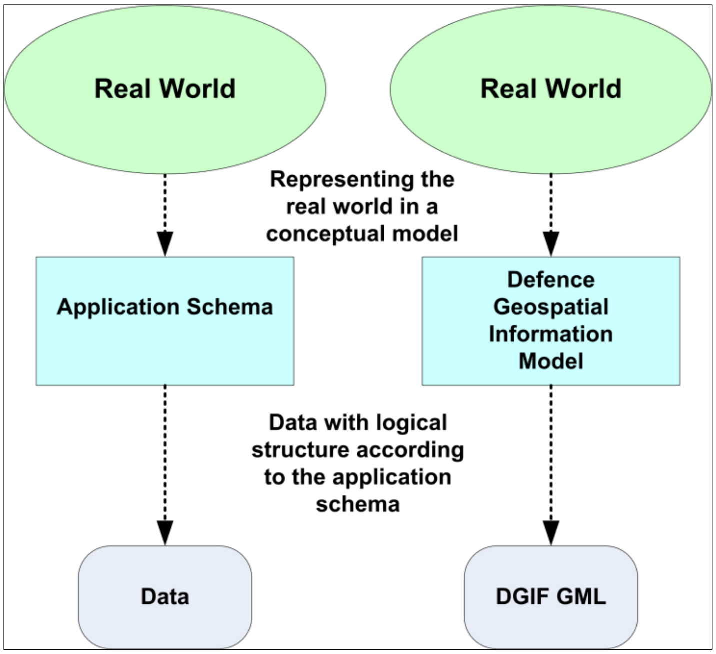

Figure 3: Representing real world objects according to an application schema

A well-defined application schema enables a consistent view of the real world objects that are relevant for an application. Data collected or transformed according to the DGIF application schema represent a homogeneous dataset that can be provided to both a publisher and a user of specific products. The dataset can thereby be tailored, i.e. subsetted via a selection, as required by the given use case.

[4] The conceptual model is usually represented by a UML model.

7 Structure of a DGIM Product Specification Application Schema

The application schema belonging to a DPS consists of feature types and associated properties. The conceptual schema language used to describe the application schema is Unified Modeling Language (UML). Figure 4 depicts an example UML diagram of the waterfall concept, with classes, properties, and relationships depicted with standard symbology, syntax, and notation. UML is described in greater detail in DGIWG 205 - Defence Geospatial Information Model (DGIM) 3.0.

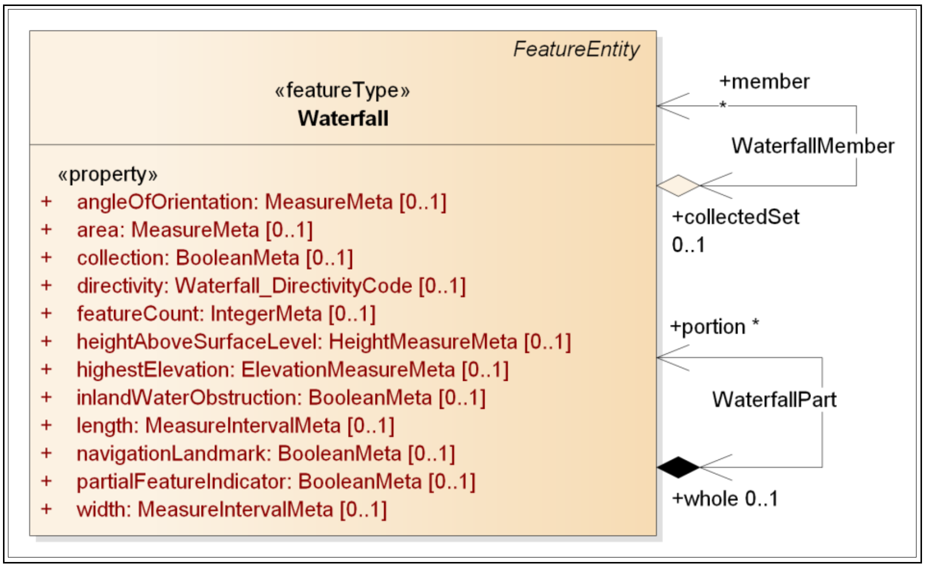

Figure 4: UML diagram of the feature type “Waterfall“

7.1 Entities

An entity represents a class of things or objects of the real world (Feature Types). Examples are waterfall, tunnel, stadium, and climate zone. They can also be additional (non-spatial) information (Type) or an event (Event Type) or actor (Actor Type).

Feature types (usually) have a spatial property (geometry). They also have thematic and, if applicable, temporal properties. Examples are name, colour, and construction date.

Properties have a certain data type (see section 7.3). A feature type may allow multiple values for a property. A feature can, for example, have multiple names. The “multiplicity” of a property defines how many values a feature can or must have for that property.

7.2 Concept Definitions

All feature types and properties in the DGIM have a formal definition and description that specifies the meaning of a given concept. These definitions and descriptions are stored and managed in the Defence Geospatial Feature Concept Dictionary (DGFCD).

7.3 Data Types

Each property of a feature type has a specific data type. The following sections briefly describe the data types used in the DGIM and relevant aspects.

7.3.1 Basic Data Types

7.3.1.1 Text

(Class) Name in the conceptual model: CharacterString

A property of type text can carry any text as its value.

7.3.1.2 Boolean

(Class) Name in the conceptual model: Boolean

The value of a boolean property is either “true” or “false”.

7.3.1.3 Numeric Values

7.3.1.3.1 Integer

(Class) Name in the conceptual model: Integer / IntegerInterval

Properties of type integer are used when the property is countable but may also carry a negative value.

Under specific circumstances an integer property can also be represented as an interval of two numbers, for example the *population* of an entity *country*.

7.3.1.3.2 Rational Number

(Class) Name in the conceptual model: Real / RealInterval

A rational number is often represented by a floating-point number.

Note that the DGIM differentiates between “simple” rational numbers and measures (see 7.3.1.3.3). A property of type rational number does not carry a unit of measure for its value.

Under specific circumstances a real property can also be represented as an interval of two floating-point numbers, for example the *unemployment rate* of an entity *country*.

7.3.1.3.3 Measure

(Class) Name in the conceptual model: Measure / MeasureInterval

This data type is used for properties that are being measured, for example length, speed, or frequency. In addition to a rational number that represents the measured value, a measure also contains the unit of measure that the value is given in.

Under specific circumstances a measured property can also be represented as an interval of two measures, for example the *depth* of an entity *Lake*.

7.3.2 Enumeration

A property of type enumeration carries a value from a closed list of values.

The user chooses one (or more) of these values and assigns it as value(s) for the property of a feature.

7.3.3 Geometry

All feature types can be represented with different geometry types (for example point, curve, surface, multi surface, or solid).

7.3.4 Null Values (ONINAs)

If the value of a property is not known, a standard value can be set to express this situation. The DGIM recognizes three different cases:

- Other (O) – None of the values listed for the feature property applies, but an unlisted one. The actual value should be documented in another property or an external document.
- No Information (NI) – The feature that the property belongs to was recorded but no value could be determined for the property.
- Not Applicable (NA) – The property does not apply, i.e. it does not exist for the feature.

The following tables define how ONINA values are mapped in the four different variants of GML schema available as exchange formats for DPS (GML 3.2, GML 2.1, complex and flattened).

Table 5: Mapping of ONINA values in GML application schema (complex form)

ONINA Value	Mapping in GML 3.2 (complex)
noInformation	https://www.dgiwg.org/codelist/*{codelist-Identifier}*/noInformation
notApplicable	https://www.dgiwg.org/codelist/*{codelist-Identifier}*/notApplicable
other	https://www.dgiwg.org/codelist/*{codelist-Identifier}*/other

Table 6: Mapping of ONINA values in GML application schema (flattened form)

Data Type	ONINA Value		
	noInformation	notApplicable	other
Numeric	-999999	-32765	
Text	No Information	Not Applicable	
Codelist	noInformation	notApplicable	other
Enumeration	noInformation	notApplicable	other
Boolean	-999999		
Geometry	If an actual geometry is not available then the according GML Schema element is not encoded (i.e. omitted).		

Note: If an XML instance of a GML Schema does not contain an element for a specific property[5] it is assumed that no information is available for this property. Thus implicitly the ONINA value noInformation is assumed as property value. This applies both in complex and flattened GML Schema.

[5] In this case the property is represented in the GML Schema by an optional element.

7.3.4.1 Constraints

Certain textual property values must comply with specific expressions/patterns (for example dates and time). In addition, the length of text values may be restricted, and limits may exist for numeric values.

The conceptual model of the DGIM represents these restrictions via constraints. They need to be adhered to during data collection and can possibly be automatically checked for a given data record by Schematron[6].

7.4 Metadata

The DGIM defines metadata for both feature types and their properties (more specifically: each property value).

Feature types inherit common metadata properties from the supertype Entity via the entity FeatureGAMetadata, while metadata for property values is defined by the supertype TypeMeta via the entity FeatureAttGAMetadata, which is common to all data types (see

).

Section 8 describes how metadata is used in GML application schema.

A detailed overview on the metadata elements can be found in DGIWG 205 - Defence Geospatial Information Model (DGIM) 3.0.

[6] For XML content data referring to a complex GML 3.2.1 application schema of a DPS a Schematron schema is available. This can be used to check that OCL constraints defined in the UML model of the DPS application schema are adhered to.

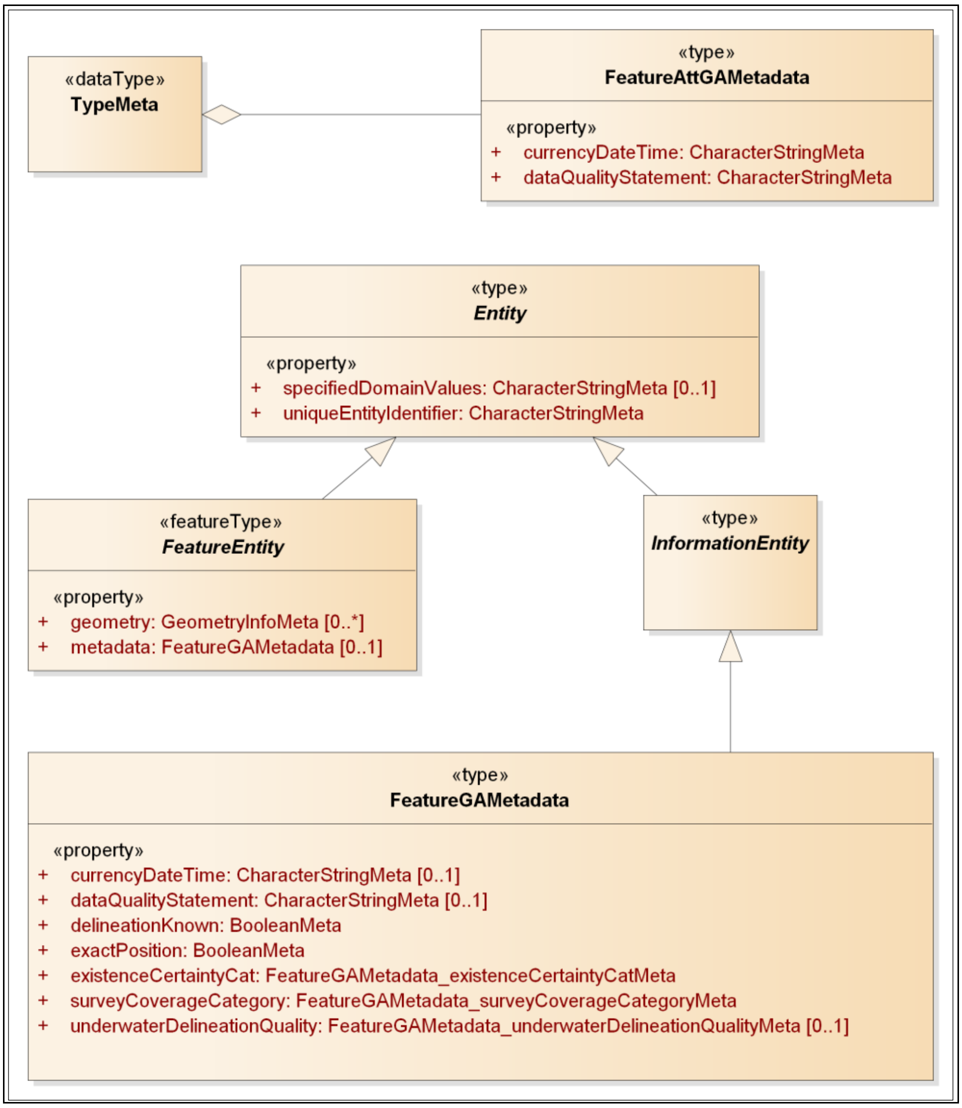

Figure 5: Metadata on feature level (via common supertype FeatureEntity) and property level (via supertype TypeMeta, common to all datatypes)[7]

8 GML Application Schema

This section describes the exchange formats of the DPS GML application schema as well as data exchange based on these schemas.

8.1 Target Namespaces

The following target namespaces are relevant for GML 3.2.1 based DPS application schema:

- GML (3.2.1) DGIF application schema, with {DPS-Identifier} replaced by the English short name of the DPS, for example "DPS-Exchange50kTopoVector" for the data product specification for an exchange product for 1:50k vector data:

[7] Some of the attributes shown in the UML diagram – for example the attribute "classification" of type "ClassificationInfo" – do not have an explicit cardinality. According to the UML specification the default cardinality of such attributes is "1".

  - in complex form:
    http://www.dgiwg.org/xmlns/DGIF/{DPS-Identifier}/{Version-Number}

  - in flattened form:
    http://www.dgiwg.org/xmlns/DGIF/{DPS-Identifier}/{Version-Number}/flat

- Geography Markup Language (GML) – version 3.2.1: *http://www.opengis.net/gml/3.2*

8.2 Schematron Schema

In addition to XML Schema files, Schematron files are available for the complex form of the GML 3.2.1 application schema. The Schematron files support additional consistency checks (of XML instance documents), based upon constraints defined in the application schema (see section 7.3.4.1). They are named as the application schema followed by the term "schtron", for example "DPS-Exchange50kTopoVector_schtron".

8.3 Configurations

The GML application schema is highly configurable so that specific configuration can be provided depending on the use case, the supported product or the limitations of existing systems. The section will be under current review and specific configurations will be added when necessary.

The complex form represents a direct derivation of the DGIM structures. These structures have been considerably simplified in the flatted form to enable data exchange via certain file formats (e.g. Shapefiles).

8.3.1 Complex-AlphaCode

The complex application schema uses the whole GML specification to cover all structures described in the DGIM. This includes information entities, attribute metadata and associations between features.

8.3.2 Flat-531 Code

Complex structures from the DGIM application schema have been simplified in the flattened form of GML application schema. The following list provides an overview of the changes:

- Attribute metadata is not supported in the flattened form.

- Breaking up ONINA structures: a special property value provides the reason why an actual value is not available. This approach does not work for complex data types, especially geometry types. The minimum cardinality of geometry property types is therefore set to 0. If a geometry is not available then neither a geometry value is provided nor a reason why.

- Breaking up inheritance hierarchies: properties of a supertype are copied to its subtypes.

- Breaking up properties that can have multiple values: instead of a potentially unlimited number of property values, only a fixed number of values is supported.

- The content of complex data types (except geometries) or associated info classes is transferred into a simple, list-like structure that uses basic data types.
- Associations between featureTypes are transformed to properties of the featureTypes.
- Feature types and their properties are identified via their 531 codes, instead of their AlphaCodes.
- Geometry types are restricted to ISO 19125-1 Simple Feature geometry types: point, line, surface. Aggregates are supported for line and surface types (see section 8.4.5). With this restriction the flattened form only supports 2D geometries and linear interpolation.
- Feature types are separated and identified based upon the dimension of their geometries: feature types in the complex form can use different geometries. In the flattened form, each feature type is separated into three classes, each of which has a specific geometry type (identifiable via the class name suffix: "_P" for point geometries, "_C" for line geometries, and "_S" for surface geometries).
- Constraints can no longer be automatically checked.
- Measures are encoded as rational numbers. The unit of measure recommended by the application schema applies.

Another effect of the flattening process is that Info-classes are broken up. In the complex form of GML application schema instances of these classes can be (re-) used by multiple feature instances (see section 7.4). This is not possible in the flattened form. Properties of Info-class instances are incorporated in the flattened representation of a feature without identifying the source Info-class.

8.3.3 Flat-AlphaCode

The Flat-AlphaCode is equal to the Flat-531 Code described in section 8.3.2 but is using alphaCodes for feature, attribute and enum identification.

8.4 Encoding

This section explains – via an example – how information can be exchanged using a GML application schemas for an urban scale DPS. This example shows how to encode and decode information about a waterfall (see section 8.5). This covers the transformation phases required for exchanging data between a publisher and a user (see section 6.3).

8.4.1 Initial Data

The following information is known about the waterfall which is used as an example in this section:

- Object metadata:
    - Naming:
        - The name of the waterfall is: Röthbachfall
        - The waterfall does not have a specific tactical use.

- The military restriction of the feature is: unclassified
- Identification:
  - The identifier of the waterfall is: rothbachfall
  - The numeric identifier is: 74831414
- Additional information about the waterfall is available at the following external sources:
  - http://en.wikipedia.org/wiki/Röthbach_Waterfall
  - http://www.worldwaterfalldatabase.com/waterfall/Rothbachfall-42/
- Reliability: the waterfall definitely exists.
- Source: data about the waterfall was collected by agency XYZ

- Application data:
  - The position of the waterfall is:
    - Geodetic latitude: 47.501061 decimal degrees
    - Geodetic longitude: 13.012573 decimal degrees
    - in WGS 84 2D coordinate reference system
    - Information on latitude and longitude is approximate.
  - Base elevation is 780 meters.
  - The height (above surface level) of the waterfall is between 468 and 471 meters.
    - Attribute metadata: source of this information is agency ABC; the information was collected on May 15th, 2011.
  - The highest elevation is 1250 meters.
  - The vertical datum is mean sea level.

There is no additional information.

8.4.2 Encoding - Complex-AlphaCode

The basic framework of an XML instance for encoding the waterfall feature (in GML complex form) is as follows:

```
<Waterfall gml:id=”ID001” xmlns=”http://www.dgiwg.org/xmlns/DGIF/{DPS-Identifier}/{Version-Number}”
 xmlns:gco=”http://www.isotc211.org/2005/gco” xmlns:gmd=”http://www.isotc211.org/2005/gmd”
 xmlns:gml=”http://www.opengis.net/gml/3.2”
 xmlns:gmldgiwgsp=”http://www.dgiwg.org/xmlns/GML/3.2/profiles/spatial/1.0/”
 xmlns:gmlexr=”http://www.opengis.net/gml/3.3/exr” xmlns:gsr=”http://www.isotc211.org/2005/gsr”
 xmlns:gss=”http://www.isotc211.org/2005/gss” xmlns:gts=”http://www.isotc211.org/2005/gts”
 xmlns:xlink=”http://www.w3.org/1999/xlink” xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”>

</Waterfall>
```

- The namespace (attribute: xmlns) contains the target namespace of the GML schema of the DPS. If the data shall be encoded according to a different feature catalogue, the namespace needs to be adjusted accordingly.

- The other namespaces should be (re-) used as is.
- The attribute gml:id contains an identifier of the waterfall, which is unique within the XML instance (at least).

Now the available information about the waterfall can be added. The order and structure in which the information needs to be added is specified by the GML schema of the chosen feature catalogue.

First of all the information on military restrictions is encoded (and added one level below the <Waterfall> element – which is the same for the following information items):

```
<classification>
 <ClassificationInfo gml:id="ID002">
  <securityClassification>unclassified</securityClassification>
 </ClassificationInfo>
</classification>
```

- The ClassificationInfo element is marked with a unique ID (ID002), just like the Waterfall element. This applies to all Info elements.
- The value of the securityClassification property is added to the according element.
- All property values of the classes ClassificationInfo, RestrictionInfo, ReliabilityInfo, IdentificationInfo, and SourceInfo are encoded without additional dataType/union elements. The encoding with dataType/union elements is shown in the following example (where information on naming is encoded)[8].

Naming information is encoded as follows:

```
<names>
 <GeoNameInfo gml:id="ID003">
  <name>
   <CharacterStringMeta>
    <valueOrReason>Röthbachfall</valueOrReason>
   </CharacterStringMeta>
  </name>
  <tacticalUsage>
   <BooleanMeta>
    <valueOrReason>false</valueOrReason>
   </BooleanMeta>
  </tacticalUsage>
 </GeoNameInfo>
</names>
```

- Again an ID is assigned to the Info element.
- The name of the waterfall is encoded via a nested structure of CharacterStringMeta and valueOrReason elements. This structure supports the addition of attribute value metadata, which is shown later on when the height of the waterfall is encoded. The tactical use is encoded in the same way (with the Meta element using the according data type; here: boolean).

[8] The different approach is caused by the modeling of Info classes in the „Metadata“ package of the application schema. The properties – more specifically the values of these properties – of ClassificationInfo, RestrictionInfo, SourceInfo, IdentificationInfo, and ReliabilityInfo classes cannot contain metadata. Nested attribute metadata structures are thus avoided.

Following the structure specified in the GML schema, the geometry information of the waterfall is now encoded:

```xml
<geometry>
 <GeometryInfoMeta>
  <valueOrReason>
   <PointGeometryInfo gml:id=”ID004”>
    <horizCoordMetadata>
     <HorizCoordMetadata>
      <horizAccuracyCategory>
       <HorizAccuracyCategory_HorizCoordMetadataMeta>
        <valueOrReason>approximate</valueOrReason>
       </HorizAccuracyCategory_HorizCoordMetadataMeta>
      </horizAccuracyCategory>
     </HorizCoordMetadata>
    </horizCoordMetadata>
    <representationScale>
     <IntegerMeta>
      <valueOrReason nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763” xsi:nil=”true”/>
     </IntegerMeta>
    </representationScale>
    <vertCoordMetadata>
     <VertCoordMetadata>
      <elevationAccuracyCategory>
       <ElevationAccuracyCategory_VertCoordMetadataMeta>
        <valueOrReason>approximate</valueOrReason>
       </ElevationAccuracyCategory_VertCoordMetadataMeta>
      </elevationAccuracyCategory>
     </VertCoordMetadata>
    </vertCoordMetadata>
    <angleOfOrientation>
     <MeasureMeta>
      <valueOrReason nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=202222” uom=”nil”
       xsi:nil=”true”/>
     </MeasureMeta>
    </angleOfOrientation>
    <area>
     <MeasureMeta>
      <valueOrReason nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=202222” uom=”nil”
       xsi:nil=”true”/>
     </MeasureMeta>
    </area>
    <geometry>
     <gml:Point gml:id=”ID005” srsName=”http://www.opengis.net/def/crs/EPSG/0/4326”>
      <gml:pos>47.501061 13.012573</gml:pos>
     </gml:Point>
    </geometry>
    <length>
     <MeasureIntervalMeta>
      <valueOrReason nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=202222” xsi:nil=”true”
      />
     </MeasureIntervalMeta>
    </length>
    <width>
     <MeasureIntervalMeta>
      <valueOrReason nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=202222” xsi:nil=”true”
      />
     </MeasureIntervalMeta>
    </width>
```

```
    </PointGeometryInfo>
   </valueOrReason>
  </GeometryInfoMeta>
</geometry>
```

- The information is represented via a PointGeometryInfo.
- That latitude and longitude values are approximate is defined via the elements horizAccuracyCategory and elevationAccuracyCategory.
- Further information on positional accuracy is not available. However, in the GML schema the element representationScale is mandatory. Because no information is available for this property, the valueOrReason element is encoded as follows:
    - The attribute xsi:nil="true"tells the XML processor that no content is available for the element. This ensures that the XML instance is valid, even though the respresentationScale element is not defined as optional in the GML schema.
    - The nilReason attribute points out the reason why no actual representationScale value is provided. Section 7.3.4 defines the pointers/references that shall be used to identify ONINA values in GML schema (complex form).
- The encoding of the point geometry follows the GML standard – see section 8.4.5.
- The encoding of other elements that are required by the GML schema, but for which no information is available or which are not applicable, is achieved via ONINA values as explained before.

  However, this example points out another aspect: because the attribute "uom" in the Measure/valueOrReason element is mandatory, a value must be provided in order for the XML instance to be valid. In this case the actual value is irrelevant (and was set to "nil" in the example), because the valueOrReason element provides an ONINA value via the xsi:nil and nilReason attributes.

Now the identification information is encoded:

```
<identification>
  <IdentificationInfo gml:id="identInfo_01">
    <genericEntityIdentifier> roethbachfall</genericEntityIdentifier>
    <numericFeatureIdentifier>74831414</numericFeatureIdentifier>
  </IdentificationInfo>
</identification>
```

There is nothing noteworthy here.

Links to further information are encoded next:

```
<informationReference>
  <CharacterStringMeta>
    <valueOrReason>http://en.wikipedia.org/wiki/Röthbach_Waterfall</valueOrReason>
  </CharacterStringMeta>
</informationReference>
<informationReference>
  <CharacterStringMeta>
    <valueOrReason>http://www.worldwaterfalldatabase.com/waterfall/Rothbachfall-42/</valueOrReason>
  </CharacterStringMeta>
</informationReference>
```

- This is an example of a property (informationReference) with maximum cardinality greater than 1. Multiple property values are represented via multiple elements located on the same level. Also see the cardinality indicators of elements in the GML schema.

Reliability information is encoded as follows:

```
<reliability>
  <ReliabilityInfo gml:id="reliabInfo_01">
    <existenceCertaintyCat>definite</existenceCertaintyCat>
    <existenceAssessment xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
  </ReliabilityInfo>
</reliability>
```

Information from individual objects, for example Info class instances, can also be referenced via xlink:href attributes:

```
<reliability xlink:href="http://my.service.org/reliabInfo_01"/>
```

Depending upon the use case this approach can save storage space and bandwidth when exchanging data. However, it assumes that the referenced storage location is accessible.

Information on restrictions is not available (in this example). Nevertheless, because the schema requires the according properties, they are encoded as follows:

```
<restriction>
  <RestrictionInfo gml:id="restrInfo_01">
    <commercialCopyrightNotice xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <commercialDistribRestrict xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <ownerAuthority xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <releasability xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <releasabilityRestriction xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
  </RestrictionInfo>
</restriction>
```

Now further object metadata, information about the source of the data, is encoded:

```
<source>
  <SourceInfo gml:id="sourceInfo_01">
    <sourceAgency xlink:href="http://register.org/XYZ" xlink:title="XYZ"/>
    <sourceDateTime xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <sourceDescription xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <sourceType xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <productionAuthority xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <creationDateTime xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <updateReason xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <extractionSpec xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <dataCaptureSpec xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
  </SourceInfo>
</source>
```

- For most of the required properties no information is available. Information about the agency that collected the data can be provided, though: agency XYZ. The data type of property sourceAgency is a codelist, more specifically the codelist “OrganizationCode”. This codelist is managed outside of the application schema. The value of the sourceAgency element in this example points to an entry in a representation of the codelist.

Now further application data is encoded, starting with the base elevation:

```
<baseElevation>
  <MeasureMeta>
    <valueOrReason uom=”m”>780</valueOrReason>
  </MeasureMeta>
</baseElevation>
```

- The base elevation is given via a measure (see section 7.3.1.3.3). The measure value is 780 meters. The unit of measure is given via a UCUM code[9].

Information about the height of the waterfall is encoded as follows:

```
<heightAboveSurfaceLevel>
  <MeasureIntervalMeta>
    <attributeMetadata>
      <FeatureAttMetadata gml:id=”ID006”>
        <source>
          <SourceInfo gml:id=”ID007”>
            <sourceAgency xlink:href=”http://register.org/ABC” xlink:title=”ABC”/>
            <sourceDateTime>2011-05-15</sourceDateTime>
            <sourceDescription nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763” xsi:nil=”true”/>
            <sourceType>mapChartOrGeodeticData</sourceType>
            <productionAuthority nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763” xsi:nil=”true”/>
            <creationDateTime>2013-10-15T09:11:00Z</creationDateTime>
            <updateReason nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763” xsi:nil=”true”/>
            <extractionSpec nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763” xsi:nil=”true”/>
            <dataCaptureSpec nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763” xsi:nil=”true”/>
          </SourceInfo>
        </source>
      </FeatureAttMetadata>
    </attributeMetadata>
    <valueOrReason>
      <MeasureIntervalUnion>
        <intervalValue>
          <MeasureInterval>
            <lowerValue uom=”m”>468</lowerValue>
            <upperValue uom=”m”>471</upperValue>
          </MeasureInterval>
        </intervalValue>
      </MeasureIntervalUnion>
    </valueOrReason>
  </MeasureIntervalMeta>
</heightAboveSurfaceLevel>
```

- A new aspect in this example is the use of a MeasureInterval.
- In addition to the actual value of the property heightAboveSurfaceLevel, metadata for the value itself is provided, in this case the source of the value.

The encoding of highest elevation and vertical datum does not show further specifics:

```
<highestElevation>
  <MeasureMeta>
    <valueOrReason uom=”m”>1250</valueOrReason>
  </MeasureMeta>
</highestElevation>
<verticalDatum>
  <VerticalDatum_WaterfallMeta>
```

[9] see http://unitsofmeasure.org

```
    <valueOrReason>meanSeaLevel</valueOrReason>
  </VerticalDatum_WaterfallMeta>
</verticalDatum>
```

The complete XML instance from this example is contained in an Annex (see section A.1).

8.4.3 Encoding - Flat-531 Code

The basic framework of an XML instance for encoding the waterfall feature (in GML- flattened form) is as follows:

```
<BH180_P gml:id="ID001" xmlns="http://www.dgiwg.org/xmlns/DGIF/{DPS-Identifier}/{Version-Number}/flat"
  xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
</BH180_P>
```

- Note that the choice of the root element depends on the geometry of the object. Here we use the 531 code of the Waterfall feature type with suffix "_P", because the feature has a point geometry. Other suffixes are "_C" for line geometries and "_S" for surface geometries.
- The namespace (attribute: xmlns) contains the target namespace of the GML schema for a flattened form urban scale. If the data shall be encoded according to a different feature catalogue, the namespace needs to be adjusted accordingly.
- The other namespaces should be (re-) used as is.
- The attribute gml:id contains an identifier of the waterfall, which is unique within the XML instance (at least).

Now the available information about the waterfall can be added. The order and structure in which the information needs to be added is specified by the GML schema of the chosen feature catalogue.

First of all the information on military restrictions is encoded (and added one level below the <BH180_P > element – which is the same for the following information items):

```
<sec>34</sec>
```

- All properties of the waterfall are identified via their 531 code. Due to the flattening process, element names with a concatenation of 531 codes can result. This will become more apparent in the following encoding examples.

  In order to easily identify the 531 code that belongs to a property the HTML documentation of the feature catalogue can be consulted. The documentation of the GML schema element also provides a hint about the AlphaCode of the element, in case that the name of that element is a 531 code.
- The data type used to denote the military classification, more specifically the security classification, is an enumeration. The enumeration value also uses a 531 code (in this case: "34"). In order to identify the correct code the HTML documentation of the feature catalogue can be consulted, as well as the documentation of the "sec" element in the GML schema.
- Another aspect of this example is that the identity of Info class instances, which is represented in the complex form of the GML schema, is not represented in the flattened form.

Naming information is encoded as follows:

```
<nam_1>Röthbachfall</nam_1>
<tru>1000</tru>
```

- The name of the waterfall is encoded via the nam_1 element. The suffix "_1" indicates that the element contains one of a constrained list of possible values. In the complex form, the number of values for the property "name" is unlimited. In the flattened form, however, only a fixed number of names can be provided. At the moment eight values are possible, which are represented accordingly in the GML schema via the elements nam_1, nam_2, …, nam_8.
- The tactical use is provided via the data type boolean. In the flattened form a 531 code is used to represent the boolean value (1000 for "false" and 1001 for "true") – just like for enumerations.

Following the structure specified in the GML schema, the geometry information of the waterfall is now encoded:

```
<xg1_1-zi007.acc>2</xg1_1-zi007.acc>
<xg1_1-zi007.ela>2</xg1_1-zi007.ela>
<xg1_1-zi007.aoo>-32765</xg1_1-zi007.aoo>
<xg1_1-zi007.ara>-32765</xg1_1-zi007.ara>
<xg1_1-zi007.xg1>
  <gml:Point gml:id="ID005" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
   <gml:pos>47.501061 13.012573</gml:pos>
  </gml:Point>
</xg1_1-zi007.xg1>
<xg1_1-zi007.len-s>-32765</xg1_1-zi007.len-s>
<xg1_1-zi007.wid-s>-32765</xg1_1-zi007.wid-s>
```

- Apparently the elements are a result of flattening a complex data structure:
  - „xg1" represents the „geometry" property of the feature type (here: Waterfall).
  - Again, „_1" indicates that the geometry is a property for which multiple values can be provided. Because this example only uses one point, the first element of the list of available geometry elements is used.
  - „-zi007" is the 531 code of the actual GeometryInfo type that represents the geometry information. In this example the PointGeometryInfo type is used.
  - Elements separated by „." are properties of the PointGeometryInfo class, identified by their 531 code.
  - To identify the 531 code the HTML documentation of the feature catalogue can be used, as well as the documentation of the elements in the GML schema (flattened form).
  - Many elements (for example xg1_1-zi007.aoo) contain the value -32765. It indicates that an actual value does not apply (Not Applicable) for the according property. Section 7.3.4 defines which code values shall be used to represent ONINA values in GML schema (flattened form).
  - The example contains another special case: if a property of data type MeasureInterval (here for example the geometry length) shall contain an ONINA value, then the value can be provided via the "single" option. This option is indicated via the "-s" in the element xg1_1-zi007.len-s. If a MeasureInterval

actually contains an interval, then “-i” is used instead[10]. This is shown in a following example (where the height of the waterfall is encoded).

- That the latitude and longitude values are approximate is defined via the elements xg1_1-zi007.acc and xg1_1-zi007.ela. The data types are enumerations. In the flattened form enumeration values are given via 531 codes (here “2” represents “approximate”).
- The encoding of the point geometry follows the GML standard – see section 8.4.5.

Now the identification information is encoded:

```
<bgi>roethbachfall</bgi>
<uid>74831414</uid>
```

There is nothing noteworthy here.

Links to further information are encoded next:

```
<hyl_1>http://en.wikipedia.org/wiki/Röthbach_Waterfall</hyl_1>
<hyl_2>http://www.worldwaterfalldatabase.com/waterfall/Rothbachfall-42/</hyl_2>
```

There is also nothing noteworthy here.

Reliability information is encoded as follows:

```
<coe>1</coe>
```

Information on restrictions is not available (in this example). Nevertheless, because the schema requires some of the according properties, they are encoded as follows

```
<ccn>No Information</ccn>
<cdr>No Information</cdr>
<ona_1>No Information</ona_1>
<qle>No Information</qle>
```

Now further object metadata, information about the source of the data, is encoded:

```
<sag>XYZ</sag>
<sdv>No Information</sdv>
<sdp_1>No Information</sdp_1>
<srt>-999999</srt>
<pau_1>noInformation</pau_1>
<cdt>No Information</cdt>
<ure>No Information</ure>
<ets>-999999</ets>
<exg>No Information</exg>
```

- For most of the required properties no information is available. Information about the agency that collected the data can be provided, though: agency XYZ. The data type of property sourceAgency – here identified via the 531 code “sag” – is a codelist, more specifically the codelist “OrganizationCode”. This codelist is managed outside of the application schema. The value of the sourceAgency element in this example points to an entry in a representation of the codelist.

[10] The interval is then defined via two elements, with „l“ and „u“ appended to the element name to denote the lower and upper limits of the interval.

Now further application data is encoded:

```
<bel>780</bel>
<hgt-i.l>468</hgt-i.l>
<hgt-i.u>471</hgt-i.u>
<zvh_1>1250</zvh_1>
<vdt_1>3</vdt_1>
```

- The base elevation is given via a measure (see section 7.3.1.3.3). The measure value is 780. The unit of measure is not provided in the flattened schema. The unit of measure recommended by the application schema is used instead. This applies to all measures.
- The height of the waterfall is given via an interval, more specifically two elements that define the lower limit (hgt-i.l – „l“ for *lower*) and upper limit (hgt-i.u – „u“ for *upper*) of the interval.
- The flattened schema does not support attribute value metadata. Therefore information about the source of the height value cannot be represented here.

The complete XML instance from this example is contained in an Annex (see section A.2).

8.4.4 Feature Collection

The encoding examples from the previous sections are based on an XML instance with exactly one feature. If information about exactly one feature needs to be exchanged, this is sufficient. In order to exchange multiple XML encoded features, the Web Feature Service (WFS) FeatureCollection can be used, as shown in the following examples.

GML 3.2.1 (complex form)

```
<wfs:FeatureCollection numberMatched=”unknown” numberReturned=”2” timeStamp=”2014-05-04T18:13:51”
 xmlns:gml=”http://www.opengis.net/gml/3.2”
 xmlns:hok=”http://www.dgiwg.org/xmlns/DGIF/{DPS-Identifier}/{Version-Number}”
 xmlns:wfs=”http://www.opengis.net/wfs/2.0”
 xmlns:xlink=”http://www.w3.org/1999/xlink” xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”>
 <wfs:member>
  <hok:Waterfall gml:id=”ID001”> ... </hok:Waterfall>
 </wfs:member>
 <wfs:member>
  <hok:InlandWater gml:id=“ID002“> … </hok:InlandWater>
 </wfs:member>
</wfs:FeatureCollection>
```

GML 3.2.1 (flattened form)

```
<wfs:FeatureCollection numberMatched="unknown" numberReturned="2" timeStamp="2014-05-04T18:13:51"
 xmlns:gml="http://www.opengis.net/gml/3.2"
 xmlns:hok=" http://www.dgiwg.org/xmlns/DGIF/{DPS-Identifier}/{Version-Number}/flat"
 xmlns:wfs="http://www.opengis.net/wfs/2.0"
 xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <wfs:member>
  <hok:BH180_P gml:id="ID001"> ... </hok:BH180_P>
 </wfs:member>
 <wfs:member>
  <hok:BH180_P gml:id="ID002"> ... </hok:BH180_P>
 </wfs:member>
</wfs:FeatureCollection>
```

8.4.5 Geometry

8.4.5.1 Common Rules according to DGIWG-Profiles

Generally, requirements for GML representations for level L1 (free geometry) apply for the encoding of geometry (See *DGIWG 100 – DGIWG 2D Spatial Schema Profile* for more information).

Table 7: Allowed GML geometry objects

Dimension	Simple Geometry	Aggregation
0	gml:Point	gml:MultiPoint
1	gml:Curve, gml:LineString	gml:MultiCurve
2	gml:Surface, gml:Polygon, *gml:PolyhedralSurface, gml:Tin, gml:TriangulatedSurface*	gml:MultiSurface
3	*gml:Solid*	*gml:MultiSolid*

GML geometry objects written in italics shall only be used for geometries with a three-dimensional coordinate reference system.

A segment of a gml:Curve is either a gml:LineStringSegment, a gml:ArcString, or a gml:GeodesicString.

Each patch of a gml:Surface is a gml:PolygonPatch or a *gml:Triangle* (only in 3D).

The DGIWG Profile also allows gml:MultiGeometry, but this cannot occur in DGIF exchange files.

The following elements are allowed in rings that describe the borders of a gml:Surface and gml:Polygon:

- gml:LinearRing
- gml:Ring

8.4.5.2 Additional Recommendations

GIS software usually does not support all variants of how GML geometries can be encoded. The following recommendations for the encoding of geometries in DPS are based on practical experience, and aim to support simple readability of the exchange files. The recommendations are irrelevant if the capabilities to process geometries of the target system are known.

1. The interpolation of lines should be restricted to linear interpolation, i.e. without arcs (gml:ArcString) and explicit geodesic lines (gml:GeodesicString).
2. Geometries should always belong to a feature, i.e. the GML encoding of the feature contains the full geometry. This means that the GML representation of geometries does not contain links (xlink:href attributes) to other geometries.

   Because geometries are not shared, the XML-ID-valued gml:id attributes on geometry objects do not need to be persistent; it is only required that they are unique within each GML document. In two different data exports the gml:id of the same geometry object can therefore be different – as long as the gml:id is unique within the XML document.
3. Upon multiple alternatives, the simplest, most compact geometry representation should be chosen. Based upon recommendation 2, gml:Curve, gml:Surface, and gml:Ring are not needed. Instead, gml:LineString, gml:Polygon / gml:LinearRing can be used.
4. Coordinates should always be provided via the elements gml:pos (for gml:Point) and gml:posList (for all other geometries), respectively.
5. The commonly used definitions of the EPSG registry should be used for coordinate reference systems. OGC recommends the use of http-URIs for coordinate reference systems following the schema *http://www.opengis.net/def/crs/EPSG/0/{code}* where "{code}" has to be replaced by the code of the actual coordinate reference system, for example *http://www.opengis.net/def/crs/EPSG/0/4326* for geodetic WGS 84 coordinates with the axis order lat/long.

   The attributes srsDimension, axisLabels, and uomLabels do not need to be set, because usually they are not evaluated by software.
6. If the 2D profile is sufficient, it is recommended to stick to it.

Recommendations 1 and 2 are especially consistent with the Simple-Feature standard (ISO 19125-1:2004), which typically is the basis for geometry representations in SQL databases. Requirements that go beyond this, for example in the maritime or aeronautical domains, where non-linear, geodetic interpolations are used, are often not readily supported by GIS software.

Accordingly, when following the requirements the table with allowed GML geometry objects can be reduced to:

Dimension	Simple Geometry	Aggregation
0	gml:Point	gml:MultiPoint
1	gml:LineString	gml:MultiCurve
2	gml:Polygon mit gml:LinearRing	gml:MultiSurface
3	-	-

8.4.5.3 Examples

For each of the recommended geometries an example is provided in GML 3.2.1. In the examples the namespace prefix *gml* identifies the XML namespace *http://www.opengis.net/gml/3.2*.

<u>gml:Point</u>

```
<gml:Point gml:id="ID005" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
  <gml:pos>47.501061 13.012573</gml:pos>
</gml:Point>
```

gml:LineString

```
<gml:LineString gml:id="ID006" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
  <gml:posList>53.7235 -0.8641 53.7263 -0.8617 53.7255 -0.8594</gml:posList>
</gml:LineString>
```

gml:Polygon

Note that for the boundary (encoded by gml:LinearRing or gml:Ring) of surfaces the orientation is strictly specified. The outer boundary shall be oriented counter-clockwise, while the inner boundary shall be encoded clockwise.

```
<gml:Polygon gml:id="ID007" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
  <gml:exterior>
    <gml:LinearRing gml:id="ID007_1">
      <gml:posList>53.7235 -0.8641 53.7263 -0.8617 53.7255 -0.8594 53.724 -0.8599 53.7235
        -0.8641</gml:posList>
    </gml:LinearRing>
  </gml:exterior>
  <gml:interior>
    <gml:LinearRing gml:id="ID007_3">
      <gml:posList>53.7243 -0.8628 53.7248 -0.8628 53.7248 -0.8632 53.7243 -0.8632 53.7243
        -0.8628</gml:posList>
    </gml:LinearRing>
  </gml:interior>
</gml:Polygon>
```

gml:MultiPoint

```
<gml:MultiPoint gml:id="ID005" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
  <gml:pointMember>
    <gml:Point gml:id="ID005_1"> … </gml:Point>
  </gml:pointMember>
  <gml:pointMember>
    <gml:Point gml:id="ID005_2"> … </gml:Point>
  </gml:pointMember>
</gml:MultiPoint>
```

gml:MultiCurve

```
<gml:MultiCurve gml:id="ID005" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
  <gml:curveMember>
    <gml:LineString gml:id="ID005_1"> … </gml:LineString>
  </gml:curveMember>
  <gml:curveMember>
    <gml:LineString gml:id="ID005_2"> … </gml:LineString>
  </gml:curveMember>
</gml:MultiCurve>
```

gml:MultiSurface

```
<gml:MultiSurface gml:id="ID005" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
  <gml:surfaceMember>
    <gml:Polygon gml:id="ID005_1"> … </gml:Polygon>
```

```
  </gml:surfaceMember>
  <gml:surfaceMember>
    <gml:Polygon gml:id="ID005_2"> … </gml:Polygon>
  </gml:surfaceMember>
</gml:MultiSurface>
```

8.5 Decoding Data

This section explains how data contained in GML/XML documents can be decoded.

In some cases GML supports different ways to encode information. In this section we use the recommended option that has been used in the previous section. If other variants are of interest, consult the GML standard.

The format of a given GML document can be identified via the namespaces of the encoded objects (see section 8.1). The following example shows (the framework for) an object that is encoded in GML (complex form):

```
<Waterfall gml:id="ID001" xmlns=" http://www.dgiwg.org/xmlns/DGIF/{DPS-Identifier}/{Version-Number}" …>
```

Once the format is known then in another preprocessing step the GML document can be validated against the according schema (XML Schema and maybe also Schematron schema).

Then the actual content can be read / decoded.

8.5.1 Decoding - Complex-AlphaCode

The XML document from section A.1 is used as example.

At first, the type of the encoded feature is identified, via its element name. In the complex form the element name represents the AlphaCode of the feature type. The mapping (between element name and AlphaCode) therefore is unique.

Once the feature type (here: Waterfall) has been identified, the property values can be read. To do so the list of elements on the level directly beneath the Waterfall element is examined.

```
  <classification>
    <ClassificationInfo gml:id="ID002">
      <securityClassification>unclassified</securityClassification>
    </ClassificationInfo>
  </classification>
```

- The element classification contains information about the military classification. It contains an object (identified in the XML document via the ID "ID002") which defines – via the element securityClassification – that the information (about the Waterfall) is "unclassified". The mapping between elements of the XML document and elements of the application schema is unique:
  - Elements with lowerCamelCase name represent properties of a class / an object of this class,
  - Elements with UpperCamelCase name represent a class / an object of this class.

The next element contains information about the naming of the waterfall:

```xml
<names>
 <GeoNameInfo gml:id="ID003">
  <name>
   <CharacterStringMeta>
    <valueOrReason>Röthbachfall</valueOrReason>
   </CharacterStringMeta>
  </name>
  <tacticalUsage>
   <BooleanMeta>
    <valueOrReason>false</valueOrReason>
   </BooleanMeta>
  </tacticalUsage>
 </GeoNameInfo>
</names>
```

There are no specific aspects to note. Decoding the element tells us that the (more specifically "a") name of the waterfall is "Röthbachfall" and that the feature does not have a special tactical use.

Now geometry information is decoded:

```xml
<geometry>
 <GeometryInfoMeta>
  <valueOrReason>
   <PointGeometryInfo gml:id="ID004">
    <horizCoordMetadata>
     <HorizCoordMetadata>
      <horizAccuracyCategory>
       <HorizAccuracyCategory_HorizCoordMetadataMeta>
        <valueOrReason>approximate</valueOrReason>
       </HorizAccuracyCategory_HorizCoordMetadataMeta>
      </horizAccuracyCategory>
     </HorizCoordMetadata>
    </horizCoordMetadata>
    <representationScale>
     <IntegerMeta>
      <valueOrReason nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763" xsi:nil="true"/>
     </IntegerMeta>
    </representationScale>
    <vertCoordMetadata>
     <VertCoordMetadata>
      <elevationAccuracyCategory>
       <ElevationAccuracyCategory_VertCoordMetadataMeta>
        <valueOrReason>approximate</valueOrReason>
       </ElevationAccuracyCategory_VertCoordMetadataMeta>
      </elevationAccuracyCategory>
     </VertCoordMetadata>
    </vertCoordMetadata>
    <angleOfOrientation>
     <MeasureMeta>
      <valueOrReason nilReason="https://www.dgiwg.org/FAD/fdd/view?i=202222" uom="nil"
       xsi:nil="true"/>
     </MeasureMeta>
    </angleOfOrientation>
    <area>
     <MeasureMeta>
      <valueOrReason nilReason="https://www.dgiwg.org/FAD/fdd/view?i=202222" uom="nil"
       xsi:nil="true"/>
     </MeasureMeta>
```

```xml
        </area>
        <geometry>
          <gml:Point gml:id="ID005" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
            <gml:pos>47.501061 13.012573</gml:pos>
          </gml:Point>
        </geometry>
        <length>
          <MeasureIntervalMeta>
            <valueOrReason nilReason="https://www.dgiwg.org/FAD/fdd/view?i=202222" xsi:nil="true"
            />
          </MeasureIntervalMeta>
        </length>
        <width>
          <MeasureIntervalMeta>
            <valueOrReason nilReason="https://www.dgiwg.org/FAD/fdd/view?i=202222" xsi:nil="true"
            />
          </MeasureIntervalMeta>
        </width>
      </PointGeometryInfo>
    </valueOrReason>
  </GeometryInfoMeta>
</geometry>
```

- The xsi:nil and nilReason attribute values in the respresentationScale/IntegerMeta/valueOrReason element define that no actual value is available for the property representationScale. Instead, the nilReason attribute value points to an ONINA code (see section 7.3.4). In this way, ONINA code values for properties are generally provided.
- The geometry is defined via a GML (3.2.1) Point. It shall be decoded following the GML 3.2.1 standard (see section 8.4.5):
  - The attribute srsName with value http://www.opengis.net/def/crs/EPSG/0/4326 defines that the coordinate reference system is WGS84 2D (that has the EPSG code 4326).
  - Based upon the definition of the coordinate reference system – especially its coordinate system (which defines the order of coordinate axes and their semantics) – the coordinates contained in the gml:pos element are read: the first number provides the geodetic latitude (47.501061 decimal degrees) while the second number provides the geodetic longitude (13.012573 decimal degrees).
- Decoding the other elements shows that the positional accuracy is approximate and that the properties angleOfOrientation, area, length and width are not applicable.

There is nothing specific to add regarding the decoding of identification information and additional information:

```xml
<identification>
  <IdentificationInfo gml:id="identInfo_01">
    <genericEntityIdentifier>roethbachfall</genericEntityIdentifier>
    <numericFeatureIdentifier>74831414</numericFeatureIdentifier>
  </IdentificationInfo>
</identification>
<informationReference>
  <CharacterStringMeta>
    <valueOrReason>http://en.wikipedia.org/wiki/Röthbach_Waterfall</valueOrReason>
  </CharacterStringMeta>
</informationReference>
```

```
<informationReference>
  <CharacterStringMeta>
    <valueOrReason>http://www.worldwaterfalldatabase.com/waterfall/Rothbachfall-42/</valueOrReason>
  </CharacterStringMeta>
</informationReference>
```

- The identifier for the waterfall is: roethbachfall
- The numeric identifier is: 74831414
- Additional information about the waterfall is available at the following locations:
  - http://en.wikipedia.org/wiki/Röthbach_Waterfall
  - http://www.worldwaterfalldatabase.com/waterfall/Rothbachfall-42/

Evaluation of encoded reliability information shows that the waterfall definitely exists:

```
<reliability>
  <ReliabilityInfo gml:id=”reliabInfo_01”>
    <existenceCertaintyCat>definite</existenceCertaintyCat>
    <existenceAssessment xsi:nil=”true” nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763”/>
  </ReliabilityInfo>
</reliability>
```

Objects like the ReliabilityInfo object shown in this example can also be referenced in XML documents via xlink:href, as follows:

```
<reliability xlink:href=”http://my.service.org/reliabInfo_01”/>
```

In this case the URI (which often is a URL) has to be resolved and the referenced resource decoded.

Data about restrictions is encoded in the XML document, but only because this is required by the according GML schema. The encoding uses ONINA code values to indicate that no information is available for any of the properties:

```
<restriction>
  <RestrictionInfo gml:id=”restrInfo_01”>
    <commercialCopyrightNotice xsi:nil=”true” nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763”/>
    <commercialDistribRestrict xsi:nil=”true” nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763”/>
    <ownerAuthority xsi:nil=”true” nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763”/>
    <releasability xsi:nil=”true” nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763”/>
    <releasabilityRestriction xsi:nil=”true” nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763”/>
  </RestrictionInfo>
</restriction>
```

This is almost the same for the source of information about the waterfall:

```
<source>
  <SourceInfo gml:id=”sourceInfo_01”>
    <sourceAgency xlink:href=”http://register.org/XYZ” xlink:title=”XYZ”/>
    <sourceDateTime xsi:nil=”true” nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763”/>
    <sourceDescription xsi:nil=”true” nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763”/>
    <sourceType xsi:nil=”true” nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763”/>
    <productionAuthority xsi:nil=”true” nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763”/>
    <creationDateTime xsi:nil=”true” nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763”/>
    <updateReason xsi:nil=”true” nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763”/>
    <extractionSpec xsi:nil=”true” nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763”/>
    <dataCaptureSpec xsi:nil=”true” nilReason=”https://www.dgiwg.org/FAD/fdd/view?i=203763”/>
```

```xml
  </SourceInfo>
</source>
```

- Only the agency that collected the waterfall data is known. The element sourceAgency points to a specific value of a codelist, via the xlink:href attribute value. The according information can be retrieved. The value can – as shown in the example – also be provided via the xlink:title attribute. So the data originates from agency XYZ.

Now further application data is decoded:

```xml
<baseElevation>
  <MeasureMeta>
    <valueOrReason uom="m">780</valueOrReason>
  </MeasureMeta>
</baseElevation>
```

- The base elevation is given as a measure (see the “MeasureMeta“ element). The value of the property is 780 meters. The unit of measure is defined via the uom attribute, which contains a UCUM code[11].

The following information is available about the height of the waterfall:

```xml
<heightAboveSurfaceLevel>
  <MeasureIntervalMeta>
    <attributeMetadata>
      <FeatureAttMetadata gml:id="ID006">
        <source>
          <SourceInfo gml:id="ID007">
            <sourceAgency xlink:href="http://register.org/ABC" xlink:title="ABC"/>
            <sourceDateTime>2011-05-15</sourceDateTime>
            <sourceDescription nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763" xsi:nil="true"/>
            <sourceType>mapChartOrGeodeticData</sourceType>
            <productionAuthority nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763" xsi:nil="true"/>
            <creationDateTime>2013-10-15T09:11:00Z</creationDateTime>
            <updateReason nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763" xsi:nil="true"/>
            <extractionSpec nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763" xsi:nil="true"/>
            <dataCaptureSpec nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763" xsi:nil="true"/>
          </SourceInfo>
        </source>
      </FeatureAttMetadata>
    </attributeMetadata>
    <valueOrReason>
      <MeasureIntervalUnion>
        <intervalValue>
          <MeasureInterval>
            <lowerValue uom="m">468</lowerValue>
            <upperValue uom="m">471</upperValue>
          </MeasureInterval>
        </intervalValue>
      </MeasureIntervalUnion>
    </valueOrReason>
  </MeasureIntervalMeta>
</heightAboveSurfaceLevel>
```

[11] see http://unitsofmeasure.org

- Again a measure is provided, this time as an interval. The waterfall thus has a height (above surface level) between 468 and 471 meters.
- In addition metadata for the (property) value itself is available: the source of the value is agency ABC; the value has been collected on May 15th, 2011.

There is nothing to add about the rest of the data contained in the example:

```
<highestElevation>
  <MeasureMeta>
    <valueOrReason uom="m">1250</valueOrReason>
  </MeasureMeta>
</highestElevation>
<verticalDatum>
  <VerticalDatum_WaterfallMeta>
    <valueOrReason>meanSeaLevel</valueOrReason>
  </VerticalDatum_WaterfallMeta>
</verticalDatum>
```

- The vertical datum is mean sea level.
- The height above mean sea level is 1250 meters.

8.5.2 Decoding - Flat-531 Code

The XML document from section A.2 is used as example.

At first, the type of the encoded feature is identified, via its element name. In the flattened form the element name represents the 531 code of the feature type, where the suffix "_P", "_C" or "_S" is not important for the moment (it only identifies the type of the geometry used in the encoded data). The mapping (between element name and 531 code) therefore is unique.

In the example, the name of the element containing feature information is BH180_P. The 531 code BH180 belongs to the feature type Waterfall.

Once the feature type has been identified, the property values can be read. To do so the list of elements on the level directly beneath the BH180 element is examined

```
<sec>34</sec>
```

- All waterfall properties are identified by their 531 code. Sometimes the flattening process results in element names that represent a concatenation of 531 codes. This will become apparent in following examples.

  The feature catalogue can be used to easily identify the 531 code that belongs to a feature property. The documentation of a GML schema element (in flattened form) also points out the AlphaCode of the element whose name is given via a 531 code.
- The element sec contains information about the military classification, more specifically the security classification (here: „unclassified“). The datatype of this property is an enumeration. In the flattened form, enumeration values are given with their 531 code (here: "34"). The feature catalogue can be used to identify the code.

The next elements contain information about the naming of the waterfall:

```
<nam_1>Röthbachfall</nam_1>
<tru>1000</tru>
```

- The name of the waterfall is given by the nam_1 element. The suffix "_1" indicates that the element provides one out of a list of potential property values. Multiple names would be represented by elements nam_1, nam_2, etc, and decoded accordingly.
- The tactical use is provided via a boolean datatype. In the flattened form the boolean value is given via a 531 code – just like for enumerations (1000 for "false" and 1001 for "true").

Now geometry information is decoded:

```
<xg1_1-zi007.acc>2</xg1_1-zi007.acc>
<xg1_1-zi007.ela>2</xg1_1-zi007.ela>
<xg1_1-zi007.aoo>-32765</xg1_1-zi007.aoo>
<xg1_1-zi007.ara>-32765</xg1_1-zi007.ara>
<xg1_1-zi007.xg1>
  <gml:Point gml:id="ID005" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
   <gml:pos>47.501061 13.012573</gml:pos>
  </gml:Point>
</xg1_1-zi007.xg1>
<xg1_1-zi007.len-s>-32765</xg1_1-zi007.len-s>
<xg1_1-zi007.wid-s>-32765</xg1_1-zi007.wid-s>
```

- Apparently the elements are the result of flattening a complex data structure:
  - "xg1" represents the property "geometry" of the waterfall.
  - "_1" indicates that multiple values can be given for the geometry property.
  - "-zi007" provides the 531 code of the actual GeometryInfo object that contains the geometry information. In this example a PointGeometryInfo object is used.
  - Further parts of the element name, separated by ".", are PointGeometryInfo class properties identified by their 531 codes.
  - The feature catalogue can be used to identify the 531 code.
  - Many elements (e.g. xg1_1-zi007.aoo) contain the value -32765. It indicates that in this case a value does not apply for the according property. Section 7.3.4 defines how ONINA code values shall be used / mapped in GML schema (flattened form).
  - The example contains another special case: if an ONINA shall be given for a property of type MeasureInterval (for example for the geometry length) then it is given via the "single" option of the MeasureInterval. This option is indicated via the "-s" in the xg1_1-zi007.len-s element name. If an interval shall be provided for a MeasureInterval then "-i" is used instead[12]. This is shown in one of the following examples (where the height of the waterfall is decoded).
- The elements xg1_1-zi007.acc and xg1_1-zi007.ela indicate that the accuracy of the latitude and longitude values is approximate. The datatypes of the according properties are enumerations, whose values are given via their 531 codes in the flattened form (in this case "2" represents "approximate").
- The geometry is defined via a GML Point. It shall be decoded following the GML standard (see section 8.4.5):

[12] The interval is then given via two elements, with suffix „l" and „u" to define the lower and upper limits, respectively.

  - The attribute srsName with value http://www.opengis.net/def/crs/EPSG/0/4326 defines that the coordinate reference system is WGS84 2D (that has the EPSG code 4326).
  - Based upon the definition of the coordinate reference system – especially its coordinate system (which defines the order of coordinate axes and their semantics) – the coordinates contained in the gml:pos element are read: the first number provides the geodetic latitude (47.501061 decimal degrees) while the second number provides the geodetic longitude (13.012573 decimal degrees).

There is nothing specific to add regarding the decoding of identification information and additional information:

```
<bgi>roethbachfall</bgi>
<uid>74831414</uid>
<hyl_1>http://en.wikipedia.org/wiki/Röthbach_Waterfall</hyl_1>
<hyl_2>http://www.worldwaterfalldatabase.com/waterfall/Rothbachfall-42/</hyl_2>
```

- The identifier for the waterfall is: roethbachfall
- The numeric identifier is: 74831414
- Additional information about the waterfall is available at the following locations:
  - http://en.wikipedia.org/wiki/Röthbach_Waterfall
  - http://www.worldwaterfalldatabase.com/waterfall/Rothbachfall-42/

Evaluation of encoded reliability information shows that the waterfall definitely exists:

```
<coe>1</coe>
```

Data about restrictions is encoded in the XML document, but only because this is required by the according GML schema. The encoding uses ONINA code values to indicate that no information is available for any of the properties:

```
<ccn>No Information</ccn>
<cdr>No Information</cdr>
<ona_1>No Information</ona_1>
<qle>No Information</qle>
```

This is almost the same for the source of information about the waterfall:

```
<sag>XYZ</sag>
<sdv>No Information</sdv>
<sdp_1>No Information</sdp_1>
<srt>-999999</srt>
<pau_1>noInformation</pau_1>
<cdt>No Information</cdt>
<ure>No Information</ure>
<ets>-999999</ets>
<exg>No Information</exg>
```

- Only the agency that collected the waterfall data is known. The element sag contains a specific value of a codelist (which, according to the application schema, contains codes for organisations). So the data originates from agency XYZ.

Now further application data is decoded:

```
<bel>780</bel>
```

- According to the application schema the base elevation is a measure. Its value is 780 meters. In the flattened form the unit of measure is not encoded. Instead, the unit of measure that the application schema recommends for a property is used.

The following information is available about the height of the waterfall:

```
<hgt-i.l>468</hgt-i.l>
<hgt-i.u>471</hgt-i.u>
```

- Again a measure is provided, this time as an interval. The waterfall thus has a height (above surface level) between 468 and 471 meters (with the unit of measure being defined by the appliation schema, as mentioned before).

There is nothing to add about the rest of the data contained in the example:

```
<zvh_1>1250</zvh_1>
<vdt_1>3</vdt_1>
```

- The vertical datum (defined via the element vdt_1) is mean sea level.
- The height above mean sea level is 1250 meters.

Annex A - XML Examples

A.1 XML instance of a waterfall, encoded in GML 3.2.1 (complex form)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Waterfall gml:id="ID001" xmlns=" http://www.dgiwg.org/xmlns/DGIF/{DPS-Identifier}/{Version-Number}"
 xmlns:gco="http://www.isotc211.org/2005/gco" xmlns:gmd="http://www.isotc211.org/2005/gmd"
 xmlns:gml="http://www.opengis.net/gml/3.2"
 xmlns:gmldgiwgsp="http://www.dgiwg.org/xmlns/GML/3.2/profiles/spatial/1.0/"
 xmlns:gmlexr="http://www.opengis.net/gml/3.3/exr" xmlns:gsr="http://www.isotc211.org/2005/gsr"
 xmlns:gss="http://www.isotc211.org/2005/gss" xmlns:gts="http://www.isotc211.org/2005/gts"
 xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <classification>
  <ClassificationInfo gml:id="ID002">
   <securityClassification>unclassified</securityClassification>
  </ClassificationInfo>
 </classification>
 <names>
  <GeoNameInfo gml:id="ID003">
   <name>
    <CharacterStringMeta>
     <valueOrReason>Röthbachfall</valueOrReason>
    </CharacterStringMeta>
   </name>
   <tacticalUsage>
    <BooleanMeta>
     <valueOrReason>false</valueOrReason>
    </BooleanMeta>
   </tacticalUsage>
  </GeoNameInfo>
 </names>
 <geometry>
  <GeometryInfoMeta>
   <valueOrReason>
    <PointGeometryInfo gml:id="ID004">
     <horizCoordMetadata>
      <HorizCoordMetadata>
       <horizAccuracyCategory>
        <HorizAccuracyCategory_HorizCoordMetadataMeta>
         <valueOrReason>approximate</valueOrReason>
        </HorizAccuracyCategory_HorizCoordMetadataMeta>
       </horizAccuracyCategory>
      </HorizCoordMetadata>
     </horizCoordMetadata>
     <representationScale>
      <IntegerMeta>
       <!-- No information -->
       <valueOrReason nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763" xsi:nil="true"
       />
      </IntegerMeta>
     </representationScale>
     <vertCoordMetadata>
      <VertCoordMetadata>
       <elevationAccuracyCategory>
        <ElevationAccuracyCategory_VertCoordMetadataMeta>
```

```xml
          <valueOrReason>approximate</valueOrReason>
         </ElevationAccuracyCategory_VertCoordMetadataMeta>
        </elevationAccuracyCategory>
       </VertCoordMetadata>
      </vertCoordMetadata>
      <angleOfOrientation>
       <MeasureMeta>
        <!-- Not applicable -->
        <valueOrReason nilReason="https://www.dgiwg.org/FAD/fdd/view?i=202222" uom="nil"
         xsi:nil="true"/>
       </MeasureMeta>
      </angleOfOrientation>
      <area>
       <MeasureMeta>
        <!-- Not applicable -->
        <valueOrReason nilReason="https://www.dgiwg.org/FAD/fdd/view?i=202222" uom="nil"
         xsi:nil="true"/>
       </MeasureMeta>
      </area>
      <geometry>
       <gml:Point gml:id="ID005" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
        <gml:pos>47.501061 13.012573</gml:pos>
       </gml:Point>
      </geometry>
      <length>
       <MeasureIntervalMeta>
        <!-- Not applicable -->
        <valueOrReason nilReason="https://www.dgiwg.org/FAD/fdd/view?i=202222" xsi:nil="true"
        />
       </MeasureIntervalMeta>
      </length>
      <width>
       <MeasureIntervalMeta>
        <!-- Not applicable -->
        <valueOrReason nilReason="https://www.dgiwg.org/FAD/fdd/view?i=202222" xsi:nil="true"
        />
       </MeasureIntervalMeta>
      </width>
     </PointGeometryInfo>
    </valueOrReason>
   </GeometryInfoMeta>
  </geometry>
  <identification>
   <IdentificationInfo gml:id="identInfo_01">
    <genericEntityIdentifier>roethbachfall</genericEntityIdentifier>
    <numericFeatureIdentifier>74831414</numericFeatureIdentifier>
   </IdentificationInfo>
  </identification>
  <informationReference>
   <CharacterStringMeta>
    <valueOrReason>http://en.wikipedia.org/wiki/Röthbach_Waterfall</valueOrReason>
   </CharacterStringMeta>
  </informationReference>
  <informationReference>
   <CharacterStringMeta>
    <valueOrReason>http://www.worldwaterfalldatabase.com/waterfall/Rothbachfall-42/</valueOrReason>
   </CharacterStringMeta>
  </informationReference>
  <reliability>
   <ReliabilityInfo gml:id="reliabInfo_01">
```

```xml
    <existenceCertaintyCat>definite</existenceCertaintyCat>
    <existenceAssessment xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
  </ReliabilityInfo>
</reliability>
<restriction>
  <RestrictionInfo gml:id="restrInfo_01">
    <commercialCopyrightNotice xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <commercialDistribRestrict xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <ownerAuthority xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <releasability xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <releasabilityRestriction xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
  </RestrictionInfo>
</restriction>
<source>
  <SourceInfo gml:id="sourceInfo_01">
    <sourceAgency xlink:href="http://register.org/XYZ" xlink:title="XYZ"/>
    <sourceDateTime xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <sourceDescription xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <sourceType xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <productionAuthority xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <creationDateTime xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <updateReason xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <extractionSpec xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
    <dataCaptureSpec xsi:nil="true" nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763"/>
  </SourceInfo>
</source>
<sounding xlink:href="http://my.register.org/sounding/1040"/>
<baseElevation>
  <MeasureMeta>
    <valueOrReason uom="m">780</valueOrReason>
  </MeasureMeta>
</baseElevation>
<heightAboveSurfaceLevel>
  <MeasureIntervalMeta>
    <attributeMetadata>
      <FeatureAttMetadata gml:id="ID006">
        <source>
          <SourceInfo gml:id="ID007">
            <sourceAgency xlink:href="http://register.org/ABC" xlink:title="ABC"/>
            <sourceDateTime>2011-05-15</sourceDateTime>
            <sourceDescription nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763" xsi:nil="true"/>
            <sourceType>mapChartOrGeodeticData</sourceType>
            <productionAuthority nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763" xsi:nil="true"/>
            <creationDateTime>2013-10-15T09:11:00Z</creationDateTime>
            <updateReason nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763" xsi:nil="true"/>
            <extractionSpec nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763" xsi:nil="true"/>
            <dataCaptureSpec nilReason="https://www.dgiwg.org/FAD/fdd/view?i=203763" xsi:nil="true"/>
          </SourceInfo>
        </source>
      </FeatureAttMetadata>
    </attributeMetadata>
    <valueOrReason>
      <MeasureIntervalUnion>
        <intervalValue>
          <MeasureInterval>
            <lowerValue uom="m">468</lowerValue>
            <upperValue uom="m">471</upperValue>
          </MeasureInterval>
        </intervalValue>
      </MeasureIntervalUnion>
```

```xml
      </valueOrReason>
    </MeasureIntervalMeta>
  </heightAboveSurfaceLevel>
  <highestElevation>
    <MeasureMeta>
      <valueOrReason uom="m">1250</valueOrReason>
    </MeasureMeta>
  </highestElevation>
  <verticalDatum>
    <VerticalDatum_WaterfallMeta>
      <valueOrReason>meanSeaLevel</valueOrReason>
    </VerticalDatum_WaterfallMeta>
  </verticalDatum>
</Waterfall>
```

A.2 XML instance of a waterfall, encoded in GML 3.2.1 (flattened form)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<BH180_P gml:id="ID001" xmlns=" http://www.dgiwg.org/xmlns/DGIF/{DPS-Identifier}/{Version-Number}/flat"
   xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <sec>34</sec>
   <nam_1>Röthbachfall</nam_1>
   <tru>1000</tru>
   <xg1_1-zi007.acc>2</xg1_1-zi007.acc>
   <xg1_1-zi007.ela>2</xg1_1-zi007.ela>
   <xg1_1-zi007.aoo>-32765</xg1_1-zi007.aoo>
   <xg1_1-zi007.ara>-32765</xg1_1-zi007.ara>
   <xg1_1-zi007.xg1>
     <gml:Point gml:id="ID005" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
      <gml:pos>47.501061 13.012573</gml:pos>
     </gml:Point>
   </xg1_1-zi007.xg1>
   <xg1_1-zi007.len-s>-32765</xg1_1-zi007.len-s>
   <xg1_1-zi007.wid-s>-32765</xg1_1-zi007.wid-s>
   <bgi>roethbachfall</bgi>
   <uid>74831414</uid>
   <hyl_1>http://en.wikipedia.org/wiki/Röthbach_Waterfall</hyl_1>
   <hyl_2>http://www.worldwaterfalldatabase.com/waterfall/Rothbachfall-42/</hyl_2>
   <coe>1</coe>
   <ccn>No Information</ccn>
   <cdr>No Information</cdr>
   <ona_1>No Information</ona_1>
   <qle>No Information</qle>
   <sag>XYZ</sag>
   <sdv>No Information</sdv>
   <sdp_1>No Information</sdp_1>
   <srt>-999999</srt>
   <pau_1>noInformation</pau_1>
   <cdt>No Information</cdt>
   <ure>No Information</ure>
   <ets>-999999</ets>
   <exg>No Information</exg>
   <bel>780</bel>
   <hgt-i.l>468</hgt-i.l>
   <hgt-i.u>471</hgt-i.u>
   <zvh_1>1250</zvh_1>
   <vdt_1>3</vdt_1>
</BH180_P>
```