



OGC TRAINING DATA MARKUP LANGUAGE FOR ARTIFICIAL INTELLIGENCE (TRAININGDML-AI) PART 3: XML ENCODING STANDARD

STANDARD
Encoding

APPROVED

Version: 1.0.1

Submission Date: 2026-03-01

Approval Date: 2026-04-28

Publication Date: 2026-08-20

Editor: Peng Yue, Ruixiang Liu, Kaixuan Wang

Notice: This document is an OGC Member approved international standard. This document is available on a royalty free, non-discriminatory basis. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

License Agreement

Use of this document is subject to the license agreement at <https://www.ogc.org/license>

Suggested additions, changes and comments on this document are welcome and encouraged. Such suggestions may be submitted using the online change request form on OGC web site: <http://ogc.standardstracker.org/>

Copyright notice

Copyright © 2026 Open Geospatial Consortium

To obtain additional rights of use, visit <https://www.ogc.org/legal>

Note

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

CONTENTS

I. ABSTRACT	v
II. KEYWORDS	v
III. PREFACE	vi
IV. SECURITY CONSIDERATIONS	vii
V. SUBMITTING ORGANIZATIONS	viii
VI. SUBMITTERS	viii
VII. ACKNOWLEDGEMENTS	ix
1. SCOPE	2
2. CONFORMANCE	4
3. NORMATIVE REFERENCES	6
4. TERMS AND DEFINITIONS	8
5. CONVENTIONS	15
5.1. Identifiers	15
5.2. Abbreviated Terms	15
6. OVERVIEW	18
6.1. Extensible Markup Language	18
7. REQUIREMENTS FOR TRAININGDML-AI XML ENCODING	20
7.1. Requirements Class: Base	20
7.2. Requirements Class: AI_TrainingDataset	30
7.3. Requirements Class: AI_TrainingData	39
7.4. Requirements Class: AI_Task	43
7.5. Requirements Class: AI_Label	44
7.6. Requirements Class: AI_Labeling	48
7.7. Requirements Class: AI_DataQuality	51
7.8. Requirements Class: AI_TDChangeset	52
ANNEX A ABSTRACT TEST SUITE	56
A.1. Introduction	56

A.2. Conformance Class: Base	56
A.3. Conformance Class: AI_TrainingDataset	61
A.4. Conformance Class: AI_TrainingData	63
A.5. Conformance Class: AI_Task	65
A.6. Conformance Class: AI_Label	66
A.7. Conformance Class: AI_Labeling	68
A.8. Conformance Class: AI_DataQuality	70
A.9. Conformance Class: AI_TDChangeset	71
 ANNEX B EXAMPLE	 74
B.1. TrainingDataset Encoding Examples	74
B.2. DataQuality Encoding Example	76
B.3. TDChangeset Encoding Example	77
B.4. Non-EO Imagery TrainingDataset Encoding Examples	77
 ANNEX C REVISION HISTORY	 80
 BIBLIOGRAPHY	 82



ABSTRACT

The OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 3: XML Encoding Standard defines requirements for encoding AI training datasets as Extensible Markup Language (XML). XML is a widely utilized format for encoding data in various applications. It employs a markup structure where data is enclosed in tags represented by opening and closing elements. These elements can have attributes and may contain nested elements, allowing for a hierarchical representation of structured information. TrainingDML-AI Part 3 is based on the OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard.



KEYWORDS

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, artificial intelligence, machine learning, deep learning, earth observation, remote sensing, training data, training sample, encoding, XML



PREFACE

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.



SECURITY CONSIDERATIONS

No security considerations have been made for this Standard.



SUBMITTING ORGANIZATIONS

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

- Wuhan University
- Luoyao Information Technology Co., Ltd
- Pixalytics Ltd
- WiSC Enterprises
- George Mason University
- Laboratoire d'Informatique de Grenoble
- Wuhan University of Technology
- Hubei University
- Chongqing Changan Zhitu Technology Co., Ltd



SUBMITTERS

All questions regarding this submission should be directed to the editors or the submitters:

NAME	AFFILIATION
Peng Yue	Wuhan University
Ruixiang Liu	Wuhan University
Kaixuan Wang	Wuhan University
Haoru Wu	Wuhan University
Boyi Shangguan	Luoyao Information Technology Co., Ltd
Samantha Lavender	Pixalytics Ltd
Jim Antonisse	WiSC Enterprises

NAME	AFFILIATION
Liping Di	George Mason University
Eugene Yu	George Mason University
Danielle Ziébelin	Laboratoire d'Informatique de Grenoble
Liangcun Jiang	Wuhan University of Technology
Lei Hu	Hubei University
Mingda Zhang	Hubei University
Kai Yan	Chongqing Changan Zhitu Technology Co., Ltd

VII

ACKNOWLEDGEMENTS

Thanks to the members of the TrainingDML-AI Standards Working Group of the OGC as well as all contributors of change requests and comments. In particular: Scott Simmons, Carl Reed, Sam Meek, Yongyi Mi, Chenxiao Zhang, Zhipeng Cao, Shuaiqi Liu, Ming Zhao, Hanwen Xu, Haipeng Deng, Baoxin Teng.



1

SCOPE

This OGC TrainingDML — AI Part 3: XML Encoding Standard defines an XML encoding for the exchange of training datasets. The TrainingDML — AI Part 3 Standard provides an XML-based encoding for the exchange of information describing training datasets, both within and between different organizations.

The document model is derived from the conceptual models defined in the OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard.



2

CONFORMANCE

This Standard defines an XML encoding for AI training datasets. The standardization target for this Standard is:

- TrainingDML-AI XML Encoding Schema

Conformance with this Standard shall be checked using all the relevant tests specified in Annex A. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing website.

All requirements-classes and conformance-classes described in this document are owned by the standard identified.

This standard identifies eleven (11) conformance classes. Each conformance class is defined by one requirements class. The tests in Annex A are organized by requirements class. So an implementation of each conformance class must pass all tests specified in Annex A for the respective requirements class.

Of these eleven conformance classes, only the `AI_TrainingDataset` conformance class is mandatory. All other conformance classes are optional. In the case where a conformance class has a dependency on another conformance class, that conformance class should also be implemented.

If `AI_TrainingDataset` conformance class is implemented, all other conformance classes will eventually be implemented based on the dependencies between the conformance classes. Therefore, all conformance classes can also be considered mandatory.



3

NORMATIVE REFERENCES

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

OGC: OGC 07-036, OpenGIS Geography Markup Language (GML) Encoding Standard, 2007

W3C Recommendation: W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures, 2012

W3C Recommendation: W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, 2012

IETF: RFC 2046, Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, 1996

ISO 19101-1:2014 Geographic information — Reference model — Part 1: Fundamentals

ISO 19103:2024 Geographic information — Conceptual schema language

ISO 19115-1:2014 Geographic information — Metadata — Part 1: Fundamentals

ISO 19156:2023 Geographic information — Observations, measurements and samples

ISO 19157-1 Geographic information — Data quality — Part 1: General requirements



4

TERMS AND DEFINITIONS

TERMS AND DEFINITIONS

This document uses the terms defined in OGC Policy Directive 49, which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this document and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

This document also uses terms defined in the OGC Standard for Modular specifications (OGC 08-131r3), also known as the ‘ModSpec’. The definitions of terms such as standard, specification, requirement, and conformance test are provided in the ModSpec.

For the purposes of this document, the following additional terms and definitions apply.

This document used the terms defined in OGC Policy Directive 49, which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this Standard and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

For the purposes of this document, the following additional terms and definitions apply.

4.1. Artificial Intelligence (AI)

refers to a set of methods and technologies that can empower machines or software to learn and perform tasks like humans.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.2. Machine Learning (ML)

is an important branch of artificial intelligence that gives computers the ability to improve their performance without explicitly being programmed to do so. ML processes create models from training data by using a set of learning algorithms, and then can use these models to make predictions. Depending on whether the training data include labels, the learning algorithms can be divided into supervised and unsupervised learning.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.3. Deep Learning (DL)

is a subset of machine learning, which is essentially a neural network with three or more layers. The number of layers is referred to as depth. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

SOURCE: <https://www.ibm.com/topics/deep-learning>

4.4. Dataset

identifiable collection of data

[SOURCE: ISO 19115-1:2014, 4.3]

Note 1 to entry: A dataset can be a smaller grouping of data which, though limited by some constraint such as spatial extent or feature type, is located physically within a larger dataset. Theoretically, a dataset can be as small as a single feature or feature attribute contained within a larger dataset. A hardcopy map or chart can be considered a dataset.

4.5. Training Dataset

a collection of samples, often labelled with known terms or expected values for supervised learning. A training dataset can be divided into training, validation, and test sets. Training samples are different from samples in OGC Observations & Measurements (O&M). They are often collected in purposive ways that deviate from purely probability sampling, with known or expected results labeled as values of a dependent variable for generating a trained predictive model.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

Note 1 to entry: A training sample label is different from those on a geographical map, which are known as map labels or annotations.

4.6. Label

refers to known or expected results annotated as values of a dependent variable in training samples. A training sample label is different from those on a geographical map, which are known as map labels or annotations.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.7. Class

<classification> result of a classification process as part of a classification system which subdivides concepts within a given topic area.

[SOURCE: ISO 19144-2:2023, 3.1.6]

4.8. Task

the specific goal that an AI application want to achieve.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.9. Provenance

information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness. In this standard provenance is a record of how training data were prepared.

SOURCE: W3C (<https://www.w3.org/TR/prov-overview/>)

4.10. Quality

degree to which a set of inherent characteristics of an object fulfils requirements [ISO 9000:2015, 3.6.2, modified—Notes 1 and 2 to entry have been deleted]. Quality of training data (such as data imbalance and mislabeling) can impact the performance of AI/ML models.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.11. Earth Observation

data and information collected about our planet, whether atmospheric, oceanic or terrestrial. This includes space-based or remotely-sensed data, as well as ground-based or in situ data.

SOURCE: GEO (https://earthobservations.org/geo_wwd.php)

4.12. Scene Classification

task of identifying scene categories of images, on the basis of a training set of images whose scene categories are known.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.13. Object Detection

task of recognizing objects such as cars from images. The objects are often localized using bounding boxes.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.14. Semantic Segmentation

task of assigning class labels to pixels of images or points of point clouds.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.15. Change Detection

recognition of changes between images acquired at different times.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.16. 3D Model Reconstruction

task that builds 3D objects and scenes from multi-view images.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.17. Generative Model

is one of the methods of large model training, which improve model performance through unsupervised pre-training. In the fine-tuning phase, labeled data plays a critical role in optimizing the model for specific vertical domains or tasks. By incorporating labeled data, the model can learn to accurately identify and extract relevant features, leading to better performance on specific downstream tasks. Overall, the combination of generative models and fine-tuning with labeled data can significantly improve the performance of large models in specialized domains or tasks.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.18. Extensible Markup Language (XML)

is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.

SOURCE: <https://www.w3.org/XML/>

4.19. XML Schema

is a means for defining the structure, content and semantics of XML documents.

SOURCE: <https://www.w3.org/XML/Schema>

4.20. Training Dataset Publisher

refers to the entity or individual responsible for creating and releasing the XML-based serialization syntax for geospatial training datasets, as defined in the TrainingDML-AI Part 3: XML Encoding Standard.



5

CONVENTIONS

This section provides details and examples for any conventions used in the document.

5.1. Identifiers

The normative provisions in this Standard are denoted by the URI:

<http://www.opengis.net/spec/TrainingDML-AI-3/1.0>

All requirements and conformance tests that appear in this document are denoted by partial URIs which are relative to this base.

5.2. Abbreviated Terms

In this document, the following abbreviations and acronyms are used or introduced:

AI	Artificial Intelligence
DL	Deep Learning
EO	Earth Observation
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
ML	Machine Learning
OGC	Open Geospatial Consortium
RS	Remote Sensing
TD	Training Data
UML	Unified Modelling Language
URL	Uniform Resource Locator

UTC	Coordinated Universal Time
W3C	World Wide Web Consortium
XML	Extensible Markup Language



6

OVERVIEW

The TrainingDML — AI Part 3: XML Encoding Standard defines an XML-based serialization syntax for geospatial training datasets. While other serialization formats are possible, such alternatives are not discussed in this Standard.

XML employs a formal class model with elements and attributes, allowing for a more structured representation of data. XML documents are hierarchical in nature, composed of nested elements that can contain both data and metadata.

A training dataset document conforming to this Standard is an XML document whose root element is an “AI_TrainingDataset” element.

6.1. Extensible Markup Language

Extensible Markup Language (XML) is a versatile and widely used markup language designed for encoding documents in a format that is both human-readable and machine-readable. It provides a set of rules for defining custom markup languages and is often employed for representing structured data in a hierarchical and platform-independent manner. XML documents consist of elements, attributes, and text content organized within a tree-like structure. Elements are enclosed within tags, and attributes provide additional information about the elements.

XML has widespread applicability and is utilized in various domains, including web development, data interchange, configuration files, and more. Its flexibility allows users to define their own document structures, making it suitable for a broad range of applications. XML is supported by many programming languages and technologies, contributing to its interoperability and adoption.

When serialized, absent properties in XML are typically represented by leaving the corresponding elements empty, without any content. Alternatively, certain XML-based formats may use attributes to signify the absence of a property. These representations convey semantic equivalence. An empty or missing element signifies that no value has been assigned, distinct from the interpretation that the provided value is empty or nil.



7

REQUIREMENTS FOR TRAININGDML-AI XML ENCODING

REQUIREMENTS FOR TRAININGDML-AI XML ENCODING

7.1. Requirements Class: Base

7.1.1. Requirements Class: XML Base Type

The XML Base Type requirements class defines the base requirements for XML encodings, which includes definitions of common types used in the TrainingDML-AI XML encoding.

REQUIREMENTS CLASS	
/req/base/xmlbasetype	
Dependency	XML
Requirement 1	/req/base/xmlbasetype/xml
Requirement 2	/req/base/xmlbasetype/datetime
Requirement 3	/req/base/xmlbasetype/namedvalue
Requirement 4	/req/base/xmlbasetype/url

The first requirement is that a TrainingDML-AI XML document is a valid XML document.

Requirement 1	/req/base/xmlbasetype/xml An instance SHALL be a conformant XML document, as defined by W3C Recommendation
---------------	---

XML has a limited range of built-in types (<https://www.w3.org/TR/xmlschema11-2/>). The following requirements provide standard XML representations of additional types required across all requirements within this specification.

The DateTime is encoded as an XML element defined as one of three elements of type “date”, “time”, “dateTime”.

Requirement 2	/req/base/xmlbasetype/datetime Each DateTime value SHALL be encoded as a text string defined in Section 3.3.7, Section 3.3.8 or Section 3.3.9 of W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes. The specification of date and time in any XML encoding of training set data SHALL be specified in UTC.
---------------	--

Examples:

- a) "2002-05-30T09:30:10Z"
- b) "2002-09-24"

The NamedValue is encoded as an XML element with two elements named “key” and “value”. Typically, the “key” element represents a unique identifier or name for the value being described, while the “value” element contains the actual data associated with that identifier. The “key” element is an open field allowing for arbitrary keys to be used. The “value” element may be any types.

Requirement 3	/req/base/xmlbasetype/namedvalue Each NamedValue value SHALL be encoded as an XML element with elements “key” and “value”, while the type of “key” is CharacterString.
---------------	--

Examples:

- a) `<key>forest</key>`
`<value>RGB(0,255,255)</value>`
- b) `<key>precision</key>`
`<value>0.8</value>`

The URL is encoded as an XML element defined as the type of “anyURI”.

Requirement 4	/req/base/xmlbasetype/url Each URL value SHALL be encoded as a text string defined in Section 3.3.17 of W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes.
---------------	---

Examples:

- a) "http://www.opengeospatial.org"
- b) "/file.txt"
- c) "../dataset/file.txt"

7.1.2. Requirements Class: ISO Metadata Type

The ISO Metadata Type requirements class defines the requirements for XML encoding of ISO metadata types.

REQUIREMENTS CLASS

/req/base/isometadatatype

Dependency XML

Requirement 5 /req/base/isometadatatype/band

Requirement 6 /req/base/isometadatatype/extent

Requirement 7 /req/base/isometadatatype/citation

Requirement 8 /req/base/isometadatatype/scope

Requirement 9 /req/base/isometadatatype/referencesystem

Requirement 10 /req/base/isometadatatype/lineage

Requirement 11 /req/base/isometadatatype/resolution

The MD_Band is encoded as an XML element.

Requirement 5 /req/base/isometadatatype/band
Each MD_Band value SHALL be encoded as an XML element matching the XML Schema type as defined in:
<https://schemas.isotc211.org/19115/-1/mrc/1.3.0/content.xsd>

Examples:

```
a) <mrc:name>
    <mcc:MD_Identifier>
      <mcc:code>
        <gco:CharacterString>red</gco:CharacterString>
      </mcc:code>
    </mcc:MD_Identifier>
  </mrc:name>
b) <mrc:name>
    <mcc:MD_Identifier>
      <mcc:code>
        <gco:CharacterString>B4</gco:CharacterString>
      </mcc:code>
    </mcc:MD_Identifier>
  </mrc:name>
c) <mrc:boundMax>
    <gco:Real>690</gco:Real>
  </mrc:boundMax>
  <mrc:boundMin>
    <gco:Real>630</gco:Real>
  </mrc:boundMin>
  <mrc:boundUnits>
    <gml:UnitDefinition gml:id="ID">
```

```

        <gml:identifier codeSpace="http://www.example.com/">nm</gml:
identifier>
    </gml:UnitDefinition>
</mrc:boundUnits>

```

The EX_Extent is encoded as an XML element.

Requirement 6	/req/base/isometadatatype/extent Each EX_Extent value SHALL be encoded as an XML element matching the XML Schema type as defined in: https://schemas.isotc211.org/19115/-1/gex/1.3.0/extent.xsd
---------------	--

Examples:

```

a) <gex:geographicElement>
    <gex:EX_BoundingPolygon>
        <gex:polygon>
            <gml:Polygon>
                <gml:exterior>
                    <gml:LinearRing>
                        <gml:posList>
                            51.556272 -0.2803943 51.5562758 -0.2787397
51.5556539 -0.278736 51.5556501 -0.2803906 51.556272 -0.2803943
                        </gml:posList>
                    </gml:LinearRing>
                </gml:exterior>
            </gml:Polygon>
        </gex:polygon>
    </gex:EX_BoundingPolygon>
</gex:geographicElement>
b) <gex:geographicElement>
    <gex:EX_GeographicBoundingBox>
        <gex:westBoundLongitude>
            <gco:Decimal>-171.76409</gco:Decimal>
        </gex:westBoundLongitude>
        <gex:eastBoundLongitude>
            <gco:Decimal>-157.86768</gco:Decimal>
        </gex:eastBoundLongitude>
        <gex:southBoundLatitude>
            <gco:Decimal>-14.42443</gco:Decimal>
        </gex:southBoundLatitude>
        <gex:northBoundLatitude>
            <gco:Decimal>21.31573</gco:Decimal>
        </gex:northBoundLatitude>
    </gex:EX_GeographicBoundingBox>
</gex:geographicElement>

```

The CI_Citation is encoded as an XML element.

Requirement 7	/req/base/isometadatatype/citation Each CI_Citation value SHALL be encoded as an XML element matching the XML Schema type as defined in: https://schemas.isotc211.org/19115/-1/cit/1.3.0/citation.xsd
---------------	--

Example:

```

<cit:title>

```



```

    <gco:CharacterString>Open Geospatial Consortium</gco:CharacterString>
  </cit:title>
  <cit:alternateTitle>
    <gco:CharacterString>OGC</gco:CharacterString>
  </cit:alternateTitle>
  <cit:identifier>
    <mcc:MD_Identifier>
      <mcc:code>
        <gco:CharacterString>artifact_id=104605</gco:CharacterString>
      </mcc:code>
      <mcc:version>
        <gco:CharacterString>1</gco:CharacterString>
      </mcc:version>
    </mcc:MD_Identifier>
  </cit:identifier>

```

The MD_Scope is encoded as an XML element.

Requirement 8	/req/base/isometadatatype/scope Each MD_Scope value SHALL be encoded as an XML element matching the XML Schema type as defined in: https://schemas.isotc211.org/19115/-1/mcc/1.3.0/commonClasses.xsd
---------------	--

Examples:

```

a) <mcc:level>
    <mcc:MD_ScopeCode codeList="http://standards.iso.org/iso/19115/resources/
CodeList/cat/codelists.xml#MD_ScopeCode" codeListValue="dataset"/>
  </mcc:level>
  <mcc:levelDescription>
    <mcc:MD_ScopeDescription>
      <mcc:dataset>
        <gco:CharacterString>whu_rs19</gco:CharacterString>
      </mcc:dataset>
    </mcc:MD_ScopeDescription>
  </mcc:levelDescription>
b) <mcc:level>
    <mcc:MD_ScopeCode codeList="http://standards.iso.org/iso/19115/resources/
CodeList/cat/codelists.xml#MD_ScopeCode" codeListValue="attribute"/>
  </mcc:level>
  <mcc:levelDescription>
    <mcc:MD_ScopeDescription>
      <mcc:attributes>
        <gco:CharacterString>completeness</gco:CharacterString>
      </mcc:attributes>
    </mcc:MD_ScopeDescription>
  </mcc:levelDescription>
c) <mcc:level>
    <mcc:MD_ScopeCode codeList="http://standards.iso.org/iso/19115/resources/
CodeList/cat/codelists.xml#MD_ScopeCode" codeListValue="coverage"/>
  </mcc:level>
  <mcc:extent>
    <gex:EX_Extent>
      <gex:geographicElement>
        <gex:EX_GeographicBoundingBox>
          <gex:westBoundLongitude>
            <gco:Decimal>-171.76409</gco:Decimal>
          </gex:westBoundLongitude>
          <gex:eastBoundLongitude>
            <gco:Decimal>-157.86768</gco:Decimal>
          </gex:eastBoundLongitude>
        </gex:EX_GeographicBoundingBox>
      </gex:geographicElement>
    </gex:EX_Extent>
  </mcc:extent>

```

```

        </gex:eastBoundLongitude>
        <gex:southBoundLatitude>
            <gco:Decimal>-14.42443</gco:Decimal>
        </gex:southBoundLatitude>
        <gex:northBoundLatitude>
            <gco:Decimal>21.31573</gco:Decimal>
        </gex:northBoundLatitude>
    </gex:EX_GeographicBoundingBox>
</gex:geographicElement>
</gex:EX_Extent>
</mcc:extent>
d) <mcc:level>
    <mcc:MD_ScopeCode codeList="http://standards.iso.org/iso/19115/resources/
CodeList/cat/codeLists.xml#MD_ScopeCode" codeListValue="feature"/>
</mcc:level>
<mcc:extent>
    <gex:EX_Extent>
        <gex:geographicElement>
            <gex:EX_BoundingPolygon>
                <gex:polygon>
                    <gml:Polygon>
                        <gml:exterior>
                            <gml:LinearRing>
                                <gml:posList>
                                    51.556272 -0.2803943 51.5562758 -
0.2787397 51.5556539 -0.278736 51.5556501 -0.2803906 51.556272 -0.2803943
                                </gml:posList>
                            </gml:LinearRing>
                        </gml:exterior>
                    </gml:Polygon>
                </gex:polygon>
            </gex:EX_BoundingPolygon>
        </gex:geographicElement>
    </gex:EX_Extent>
</mcc:extent>

```

The MD_ReferenceSystem is encoded as an XML object.

Requirement 9	/req/base/isometadatatype/referencesystem Each MD_ReferenceSystem value SHALL be encoded as an XML object matching the XML Schema type as defined in: https://standards.iso.org/iso/19115/-3/mrs/1.0/referenceSystem.xsd
---------------	---

Examples:

```

a) <mrs:referenceSystemIdentifier>
    <mcc:MD_Identifier>
        <mcc:code>
            <gco:CharacterString>EPSG:3857</gco:CharacterString>
        </mcc:code>
    </mcc:MD_Identifier>
</mrs:referenceSystemIdentifier>
b) <mrs:referenceSystemIdentifier>
    <mcc:MD_Identifier>
        <mcc:code>
            <gco:CharacterString>EPSG:4326</gco:CharacterString>
        </mcc:code>
    </mcc:MD_Identifier>
</mrs:referenceSystemIdentifier>

```

The LI_Lineage is encoded as an XML object.

Requirement 10	/req/base/isometadatatype/lineage Each LI_Lineage value SHALL be encoded as an XML object matching the XML Schema type as defined in: https://schemas.isotc211.org/19115/-1/mrl/1.3.0/lineage.xsd
----------------	--

Examples:

```
<mrl:LI_Lineage>
  <mrl:statement>
    <gco:CharacterString>
      The dataset was generated from high-resolution satellite imagery
      through a standardized
      preprocessing and manual labeling workflow for AI model training.
    </gco:CharacterString>
  </mrl:statement>
  <mrl:source>
    <mrl:LI_Source>
      <mrl:description>
        <gco:CharacterString>
          High-resolution multispectral satellite imagery acquired in
          2023 covering urban
          areas.
        </gco:CharacterString>
      </mrl:description>
      <mrl:sourceCitation>
        <cit:CI_Citation>
          <cit:title>
            <gco:CharacterString>
              Urban Satellite Imagery Dataset 2023
            </gco:CharacterString>
          </cit:title>
        </cit:CI_Citation>
      </mrl:sourceCitation>
    </mrl:LI_Source>
  </mrl:source>
  <mrl:processStep>
    <mrl:LI_ProcessStep>
      <mrl:description>
        <gco:CharacterString>
          Radiometric calibration and geometric correction were
          applied to the raw
          satellite
          images.
        </gco:CharacterString>
      </mrl:description>
      <mrl:rationale>
        <gco:CharacterString>
          To ensure spatial alignment and consistent reflectance
          values for subsequent
          labeling.
        </gco:CharacterString>
      </mrl:rationale>
      <mrl:stepDateTime>
        <gml:TimeInstant>
          <gml:timePosition>2023-06-15T10:00:00Z</gml:timePosition>
        </gml:TimeInstant>
      </mrl:stepDateTime>
    </mrl:LI_ProcessStep>
  </mrl:processStep>
</mrl:LI_Lineage>
```

```

    </mrl:processStep>
  <mrl:processStep>
    <mrl:LI_ProcessStep>
      <mrl:description>
        <gco:CharacterString>
          Trained annotators manually labeled buildings and roads
using a GIS-based
          annotation
          platform.
        </gco:CharacterString>
      </mrl:description>
      <mrl:rationale>
        <gco:CharacterString>
          To generate high-quality ground truth data for supervised
deep learning models.
        </gco:CharacterString>
      </mrl:rationale>
      <mrl:stepDateTime>
        <gml:TimePeriod>
          <gml:beginPosition>2023-06-20</gml:beginPosition>
          <gml:endPosition>2023-07-10</gml:endPosition>
        </gml:TimePeriod>
      </mrl:stepDateTime>
    </mrl:LI_ProcessStep>
  </mrl:processStep>
</mrl:LI_Lineage>

```

The MD_Resolution is encoded as an XML object.

Requirement 11	/req/base/isometadatatype/resolution Each MD_Resolution value SHALL be encoded as an XML object matching the XML Schema type as defined in: https://standards.iso.org/iso/19115/-3/mri/1.0/identification.xsd
----------------	---

Examples:

```

a) <mri:equivalentScale>
    <mri:MD_RepresentativeFraction>
      <mri:denominator>
        <gco:Integer>10000</gco:Integer>
      </mri:denominator>
    </mri:MD_RepresentativeFraction>
  </mri:equivalentScale>
b) <mri:distance>
    <gco:Distance uom="m">10</gco:Distance>
  </mri:distance>
c) <mri:vertical>
    <gco:Distance uom="m">1</gco:Distance>
  </mri:vertical>
d) <mri:angularDistance>
    <gco:Angle uom="degree">0.01</gco:Angle>
  </mri:angularDistance>
e) <mri:levelOfDetail>
    <gco:CharacterString>High resolution imagery suitable for urban analysis
  </gco:CharacterString>
</mri:levelOfDetail>

```

7.1.3. Requirements Class: ISO Quality Type

The ISO Quality Type requirements class defines the requirements for XML encoding of ISO quality types.

REQUIREMENTS CLASS	
/req/base/isoqualitytype	
Dependency	XML
Requirement 12	/req/base/isoqualitytype/element

The QualityElement object is encoded as an XML element with properties shown in Table 1.

Requirement 12	/req/base/isoqualitytype/element Each QualityElement value SHALL be encoded as an XML element with properties shown in Table 1.
----------------	--

Table 1 — QualityElement properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
type	The type of the quality element object.	CharacterString [1..1]	Mandatory
measure	Reference to measure used.	MeasureReference [1..1]	Mandatory
evaluationMethod	Evaluation information.	EvaluationMethod [1..*]	Mandatory
result	Value obtained from applying a data quality measure.	QualityResult [1..*]	Mandatory

Example:

```
<type>FormatConsistency</type>
<measure>
  <measureDescription>
    Percentage of training samples with inconsistent image format
  </measureDescription>
</measure>
<evaluationMethod>
  <evaluationMethodDescription>
    Full test method to calculate the percentage of training samples with
    an inconsistent format
```

```

        </evaluationMethodDescription>
    </evaluationMethod>
    <result>
        <quantitativeResult>
            <value>
                0
            </value>
            <valueUnit>
                %
            </valueUnit>
        </quantitativeResult>
    </result>

```

7.1.4. Requirements Class: Geospatial Type

The Geospatial Type requirements class defines the requirements for XML encoding of geospatial types.

REQUIREMENTS CLASS	
/req/base/geospatialtype	
Dependency	XML
Dependency	GML
Requirement 13	/req/base/geospatialtype/feature

The encoding of one or more features follows GML for encoding a Feature element, with members “type”, “geometry” and “properties”. A Feature element represents a spatially bounded thing. Every Feature element is a GML element no matter where it occurs in a GML text.

Requirement 13	/req/base/geospatialtype/feature
	Each Feature value SHALL be encoded using the GML feature encoding defined by OpenGIS in GML Section 3.2:
	http://www.opengis.net/gml/3.2

Examples of Feature encodings are:

```

a) <gml:metaDataProperty>
    <gml:GenericMetaData>
        <truncated>0.0</truncated>
        <occluded>0</occluded>
        <alpha>-0.2</alpha>
    </gml:GenericMetaData>
</gml:metaDataProperty>
<gml:location>
    <gml:GeometricComplex>
        <gml:element>
            <gml:Polygon>

```

```

        <gml:exterior>
          <gml:LinearRing>
            <gml:posList>
              51.556272 -0.2803943 51.5562758 -0.2787397
51.5556539 -0.278736 51.5556501 -0.2803906 51.556272 -0.2803943
            </gml:posList>
          </gml:LinearRing>
        </gml:exterior>
      </gml:Polygon>
    </gml:element>
  </gml:GeometricComplex>
</gml:location>
b) <gml:metaDataProperty>
  <gml:GenericMetaData>
    <iscrowd>0</iscrowd>
    <area>2580</area>
  </gml:GenericMetaData>
</gml:metaDataProperty>
<gml:location>
  <gml:GeometricComplex>
    <gml:element>
      <gml:LineString>
        <gml:posList>1.0 1.0 2.0 2.0</gml:posList>
      </gml:LineString>
    </gml:element>
  </gml:GeometricComplex>
</gml:location>
c) <gml:location>
  <gml:GeometricComplex>
    <gml:element>
      <gml:Point>
        <gml:pos>1.0 1.0</gml:pos>
      </gml:Point>
    </gml:element>
  </gml:GeometricComplex>
</gml:location>

```

7.2. Requirements Class: AI_TrainingDataset

The AI_TrainingDataset requirements class defines an XML encoding for the AI_TrainingDataset module, which is based on the UML model specified in the TrainingDML-AI Part 1: Conceptual Model Standard.

REQUIREMENTS CLASS

/req/aitrainingdataset

Dependency	XML
------------	-----

Dependency	/req/base/xmlbasetype
------------	-----------------------

REQUIREMENTS CLASS	
Dependency	/req/base/isometadatatype
Dependency	/req/aitrainingdata
Dependency	/req/aitask
Dependency	/req/ailabeling
Dependency	/req/aidataquality
Dependency	/req/aitdchangeset
Requirement 14	/req/aitrainingdataset/trainingdataset
Requirement 15	/req/aitrainingdataset/metricsinliterature
Requirement 16	/req/aitrainingdataset/eotrainingdataset

The AI_TrainingDataset object is encoded as an XML element with properties shown in Table 2.

Requirement 14	/req/aitrainingdataset/trainingdataset Each AI_TrainingDataset object SHALL implement the Mandatory properties shown in Table 2.
----------------	---

Table 2 – AI_TrainingDataset properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
type	Type of the training dataset.	"AI_AbstractTraining Dataset"	Mandatory
id	Identification of the AI training dataset.	CharacterString [1..1]	Mandatory
doi	Digital object identifier of the AI training dataset.	CharacterString [0..1]	Optional
scope	Description of the scope of the training dataset.	MD_Scope [0..1]	Optional
name	Name of the AI training dataset.	CharacterString [1..1]	Mandatory
description	Description of the AI training dataset.	CharacterString [1..1]	Mandatory

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
version	Version number of the AI training dataset.	CharacterString [0..1]	Optional
amountOfTrainingData	Total number of training samples in the AI training dataset.	Int [0..1]	Optional
createdTime	Time when the AI training dataset was created.	DateTime [0..1]	Optional
updatedAtTime	Time when the AI training dataset was updated.	DateTime [0..1]	Optional
license	License description of the AI training dataset.	CharacterString [1..1]	Mandatory
providers	People or organizations who provide the AI training dataset.	CharacterString [0..*]	Optional
keywords	Keywords of the AI training dataset.	CharacterString [0..*]	Optional
metricsInLIT	Results of performance metrics achieved by AI/ML algorithms in the peer-reviewed literature.	AI_MetricsInLiterature [0..*]	Optional
statisticsInfo	Statistical results for training samples in each class.	NamedValue [0..*]	Optional
dataSources	Citation for the data sources.	CI_Citation [0..*]	Optional
numberOfClasses	Total number of classes in the AI training dataset.	Int [0..1]	Optional
classificationScheme	Classification scheme for classes used in the AI training dataset.	CharacterString [0..1]	Optional
classes	Classes used in the AI training dataset.	NamedValue [1..1]	Optional
extent	Spatial extent of the training dataset.	EX_Extent [0..1]	Optional
referenceSystem	Reference system of the training dataset.	MD_ReferenceSystem [0..1]	Optional
lineage	Information about the provenance, source(s), and/or the production process	LI_Lineage [0..*]	Optional

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
	(es) applied to the training dataset.		
tasks	Task description of the training dataset.	AI_Task [1..*]	Mandatory
labeling	Provenance information of how the training dataset is labeled.	AI_Labeling [0..*]	Optional
quality	Quality information of the training dataset.	DataQuality [0..*]	Optional
changesets	Changeset between two versions of the training dataset.	AI_TDChangeset [0..*]	Optional
data	Training data in the training dataset.	AI_AbstractTrainingData [1..*]	Mandatory

Example:

```

<type>AI_AbstractTrainingDataset</type>
<id>whu_rs19</id>
<name>WHU-RS19</name>
<description>Wuhan University-Remote Sensing 19 Categories (WHU-RS19) has 19
classes of remote sensing images scenes obtained from Google Earth</description>
<license>CC BY-SA 4.0</license>
<amountOfTrainingData>1013</amountOfTrainingData>
<createdTime>2010-01-01T00:00:00</createdTime>
<providers>Wuhan University</providers>
<keywords>Remote Sensing</keywords>
<keywords>Scene Classification</keywords>
<numberOfClasses>19</numberOfClasses>
<classes>
  <key>Airport</key>
  <value>null</value>
</classes>
<classes>
  <key>Beach</key>
  <value>null</value>
</classes>
<classes>
  <key>Bridge</key>
  <value>null</value>
</classes>
<classes>
  <key>Commercial</key>
  <value>null</value>
</classes>
<classes>
  <key>Desert</key>
  <value>null</value>
</classes>
<classes>
  <key>Farmland</key>
  <value>null</value>
</classes>

```

```

<classes>
  <key>footballField</key>
  <value>null</value>
</classes>
<classes>
  <key>Forest</key>
  <value>null</value>
</classes>
<classes>
  <key>Industrial</key>
  <value>null</value>
</classes>
<classes>
  <key>Meadow</key>
  <value>null</value>
</classes>
<classes>
  <key>Mountain</key>
  <value>null</value>
</classes>
<classes>
  <key>Park</key>
  <value>null</value>
</classes>
<classes>
  <key>Parking</key>
  <value>null</value>
</classes>
<classes>
  <key>Pond</key>
  <value>null</value>
</classes>
<classes>
  <key>Port</key>
  <value>null</value>
</classes>
<classes>
  <key>railwayStation</key>
  <value>null</value>
</classes>
<classes>
  <key>Residential</key>
  <value>null</value>
</classes>
<classes>
  <key>River</key>
  <value>null</value>
</classes>
<classes>
  <key>Viaduct</key>
  <value>null</value>
</classes>
<extent>
  <gex:geographicElement>
    <gex:EX_GeographicBoundingBox>
      <gex:westBoundLongitude>
        <gco:Decimal>-180</gco:Decimal>
      </gex:westBoundLongitude>
      <gex:eastBoundLongitude>
        <gco:Decimal>180</gco:Decimal>
      </gex:eastBoundLongitude>
      <gex:southBoundLatitude>
        <gco:Decimal>-90</gco:Decimal>
      </gex:southBoundLatitude>
    </gex:EX_GeographicBoundingBox>
  </gex:geographicElement>
</extent>

```

```

        </gex:southBoundLatitude>
        <gex:northBoundLatitude>
          <gco:Decimal>90</gco:Decimal>
        </gex:northBoundLatitude>
      </gex:EX_GeographicBoundingBox>
    </gex:geographicElement>
  </extent>
  <mrs:MD_ReferenceSystem>
    <mrs:referenceSystemIdentifier>
      <mcc:MD_Identifier>
        <mcc:code>
          <gco:CharacterString>EPSG:4326</gco:CharacterString>
        </mcc:code>
      </mcc:MD_Identifier>
    </mrs:referenceSystemIdentifier>
  </mrs:MD_ReferenceSystem>
  <tasks>
    <type>AI_EOTask</type>
    <id>whu_rs19-task</id>
    <description>Structural high-resolution satellite image indexing</
description>
    <taskType>Scene Classification</taskType>
  </tasks>
  <data>
    <type>AI_EOTrainingData</type>
    <id>airport_01</id>
    <dataSources>
      <cit:title>
        <gco:CharacterString>googleEarth</gco:CharacterString>
      </cit:title>
    </dataSources>
    <dataURL>image/Airport/airport_01.jpg</dataURL>
    <labels>
      <type>AI_SceneLabel</type>
      <class>Airport</class>
    </labels>
  </data>

```

If the optional element `AI_MetricsInLiterature` is specified, this element is encoded as XML element with properties as shown in Table 3.

Requirement 15	/req/aitrainingdataset/metricsinliterature Each <code>AI_MetricsInLiterature</code> value SHALL implement the Mandatory properties shown in Table 3.
----------------	---

Table 3 — `AI_MetricsInLiterature` properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
doi	Digital object identifier of the peer-reviewed literature.	CharacterString [1..1]	Mandatory
algorithm	AI/ML algorithms used in the peer-reviewed literature.	CharacterString [0..1]	Optional

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
metrics	Metrics and results of AI/ML algorithms in the peer-reviewed literature.	NamedValue [1..*]	Mandatory

Example:

```

<doi>10.1109/TGRS.2019.2917161</doi>
<algorithm>FACNN</algorithm>
<metrics>
  <key>Overall Accuracy</key>
  <value>0.9881</value>
</metrics>

```

The AI_EOTrainingDataset object is encoded as an XML element with properties shown in Table 2 and Table 4.

Requirement 16	/req/aitrainingdataset/eotrainingdataset Each AI_EOTrainingDataset object SHALL implement the Mandatory properties both shown in Table 2 and Table 4.		
----------------	--	--	--

Table 4 – AI_EOTrainingDataset properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
type	Type of the training dataset.	"AI_EOTrainingDataset"	Mandatory
bands	Description of the image bands used in the EO training dataset.	MD_Band [0..*]	Optional
imageSize	Size of the images used in the EO training dataset.	CharacterString [0..1]	Optional

Example:

```

<type>AI_EOTrainingDataset</type>
<id>whu_rs19</id>
<name>WHU-RS19</name>
<description>Wuhan University-Remote Sensing 19 Categories (WHU-RS19) has 19
classes of remote sensing images scenes obtained from Google Earth</description>
<license>CC BY-SA 4.0</license>
<amountOfTrainingData>1013</amountOfTrainingData>
<createdTime>2010-01-01T00:00:00</createdTime>
<providers>Wuhan University</providers>
<keywords>Remote Sensing</keywords>
<keywords>Scene Classification</keywords>
<numberOfClasses>19</numberOfClasses>
<classes>
  <key>Airport</key>
  <value>null</value>
</classes>

```

```

<classes>
  <key>Beach</key>
  <value>null</value>
</classes>
<classes>
  <key>Bridge</key>
  <value>null</value>
</classes>
<classes>
  <key>Commercial</key>
  <value>null</value>
</classes>
<classes>
  <key>Desert</key>
  <value>null</value>
</classes>
<classes>
  <key>Farmland</key>
  <value>null</value>
</classes>
<classes>
  <key>footballField</key>
  <value>null</value>
</classes>
<classes>
  <key>Forest</key>
  <value>null</value>
</classes>
<classes>
  <key>Industrial</key>
  <value>null</value>
</classes>
<classes>
  <key>Meadow</key>
  <value>null</value>
</classes>
<classes>
  <key>Mountain</key>
  <value>null</value>
</classes>
<classes>
  <key>Park</key>
  <value>null</value>
</classes>
<classes>
  <key>Parking</key>
  <value>null</value>
</classes>
<classes>
  <key>Pond</key>
  <value>null</value>
</classes>
<classes>
  <key>Port</key>
  <value>null</value>
</classes>
<classes>
  <key>railwayStation</key>
  <value>null</value>
</classes>
<classes>
  <key>Residential</key>
  <value>null</value>

```

```

</classes>
<classes>
  <key>River</key>
  <value>null</value>
</classes>
<classes>
  <key>Viaduct</key>
  <value>null</value>
</classes>
<extent>
  <gex:geographicElement>
    <gex:EX_GeographicBoundingBox>
      <gex:westBoundLongitude>
        <gco:Decimal>-180</gco:Decimal>
      </gex:westBoundLongitude>
      <gex:eastBoundLongitude>
        <gco:Decimal>180</gco:Decimal>
      </gex:eastBoundLongitude>
      <gex:southBoundLatitude>
        <gco:Decimal>-90</gco:Decimal>
      </gex:southBoundLatitude>
      <gex:northBoundLatitude>
        <gco:Decimal>90</gco:Decimal>
      </gex:northBoundLatitude>
    </gex:EX_GeographicBoundingBox>
  </gex:geographicElement>
</extent>
<mrs:MD_ReferenceSystem>
<mrs:referenceSystemIdentifier>
  <mcc:MD_Identifier>
    <mcc:code>
      <gco:CharacterString>EPSG:4326</gco:CharacterString>
    </mcc:code>
  </mcc:MD_Identifier>
</mrs:referenceSystemIdentifier>
</mrs:MD_ReferenceSystem>
<bands>
  <mrc:name>
    <mcc:MD_Identifier>
      <mcc:code>
        <gco:CharacterString>red</gco:CharacterString>
      </mcc:code>
    </mcc:MD_Identifier>
  </mrc:name>
</bands>
<bands>
  <mrc:name>
    <mcc:MD_Identifier>
      <mcc:code>
        <gco:CharacterString>green</gco:CharacterString>
      </mcc:code>
    </mcc:MD_Identifier>
  </mrc:name>
</bands>
<bands>
  <mrc:name>
    <mcc:MD_Identifier>
      <mcc:code>
        <gco:CharacterString>blue</gco:CharacterString>
      </mcc:code>
    </mcc:MD_Identifier>
  </mrc:name>
</bands>

```

```

<imageSize>6000x7600</imageSize>
<tasks>
  <type>AI_EOTask</type>
  <id>whu_rs19-task</id>
  <description>Structural high-resolution satellite image indexing</
description>
  <taskType>Scene Classification</taskType>
</tasks>
<data>
  <type>AI_EOTrainingData</type>
  <id>airport_01</id>
  <dataSources>
    <cit:title>
      <gco:CharacterString>googleEarth</gco:CharacterString>
    </cit:title>
  </dataSources>
  <dataURL>image/Airport/airport_01.jpg</dataURL>
  <labels>
    <type>AI_SceneLabel</type>
    <class>Airport</class>
  </labels>
</data>

```

7.3. Requirements Class: AI_TrainingData

The AI_TrainingData requirements class defines an XML encoding for the AI_TrainingData module, which is based on the UML model specified in the TrainingDML-AI Part 1: Conceptual Model Standard.

REQUIREMENTS CLASS	
/req/aitrainingdata	
Dependency	XML
Dependency	/req/base/xmlbasetype
Dependency	/req/base/isometadatatype
Dependency	/req/ailabel
Dependency	/req/ailabeling
Dependency	/req/aidataquality
Requirement 17	/req/aitrainingdata/trainingdata

REQUIREMENTS CLASS

Requirement 18 /req/aitrainingdata/trainingtypecode

Requirement 19 /req/aitrainingdata/eotrainingdata

The AI_TrainingData object is encoded as an XML element with properties shown in Table 5.

Requirement 17 /req/aitrainingdata/trainingdata
Each AI_TrainingData object SHALL implement the Mandatory properties shown in Table 5.

Table 5 – AI_TrainingData properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
type	Type of the training sample.	"AI_AbstractTrainingData"	Mandatory
id	Identification of an individual AI training sample.	CharacterString [1..1]	Mandatory
datasetId	Identification of the training dataset that the training sample belongs to.	CharacterString [0..1]	Optional
trainingType	Training type of the individual AI training sample.	AI_TrainingTypeCode [0..1]	Optional
numberOfLabels	Total number of labels in the individual AI training sample.	Int [0..1]	Optional
dataSources	Citation of inputs to prepare a training sample.	CI_Citation [0..*]	Optional
extent	Spatial extent of the individual training sample.	EX_Extent [0..1]	Optional
dataTime	Date and time at which the geospatial data were obtained.	DateTime [0..*]	Optional
lineage	Information about the provenance, source(s), and/or the production process(es) applied to the training sample.	LI_Lineage [0..*]	Optional

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
labels	Labels in the training data.	AI_Label [0..*]	Optional
labeling	Provenance information of how the training data is labeled.	AI_Labeling [0..*]	Optional
quality	Quality information of the training data.	DataQuality [0..*]	Optional

Example:

```

<type>AI_AbstractTrainingData</type>
<id>airport_01</id>
<dataSources>
  <cit:title>
    <gco:CharacterString>googleEarth</gco:CharacterString>
  </cit:title>
</dataSources>
<dataURL>image/Airport/airport_01.jpg</dataURL>
<extent>
  <gex:geographicElement>
    <gex:EX_GeographicBoundingBox>
      <gex:westBoundLongitude>
        <gco:Decimal>-171.76409</gco:Decimal>
      </gex:westBoundLongitude>
      <gex:eastBoundLongitude>
        <gco:Decimal>-171.56578</gco:Decimal>
      </gex:eastBoundLongitude>
      <gex:southBoundLatitude>
        <gco:Decimal>-14.42443</gco:Decimal>
      </gex:southBoundLatitude>
      <gex:northBoundLatitude>
        <gco:Decimal>-14.32568</gco:Decimal>
      </gex:northBoundLatitude>
    </gex:EX_GeographicBoundingBox>
  </gex:geographicElement>
</extent>
<dateTime>2002-05-30T09:30:10Z</dateTime>
<labels>
  <type>AI_SceneLabel</type>
  <class>Airport</class>
</labels>

```

The AI_TrainingTypeCode is encoded as a text string whose value is one of “training”, “validation”, “test” or “retraining”.

Requirement 18	/req/aitrainingdata/trainingtypecode Each AI_TrainingTypeCode value SHALL be a text string whose value is one of “training”, “validation”, “test” or “retraining”.
----------------	---

Examples:

- a) "training"
- b) "validation"
- c) "test"

d) "retraining"

The AI_EOTrainingData object is encoded as an XML element with properties both shown in Table 5 and Table 6.

Requirement 19	/req/aitrainingdata/eotrainingdata
	Each AI_EOTrainingData object SHALL implement the Mandatory properties as defined in Table 5 and Table 6.

Table 6 – AI_EOTrainingData properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
type	Type of the EO training data.	"AI_EOTrainingData"	Mandatory
spatialResolution	Spatial resolution of the individual EO training sample.	MD_Resolution [0..1]	Optional
dataURL	URLs of the EO data, including both relative and absolute paths, which can encompass local paths, network addresses, and more.	URI [1..*]	Mandatory

Example:

```
<type>AI_EOTrainingData</type>
<id>airport_01</id>
<dataSources>
  <cit:title>
    <gco:CharacterString>googleEarth</gco:CharacterString>
  </cit:title>
</dataSources>
<extent>
  <gex:geographicElement>
    <gex:EX_GeographicBoundingBox>
      <gex:westBoundLongitude>
        <gco:Decimal>-171.76409</gco:Decimal>
      </gex:westBoundLongitude>
      <gex:eastBoundLongitude>
        <gco:Decimal>-171.56578</gco:Decimal>
      </gex:eastBoundLongitude>
      <gex:southBoundLatitude>
        <gco:Decimal>-14.42443</gco:Decimal>
      </gex:southBoundLatitude>
      <gex:northBoundLatitude>
        <gco:Decimal>-14.32568</gco:Decimal>
      </gex:northBoundLatitude>
    </gex:EX_GeographicBoundingBox>
  </gex:geographicElement>
</extent>
<spatialResolution>
```

```

<mri:MD_Resolution>
  <gco:Distance uom="meter">10</gco:Distance>
</mri:MD_Resolution>
</spatialResolution>
<dateTime>2002-05-30T09:30:10Z</dateTime>
<dataURL>image/Airport/airport_01.jpg</dataURL>
<labels>
  <type>AI_SceneLabel</type>
  <class>Airport</class>
</labels>

```

7.4. Requirements Class: AI_Task

The AI_Task requirements class defines an XML encoding for the AI_Task module, which is based on the UML model specified in the TrainingDML-AI Part 1: Conceptual Model Standard.

REQUIREMENTS CLASS	
/req/aitask	
Dependency	XML
Dependency	/req/base/xmlbasetype
Requirement 20	/req/aitask/task
Requirement 21	/req/aitask/eotask

The AI_Task object is encoded as an XML element with properties as shown in Table 7.

Requirement 20	/req/aitask/task Each AI_Task object SHALL implement the Mandatory properties shown in Table 7.
----------------	--

Table 7 — AI_Task properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
type	Type of the task object.	"AI_AbstractTask"	Mandatory
id	Identification of the task.	CharacterString [1..1]	Mandatory

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
datasetId	Identification of the training dataset the training sample belongs to.	CharacterString [0..1]	Optional
description	Description of the AI task.	CharacterString [0..1]	Optional

Example:

```
<type>AI_AbstractTask</type>
<id>image-indexing-task</id>
<description>Structural high-resolution satellite image indexing</description>
```

The AI_EOTask object is encoded as an XML element with properties both shown in Table 7 and Table 8.

Requirement 21	/req/aitask/eotask Each AI_EOTask object SHALL implement the Mandatory properties shown in Table 7 and Table 8.
----------------	--

Table 8 — AI_EOTask properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
type	Type of the task object.	"AI_EOTask"	Mandatory
taskType	Type of the EO task.	CharacterString [1..1]	Mandatory

Example:

```
<type>AI_EOTask</type>
<id>image-indexing-task</id>
<description>Structural high-resolution satellite image indexing</description>
<taskType>Scene Classification</taskType>
```

7.5. Requirements Class: AI_Label

The AI_Label requirements class defines an XML encoding for the AI_Label module, which is based on the UML model specified in the TrainingDML-AI Part 1: Conceptual Model Standard.

REQUIREMENTS CLASS	
/req/ailabel	
Dependency	XML
Dependency	/req/base/xmlbasetype
Dependency	/req/base/geospatialtype
Requirement 22	/req/ailabel/label
Requirement 23	/req/ailabel/scenelabel
Requirement 24	/req/ailabel/objectlabel
Requirement 25	/req/ailabel/pixellabel
Requirement 26	/req/ailabel/imageformatcode

The AI_Label object is encoded as an XML element with properties as shown in Table 9.

Requirement 22	/req/ailabel/label Each AI_Label object SHALL implement the Mandatory properties shown in Table 9.
----------------	---

Table 9 – AI_Label properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
type	Type of the label object.	"AI_AbstractLabel"	Mandatory
isNegative	Whether the training sample related to the label is a positive or negative sample.	Bool [0..1] Default: false	Optional
confidence	Confidence score of the labeler.	Float [0..1] Default: 1.0 Range: [0, 1]	Optional

Example:

```
<type>AI_AbstractLabel</type>
<isNegative>>false</isNegative>
<confidence>1.0</confidence>
```

The AI_SceneLabel object is encoded as an XML element with properties as shown in Table 10.

Requirement 23	/req/ailabel/scenelabel Each AI_SceneLabel object SHALL implement the Mandatory properties shown in Table 10.
----------------	--

Table 10 – AI_SceneLabel properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
type	Type of the label object at the scene level.	"AI_SceneLabel"	Mandatory
class	Class that records the semantic of the scene of the training sample.	CharacterString [1..1]	Mandatory

Example:

```
<type>AI_SceneLabel</type>  
<class>Airport</class>
```

The AI_ObjectLabel object is encoded as an XML element with properties shown in Table 11.

Requirement 24	/req/ailabel/objectlabel Each AI_ObjectLabel object SHALL implement the Mandatory properties shown in Table 11.
----------------	--

Table 11 – AI_ObjectLabel properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
type	Type of the label object at the object level.	"AI_ObjectLabel"	Mandatory
object	Feature that represents the position and attributes of the object.	Feature [0..1]	Mandatory
bboxType	Type of the bbox.	CharacterString [0..1]	Optional
class	Class that records the semantic of the object type.	CharacterString [1..1]	Mandatory
dateTime	Created time of the object label.	DateTime [0..1]	Optional

Example:

```
<type>AI_ObjectLabel</type>  
<object>  
  <gml:metaDataProperty>
```

```

    <gml:GenericMetaData>
      <truncated>0.0</truncated>
      <occluded>0</occluded>
      <alpha>-0.2</alpha>
    </gml:GenericMetaData>
  </gml:metaDataProperty>
  <gml:location>
    <gml:GeometricComplex>
      <gml:element>
        <gml:Polygon>
          <gml:exterior>
            <gml:LinearRing>
              <gml:posList>
                51.556272 -0.2803943 51.5562758 -0.2787397 51.5556539
                -0.278736 51.5556501 -0.2803906 51.556272 -0.2803943
              </gml:posList>
            </gml:LinearRing>
          </gml:exterior>
        </gml:Polygon>
      </gml:element>
    </gml:GeometricComplex>
  </gml:location>
</object>
<bboxType>Horizontal BBox</bboxType>
<class>Pedestrian</class>

```

The AI_PixelLabel object is encoded as an XML element with properties as shown in Table 12.

Requirement 25	/req/ailabel/pixellabel Each AI_PixelLabel object SHALL implement the Mandatory properties shown in Table 12.
----------------	--

Table 12 — AI_PixelLabel properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
type	Type of the label object at the pixel level.	"AI_PixelLabel"	Mandatory
imageUrl	URL of the images representing the label information.	URL [1..*]	Mandatory
imageFormat	Image data format.	AI_ImageFormatCode [1..*]	Mandatory

Example:

```

<type>AI_PixelLabel</type>
<imageUrl>/label_5classes/GF2_PMS1__L1A0000647767-MSS1_label.tif</imageUrl>
<imageFormat>image/tiff; application=geotiff</imageFormat>

```

The AI_ImageFormatCode is encoded as a text string whose value is defined in Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types [RFC 2046](#).

	/req/ailabel/imageformatcode
Requirement 26	Each AI_ImageFormatCode value SHALL be encoded as a text string defined in Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types RFC 2046 .

Examples:

- a) "image/tiff; application=geotiff"
- b) "application/x-netcdf"
- c) "image/png"
- d) "image/jp2"

7.6. Requirements Class: AI_Labeling

The AI_Labeling requirements class defines an XML encoding for the AI_Labeling module, which is based on the UML model specified in the TrainingDML-AI Part 1: Conceptual Model Standard.

REQUIREMENTS CLASS	
	/req/ailabeling
Dependency	XML
Dependency	/req/base/xmlbasetype
Dependency	/req/base/isometadatatype
Requirement 27	/req/ailabeling/labeling
Requirement 28	/req/ailabeling/labeler
Requirement 29	/req/ailabeling/labelingprocedure
Requirement 30	/req/ailabeling/labelingmethodcode

The AI_Labeling object is encoded as an XML element with properties shown in Table 13.

Requirement 27	/req/ailabeling/labeling
Each AI_Labeling object SHALL implement the Mandatory properties shown in Table 13.	

Table 13 – AI_Labeling properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
type	Type of the labeling object.	"AI_Labeling"	Mandatory
id	Identifier of the labeling.	CharacterString [1..1]	Mandatory
scope	Description of the scope of the labeling.	MD_Scope [1..1]	Mandatory
labelingTime	Date and time at which the labeling procedure was carried out.	DateTime [0..1]	Optional
labelers	Labelers of the labeling activity.	AI_Labeler [0..*]	Optional
procedure	Procedure used in the labeling activity.	AI_LabelingProcedure [0..1]	Optional

Example:

```

<type>AI_Labeling</type>
<id>0</id>
<scope>
  <mcc:level>
    <mcc:MD_ScopeCode codeList="http://standards.iso.org/iso/19115/resources/
Codelist/cat/codellists.xml#MD_ScopeCode" codeListValue="dataset"/>
  </mcc:level>
  <mcc:levelDescription>
    <mcc:MD_ScopeDescription>
      <mcc:dataset>
        <gco:CharacterString>whu_rs19</gco:CharacterString>
      </mcc:dataset>
    </mcc:MD_ScopeDescription>
  </mcc:levelDescription>
</scope>
<dateTime>2015-04-20T07:04:15Z</dateTime>
<labelers>...</labelers>
<procedure>...</procedure>

```

The AI_Labeler object is encoded as an XML element with properties as shown in Table 14.

Requirement 28	/req/ailabeling/labeler Each AI_Labeler object SHALL implement the Mandatory properties shown in Table 14.
----------------	---

Table 14 – AI_Labeler properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
type	Type of the labeler object.	"AI_Labeler"	Mandatory

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
id	Identifier of the labeler.	CharacterString [1..1]	Mandatory
name	Name of the labeler.	CharacterString [1..1]	Mandatory

Example:

```
<type>AI_Labeler</type>
<id>0</id>
<name>Tom</name>
```

The AI_LabelingProcedure object is encoded as an XML element with properties as shown in Table 15.

Requirement 29	/req/ailabeling/labelingprocedure Each AI_LabelingProcedure object SHALL implement the Mandatory properties shown in Table 15.
----------------	---

Table 15 — AI_LabelingProcedure properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
type	Type of the labeling procedure object.	"AI_LabelingProcedure"	Mandatory
id	Identifier of the labeling procedure.	CharacterString [1..1]	Mandatory
methods	Methods used in the labeling procedure.	AI_LabelingMethodCode [1..*]	Mandatory
tools	Tools or software used in the labeling procedure.	CharacterString [0..*]	Optional

Example:

```
<type>AI_LabelingProcedure</type>
<id>0</id>
<methods>manual</methods>
<tools>ArcGIS</tools>
```

The AI_LabelingMethodCode is encoded as a text string whose value is one of "manual", "semi-automatic", "automatic" or "unknown".

Requirement 30	/req/ailabeling/labelingmethodcode Each AI_LabelingMethodCode value SHALL be a text string whose value is one of "manual", "semi-automatic", "automatic" or "unknown".
----------------	---

Examples:

- a) "manual"
- b) "semi-automatic"
- c) "automatic"
- d) "unknown"

7.7. Requirements Class: AI_DataQuality

The AI_DataQuality requirements class defines an XML encoding for the AI_DataQuality module, which is based on the UML model specified in the TrainingDML-AI Part 1: Conceptual Model Standard.

REQUIREMENTS CLASS	
/req/aidataquality	
Dependency	XML
Dependency	/req/base/xmlbasetype
Dependency	/req/base/isometadatatype
Dependency	/req/base/isoqualitytype
Requirement 31	/req/aidataquality/classbalancedegree

The AI_ClassBalanceDegree object is encoded as an XML element with properties as shown in Table 16.

Requirement 31	/req/aidataquality/classbalancedegree
Each AI_ClassBalanceDegree object SHALL implement the Mandatory properties as shown in Table 16.	

Table 16 – AI_ClassBalanceDegree properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
type	Type of the class balance degree object.	"AI_ClassBalanceDegree"	Mandatory

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
measure	Reference to measure used.	MeasureReference [1..1]	Mandatory
evaluationMethod	Evaluation information.	EvaluationMethod [1..*]	Mandatory
result	Value obtained from applying a data quality measure.	QualityResult [1..*]	Mandatory

Example:

```

<type>AI_ClassBalanceDegree</type>
<measure>
  <measureDescription>
    Balance degree of label classes
  </measureDescription>
</measure>
<evaluationMethod>
  <evaluationMethodDescription>
    Counting the number of training samples belonging to each class and
    calculating the balance degree
  </evaluationMethodDescription>
</evaluationMethod>
<result>
  <quantitativeResult>
    <value>
      93.5
    </value>
    <valueUnit>
      %
    </valueUnit>
  </quantitativeResult>
</result>

```

7.8. Requirements Class: AI_TDChangeset

The AI_TDChangeset requirements class defines an XML encoding for the AI_TDChangeset module, which is based on the UML model specified in the TrainingDML-AI Part 1: Conceptual Model Standard.

REQUIREMENTS CLASS	
/req/aitdchangeset	
Dependency	XML

REQUIREMENTS CLASS	
Dependency	/req/base/xmlbasetype
Dependency	/req/base/isometadatatype
Dependency	/req/aitrainingdata
Requirement 32	/req/aitdchangeset/tdchangeset

The AI_TDChangeset object is encoded as an XML element with properties shown in Table 17.

Requirement 32	/req/aitdchangeset/tdchangeset Each AI_TDChangeset object SHALL implement the Mandatory properties as shown in Table 17.
----------------	---

Table 17 – AI_TDChangeset properties

XML PROPERTY	DEFINITION	DATA TYPE AND VALUES	OBLIGATION
type	Type of the TD changeset object.	"AI_TDChangeset"	Mandatory
id	Identifier of the changeset.	CharacterString [1..1]	Mandatory
datasetId	Identifier of the training dataset the changeset belongs to.	CharacterString [0..1]	Optional
version	Version of the training dataset that the changeset belongs to.	CharacterString [0..1]	Optional
changeCount	Total number of changed training samples.	Int [1..1]	Mandatory
createdTime	The time that the changeset was created.	DateTime [0..1]	Optional
add	Added training samples.	AI_AbstractTrainingData [0..*]	Optional
modify	Modified training samples.	AI_AbstractTrainingData [0..*]	Optional
delete	Deleted training samples.	AI_AbstractTrainingData [0..*]	Optional

Example:

```

<type>AI_TDChangeset</type>
<id>changeset-dota_v1.5</id>
<datasetId>dota_v1.5</datasetId>
<createdTime>2019-01-01T00:00:00</createdTime>
<changeCount>9</changeCount>
<modify>
  <type>E0TrainingData</type>
  <id>P1228</id>
  <dataSources>
    <cit:title>
      <gco:CharacterString>GF</gco:CharacterString>
    </cit:title>
  </dataSources>
  <dataURL>train/images/P1228.png</dataURL>
  <numberOfLabels>50</numberOfLabels>
  <trainingType>training</trainingType>
  <labels>
    <type>AI_ObjectLabel</type>
    <object>
      <gml:location>
        <gml:GeometricComplex>
          <gml:element>
            <gml:Polygon>
              <gml:exterior>
                <gml:LinearRing>
                  <gml:posList>
                    2306.0 729.0 2330.0 729.0 2330.0 744.0 2306.0
744.0 2306.0 729.0
                  </gml:posList>
                </gml:LinearRing>
              </gml:exterior>
            </gml:Polygon>
          </gml:element>
        </gml:GeometricComplex>
      </gml:location>
    </object>
    <bboxType>Horizontal BBox</bboxType>
    <class>ship</class>
  </labels>
</modify>

```



ANNEX A (NORMATIVE) ABSTRACT TEST SUITE



ANNEX A

(NORMATIVE)

ABSTRACT TEST SUITE

A.1. Introduction

Conformance is tested using the XML Schema document which formalize the requirements described above.

A.2. Conformance Class: Base

The Base conformance class tests that occurrences of the basic types are encoded according to the requirements.

A.2.1. Conformance Class: XML Base Type

CONFORMANCE CLASS A.1: XML BASE TYPE	
IDENTIFIER	<code>http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/xmlbasetype</code>
REQUIREMENTS CLASS	<code>/req/base/xmlbasetype</code>
CONFORMANCE TESTS	Abstract test A.1: <code>/conf/base/xmlbasetype/xml</code> Abstract test A.2: <code>/conf/base/xmlbasetype/datetime</code> Abstract test A.3: <code>/conf/base/xmlbasetype/namedvalue</code> Abstract test A.4: <code>/conf/base/xmlbasetype/url</code>

ABSTRACT TEST A.1	
IDENTIFIER	<code>/conf/base/xmlbasetype/xml</code>

ABSTRACT TEST A.1

REQUIREMENT /req/base/xmlbasetype/xml

TEST PURPOSE Verify that the document is well-formed XML.

TEST METHOD Load the document in an XML validator. Pass if no errors reported. Fail otherwise.

ABSTRACT TEST A.2

IDENTIFIER /conf/base/xmlbasetype/datetime

REQUIREMENT /req/base/xmlbasetype/datetime

TEST PURPOSE Verify that XML instance documents claiming conformance to this specification contain valid DateTime values according to [Section 3.3.7](#), [Section 3.3.8](#) or [Section 3.3.9](#) of W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes.

TEST METHOD Inspect the instance document to verify the above requirement.

ABSTRACT TEST A.3

IDENTIFIER /conf/base/xmlbasetype/namedvalue

REQUIREMENT /req/base/xmlbasetype/namedvalue

TEST PURPOSE Verify that XML instance documents claiming conformance to this specification validate against the XML schema specified in <http://schemas.opengis.net/trainingdml-ai/part3/1.0/namedValue.xsd>.

TEST METHOD Validate the instance document against the namedValue.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.4

IDENTIFIER /conf/base/xmlbasetype/url

REQUIREMENT /req/base/xmlbasetype/url

TEST PURPOSE Verify that XML instance documents claiming conformance to this specification contain valid URL values according to the W3C definition of anyURI, which represents an Internationalized

ABSTRACT TEST A.4

Resource Identifier Reference. An anyURI value can be absolute or relative and may have an optional fragment identifier.

TEST METHOD Inspect the instance document to verify the above requirement.

A.2.2. Conformance Class: ISO Metadata Type

CONFORMANCE CLASS A.2: ISO METADATA TYPE

IDENTIFIER `http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/isometadatatype`

REQUIREMENTS CLASS `/req/base/isometadatatype`

CONFORMANCE TESTS

- Abstract test A.5: `/conf/base/isometadatatype/band`
- Abstract test A.6: `/conf/base/isometadatatype/extent`
- Abstract test A.7: `/conf/base/isometadatatype/citation`
- Abstract test A.8: `/conf/base/isometadatatype/scope`
- Abstract test A.9: `/conf/base/isometadatatype/referencesystem`
- Abstract test A.10: `/conf/base/isometadatatype/lineage`
- Abstract test A.11: `/conf/base/isometadatatype/resolution`

ABSTRACT TEST A.5

IDENTIFIER `/conf/base/isometadatatype/band`

REQUIREMENT `/req/base/isometadatatype/band`

TEST PURPOSE Verify that instance documents using the MD_Band XML elements validate against the XML schema specified in <https://schemas.isotc211.org/19115/-1/mrc/1.3.0/content.xsd>.

TEST METHOD Validate the instance document against the content.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.6

IDENTIFIER `/conf/base/isometadatatype/extent`

REQUIREMENT `/req/base/isometadatatype/extent`

ABSTRACT TEST A.6

TEST PURPOSE	Verify that instance documents using the EX_Extent XML elements validate against the XML schema specified in https://schemas.isotc211.org/19115/-1/gex/1.3.0/extent.xsd .
TEST METHOD	Validate the instance document against the extent.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.7

IDENTIFIER	/conf/base/isometadatatype/citation
REQUIREMENT	/req/base/isometadatatype/citation
TEST PURPOSE	Verify that instance documents using the CI_Citation XML elements validate against the XML schema specified in https://schemas.isotc211.org/19115/-1/cit/1.3.0/citation.xsd .
TEST METHOD	Validate the instance document against the citation.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.8

IDENTIFIER	/conf/base/isometadatatype/scope
REQUIREMENT	/req/base/isometadatatype/scope
TEST PURPOSE	Verify that instance documents using the MD_Scope XML elements validate against the XML schema specified in https://schemas.isotc211.org/19115/-1/mcc/1.3.0/commonClasses.xsd .
TEST METHOD	Validate the instance document against the commonClasses.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.9

IDENTIFIER	/conf/base/isometadatatype/referencesystem
REQUIREMENT	/req/base/isometadatatype/referencesystem
TEST PURPOSE	Verify that instance documents using the MD_ReferenceSystem XML elements validate against the XML schema specified in https://schemas.isotc211.org/19115/-1/mcc/1.3.0/referenceSystem.xsd .

ABSTRACT TEST A.9

TEST METHOD	Validate the instance document against the commonClasses.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.
--------------------	---

ABSTRACT TEST A.10

IDENTIFIER /conf/base/isometadatatype/lineage

REQUIREMENT /req/base/isometadatatype/lineage

TEST PURPOSE Verify that instance documents using the LI_Lineage XML elements validate against the XML schema specified in <https://schemas.isotc211.org/19115/-1/mcc/1.3.0/lineage.xsd>.

TEST METHOD Validate the instance document against the commonClasses.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.11

IDENTIFIER /conf/base/isometadatatype/resolution

REQUIREMENT /req/base/isometadatatype/resolution

TEST PURPOSE Verify that instance documents using the MD_Resolution XML elements validate against the XML schema specified in <https://schemas.isotc211.org/19115/-1/mcc/1.3.0/resolution.xsd>.

TEST METHOD Validate the instance document against the commonClasses.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

A.2.3. Conformance Class: ISO Quality Type

CONFORMANCE CLASS A.3: ISO QUALITY TYPE

IDENTIFIER <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/isoqualitytype>

REQUIREMENTS CLASS /req/base/isoqualitytype

CONFORMANCE TEST Abstract test A.12: /conf/base/isoqualitytype/element

ABSTRACT TEST A.12

IDENTIFIER /conf/base/isoqualitytype/element

REQUIREMENT /req/base/isoqualitytype/element

TEST PURPOSE Verify that instance documents using the QualityElement XML elements validate against the XML schema specified in <https://schemas.isotc211.org/schemas/19157/-/mdq/1.2.0/dataQualityElement.xsd>.

TEST METHOD Validate the instance document against the dataQualityElement.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

A.2.4. Conformance Class: Geospatial Type

CONFORMANCE CLASS A.4: GEOSPATIAL TYPE

IDENTIFIER <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/geospatialtype>

REQUIREMENTS CLASS /req/base/geospatialtype

CONFORMANCE TEST Abstract test A.13: /conf/base/geospatialtype/feature

ABSTRACT TEST A.13

IDENTIFIER /conf/base/geospatialtype/feature

REQUIREMENT /req/base/geospatialtype/feature

TEST PURPOSE Verify that instance documents using the Feature XML elements validate against the XML schema specified in <http://schemas.opengis.net/gml/3.2.1/gml.xsd>.

TEST METHOD Validate the instance document against the gml.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

A.3. Conformance Class: AI_TrainingDataset

The AI_TrainingDataset conformance class tests that the training dataset object is encoded according to the requirements.

CONFORMANCE CLASS A.5: AI_TRAININGDATASET

IDENTIFIER	http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/aitrainingdataset
REQUIREMENTS CLASS	/req/aitrainingdataset
PREREQUISITES	Conformance class A.1: http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/xmlbasetype Conformance class A.2: http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/isometadatatype Conformance class A.6: http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/aitrainingdata Conformance class A.7: http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/aitask Conformance class A.9: http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/ailabeling Conformance class A.10: http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/aidataquality Conformance class A.11: http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/aitdchangeset
CONFORMANCE TESTS	Abstract test A.14: /conf/aitrainingdataset/trainingdataset Abstract test A.15: /conf/aitrainingdataset/metricsinliterature Abstract test A.16: /conf/aitrainingdataset/eotrainingdataset

ABSTRACT TEST A.14

IDENTIFIER	/conf/aitrainingdataset/trainingdataset
REQUIREMENT	/req/aitrainingdataset/trainingdataset
TEST PURPOSE	Verify that instance documents using the AI_TrainingDataset XML elements listed in Table 2 validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_trainingDataset.xsd .
TEST METHOD	Validate the instance document against the ai_trainingDataset.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.15

IDENTIFIER /conf/aitrainingdataset/metricsinliterature

REQUIREMENT /req/aitrainingdataset/metricsinliterature

TEST PURPOSE Verify that instance documents using the AI_MetricsInLiterature XML elements listed in Table 3 validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_metricsInLiterature.xsd.

TEST METHOD Validate the instance document against the ai_metricsInLiterature.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.16

IDENTIFIER /conf/aitrainingdataset/eotrainingdataset

REQUIREMENT /req/aitrainingdataset/eotrainingdataset

TEST PURPOSE Verify that instance documents using the AI_EOTrainingDataset XML elements listed in Table 2 and Table 4 validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_eoTrainingDataset.xsd.

TEST METHOD Validate the instance document against the ai_eoTrainingDataset.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

A.4. Conformance Class: AI_TrainingData

The AI_TrainingData conformance class tests that the training data objects are encoded according to the requirements.

CONFORMANCE CLASS A.6: AI_TRAININGDATA

IDENTIFIER <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/aitrainingdata>

REQUIREMENTS CLASS /req/aitrainingdata

PREREQUISITES Conformance class A.1: <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/xmlbasetype>

CONFORMANCE CLASS A.6: AI_TRAININGDATA

Conformance class A.2: <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/isometadatatype>
Conformance class A.8: <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/ailabel>
Conformance class A.9: <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/ailabeling>
Conformance class A.10: <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/aidataquality>

CONFORMANCE TESTS

Abstract test A.17: </conf/aitrainingdata/trainingdata>
Abstract test A.18: </conf/aitrainingdata/trainingtypecode>
Abstract test A.19: </conf/aitrainingdata/eotrainingdata>

ABSTRACT TEST A.17

IDENTIFIER </conf/aitrainingdata/trainingdata>

REQUIREMENT </req/aitrainingdata/trainingdata>

TEST PURPOSE Verify that instance documents using the AI_TrainingData XML elements listed in Table 5 validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_trainingData.xsd.

TEST METHOD Validate the instance document against the ai_trainingData.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.18

IDENTIFIER </conf/aitrainingdata/trainingtypecode>

REQUIREMENT </req/aitrainingdata/trainingtypecode>

TEST PURPOSE Verify that instance documents using the AI_TrainingTypeCode XML elements validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_trainingTypeCode.xsd.

TEST METHOD Validate the instance document against the ai_trainingTypeCode.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.19

IDENTIFIER /conf/aitrainingdata/eotrainingdata

REQUIREMENT /req/aitrainingdata/eotrainingdata

TEST PURPOSE Verify that instance documents using the AI_EOTrainingData XML elements listed in Table 5 and Table 6 validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_eoTrainingData.xsd.

TEST METHOD Validate the instance document against the ai_eoTrainingData.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

A.5. Conformance Class: AI_Task

The AI_Task conformance class tests that the task objects are encoded according to the requirements.

CONFORMANCE CLASS A.7: AI_TASK

IDENTIFIER <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/aitask>

REQUIREMENTS CLASS /req/aitask

PREREQUISITE Conformance class A.1: <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/xmlbasetype>

CONFORMANCE TESTS Abstract test A.20: /conf/aitask/task
Abstract test A.21: /conf/aitask/eotask

ABSTRACT TEST A.20

IDENTIFIER /conf/aitask/task

REQUIREMENT /req/aitask/task

TEST PURPOSE Verify that instance documents using the AI_Task XML elements listed in Table 7 validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_task.xsd.

ABSTRACT TEST A.20

Validate the instance document against the ai_task.xsd schema to verify the above requirement.	
TEST METHOD	The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.21

IDENTIFIER /conf/aitask/eotask

REQUIREMENT /req/aitask/eotask

TEST PURPOSE Verify that instance documents using the AI_EOTask XML elements listed in Table 7 and Table 8 validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_eoTask.xsd.

TEST METHOD Validate the instance document against the ai_eoTask.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

A.6. Conformance Class: AI_Label

The AI_Label conformance class tests that the label objects are encoded according to the requirements.

CONFORMANCE CLASS A.8: AI_LABEL

IDENTIFIER <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/ailabel>

REQUIREMENTS CLASS /req/ailabel

PREREQUISITES Conformance class A.1: <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/xmlbasetype>
Conformance class A.4: <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/geospatialtype>

CONFORMANCE TESTS Abstract test A.22: /conf/ailabel/label
Abstract test A.23: /conf/ailabel/scenelabel
Abstract test A.24: /conf/ailabel/objectlabel
Abstract test A.25: /conf/ailabel/pixellabel
Abstract test A.26: /conf/ailabel/imageformatcode

ABSTRACT TEST A.22

IDENTIFIER /conf/ai/label/label

REQUIREMENT /req/ai/label/label

TEST PURPOSE Verify that instance documents using the AI_Label XML elements listed in Table 9 validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_label.xsd.

TEST METHOD Validate the instance document against the ai_label.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.23

IDENTIFIER /conf/ai/scene/sceneLabel

REQUIREMENT /req/ai/scene/sceneLabel

TEST PURPOSE Verify that instance documents using the AI_SceneLabel XML elements listed in Table 10 validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_sceneLabel.xsd.

TEST METHOD Validate the instance document against the ai_sceneLabel.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.24

IDENTIFIER /conf/ai/object/objectLabel

REQUIREMENT /req/ai/object/objectLabel

TEST PURPOSE Verify that instance documents using the AI_ObjectLabel XML elements listed in Table 11 validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_objectLabel.xsd.

TEST METHOD Validate the instance document against the ai_objectLabel.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.25

IDENTIFIER /conf/ailabel/pixellabel

REQUIREMENT /req/ailabel/pixellabel

TEST PURPOSE Verify that instance documents using the AI_PixelLabel XML elements listed in Table 12 validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_pixelLabel.xsd.

TEST METHOD Validate the instance document against the ai_pixelLabel.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.26

IDENTIFIER /conf/ailabel/imageformatcode

REQUIREMENT /req/ailabel/imageformatcode

TEST PURPOSE Verify that instance documents using the AI_ImageFormatCode XML elements conform to the requirements specified by the Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types [RFC 2046](https://tools.ietf.org/html/rfc2046).

TEST METHOD Inspect the instance document to verify the above requirement.

A.7. Conformance Class: AI_Labeling

The AI_Labeling conformance class tests that the labeling objects are encoded according to the requirements.

CONFORMANCE CLASS A.9: AI_LABELING

IDENTIFIER <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/ailabeling>

REQUIREMENTS CLASS /req/ailabeling

PREREQUISITES Conformance class A.1: <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/xmlbasetype>
Conformance class A.2: <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/isometadatatype>

CONFORMANCE CLASS A.9: AI_LABELING

CONFORMANCE TESTS	Abstract test A.27: /conf/ailabeling/labeling
	Abstract test A.28: /conf/ailabeling/labeler
	Abstract test A.29: /conf/ailabeling/labelingprocedure
	Abstract test A.30: /conf/ailabeling/labelingmethodcode

ABSTRACT TEST A.27

IDENTIFIER /conf/ailabeling/labeling

REQUIREMENT /req/ailabeling/labeling

TEST PURPOSE Verify that instance documents using the AI_Labeling XML elements listed in Table 13 validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_labeling.xsd.

TEST METHOD Validate the instance document against the ai_labeling.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.28

IDENTIFIER /conf/ailabeling/labeler

REQUIREMENT /req/ailabeling/labeler

TEST PURPOSE Verify that instance documents using the AI_Labeler XML elements listed in Table 14 validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_labeler.xsd.

TEST METHOD Validate the instance document against the ai_labeler.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

ABSTRACT TEST A.29

IDENTIFIER /conf/ailabeling/labelingprocedure

REQUIREMENT /req/ailabeling/labelingprocedure

TEST PURPOSE Verify that instance documents using the AI_LabelingProcedure XML elements listed in Table 15 validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_labelingProcedure.xsd.

ABSTRACT TEST A.29

TEST METHOD	Validate the instance document against the ai_labelingProcedure.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.
--------------------	--

ABSTRACT TEST A.30

IDENTIFIER /conf/ailabeling/labelingmethodcode

REQUIREMENT /req/ailabeling/labelingmethodcode

TEST PURPOSE Verify that instance documents using the AI_LabelingMethodCode XML elements validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_labelingMethodCode.xsd.

TEST METHOD Validate the instance document against the ai_labelingMethodCode.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

A.8. Conformance Class: AI_DataQuality

The AI_DataQuality conformance class tests that the data quality objects are encoded according to the requirements.

CONFORMANCE CLASS A.10: AI_DATAQUALITY

IDENTIFIER <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/aidataquality>

REQUIREMENTS CLASS /req/aidataquality

PREREQUISITES Conformance class A.1: <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/xmlbasetype>
Conformance class A.2: <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/isometadatatype>
Conformance class A.3: <http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/isoqualitytype>

CONFORMANCE TEST Abstract test A.31: /conf/aidataquality/classbalancedegree

ABSTRACT TEST A.31

IDENTIFIER	/conf/aidataquality/classbalancedegree
REQUIREMENT	/req/aidataquality/classbalancedegree
TEST PURPOSE	Verify that instance documents using the AI_ClassBalanceDegree XML elements listed in Table 16 validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_classBalanceDegree.xsd .
TEST METHOD	Validate the instance document against the ai_classBalanceDegree.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

A.9. Conformance Class: AI_TDChangeset

The AI_TDChangeset conformance class tests that the TD changeset objects are encoded according to the requirements.

CONFORMANCE CLASS A.11: AI_TDCHANGESET

IDENTIFIER	http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/aitdchangeset
REQUIREMENTS CLASS	/req/aitdchangeset
PREREQUISITES	Conformance class A.1: http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/base/xmlbasetype Conformance class A.6: http://www.opengis.net/spec/TrainingDML-AI-3/1.0/conf/aitrainingdata
CONFORMANCE TEST	Abstract test A.32: /conf/aitdchangeset/tdchangeset

ABSTRACT TEST A.32

IDENTIFIER	/conf/aitdchangeset/tdchangeset
REQUIREMENT	/req/aitdchangeset/tdchangeset
TEST PURPOSE	Verify that instance documents using the AI_TDChangeset XML elements listed in Table 17 validate against the XML schema specified in http://schemas.opengis.net/trainingdml-ai/part3/1.0/ai_tdChangeset.xsd .

ABSTRACT TEST A.32

TEST METHOD Validate the instance document against the ai_tdChangeset.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.



ANNEX B (INFORMATIVE) EXAMPLE



ANNEX B (INFORMATIVE) EXAMPLE

B.1. TrainingDataset Encoding Examples

B.1.1. WHU-RS19 Dataset

The WHU-RS19 dataset is widely used in scene classification of remote sensing images. This dataset is collected from Google Earth and has 19 classes including airport, beach, bridge, commercial, desert, farmland, football field, forest, industrial, meadow, mountain, park, parking, pond, port, railway station, residential, river, and viaduct. Each class contains around 50 images, with an image size of 600×600 and a resolution of 0.5 m.

An example of XML encoding of the WHU-RS19 dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/WHU-RS19.xml.

B.1.2. DOTA-v1.5 Dataset

The DOTA-v1.5 dataset is a large-scale dataset for object detection in aerial images. The sources for content in the dataset include Google Earth, Gaofen-2, and Jilin-1 imagery provided by China Resources Satellite Data Center. The 16 classes in DOTA-v1.5 are plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, large vehicle, small vehicle, helicopter, roundabout, soccer ball field, swimming pool, and container crane. Compared with other aerial image object detection datasets, the dataset has the largest number of classes. The images in the dataset have various image sizes (from 800×800 to 2000×2000) and resolutions (Google Earth/0.1 m-1 m, Gaofen-2/1 m, Jilin-1/0.72 m).

An example of XML encoding of the DOTA-v1.5 dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/DOTA-v1.5.xml.

B.1.3. KITTI 2D Object Detection Dataset

The KITTI 2D object detection dataset is a novel open-access dataset and benchmark for road area and ego-lane detection. KITTI 2D consists of 7481 annotated training images of high variability from the KITTI autonomous driving platform by two PointGrey Flea2 color cameras,

capturing a broad spectrum of urban street views and road scenes. The eight (8) classes in the KITTI 2D object detection dataset are car, van, truck, pedestrian, person_sitting, cyclist, tram, and misc. Compared with other street view object detection datasets, this dataset compresses diverse scenarios and captures real-world traffic situations, ranging from freeways over rural areas to inner-city scenes with many static and dynamic objects.

An example of XML encoding of the KITTI 2D object detection dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/KITTI.xml.

B.1.4. GID Dataset

The GID dataset is one of state-of-art land cover classification datasets. This dataset has a large spatial coverage covering many provinces in China with a relatively high spatial resolution (2 m). GID has two sets. One is the GID-5C. It has 150 images (image size 7200×6800) that are classified into 5 land cover classes. The other set is GID-15C. The images from GID-5C are sliced into 30,000 patches in GID-15C, which have three types of patch sizes (56×56, 112×112, 224×224) and are classified into 15 land cover classes.

An example of XML encoding of the GID-5C dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/GID-5C.xml.

B.1.5. Toronto3D Dataset

The Toronto3D dataset is a large urban outdoor point cloud dataset for segmentation collected by the Mobile Laser Scanning System. The dataset covers about 1 km of scene streets in Toronto, including four areas named L001, L002, L003, and L004, with a total of 78.3 million points. Each point in this dataset has 10 attributes representing the 3D position, RGB color, intensity, GPS time, scan angle rank, and category, respectively. This dataset has eight categories, including road, road mark, natural, building, utility line, pole, car, and fence.

An example of XML encoding of the Toronto3D dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/Toronto_3D.xml.

B.1.6. WHU-Building Dataset

The WHU-Building dataset is a change detection dataset collected from the Land Information New Zealand Data Service. The dataset is composed of images (with the resolution 0.2 m) in 2012 and 2016, covering 20.5 km². It includes 12,796 and 16,077 buildings respectively in 2012 and 2016.

An example of XML encoding of the WHU-Building dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/WHU-building.xml.

B.1.7. California Change Detection Dataset

The California Change Detection Dataset is composed of two images and a label image. The first image is a Landsat 8 acquisition covering Sacramento County, Yuba County and Sutter County, California, on 5 January 2017. It has nine channels covering the spectrum from deep blue to short-wave infrared, plus two long-wave infrared channels. The second image was acquired on 18 February 2017 by Sentinel-1A over the same area after the occurrence of a flood. The image is recorded in polarizations VV and VH and augmented with the ratio between the two intensities as a third channel. All these channels are log-transformed.

An example of XML encoding of the California change detection dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/UiT_HCD_California_2017.xml.

B.1.8. WHU MVS Dataset

The WHU MVS dataset is a synthetic aerial dataset created for large-scale and high-resolution Earth surface reconstruction. The basic training sample of the dataset is a multi-view unit consisting of five aerial images, and their corresponding depth maps are taken as ground truth. There are a total of 5680 pairs of five-view aerial images in the dataset. All the images are simulated from a 3D surface model, which is produced by Smart3D software using Unmanned Aerial Vehicle (UAV) images and refined by manual editing.

An example of XML encoding of the WHU MVS dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/WHU_MVS.xml.

B.1.9. iSAID Dataset

The iSAID dataset is the first benchmark dataset for instance segmentation in aerial images. This large-scale and densely annotated dataset contains 655,451 object instances for 15 categories across 2,806 high-resolution images. The images of iSAID is the same as the DOTA-v1.0 dataset, which are mainly collected from the Google Earth, some are taken by satellite JL-1, the others are taken by satellite GF-2 of the China Centre for Resources Satellite Data and Application. The object categories in iSAID include: plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, large vehicle, small vehicle, helicopter, roundabout, soccer ball field and swimming pool.

An example of XML encoding of the iSAID dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/blob/main/use-cases/examples/1.0/iSAID.xml.

B.2. DataQuality Encoding Example

B.2.1. WHU-RS19 Data Quality

An encoded data quality example of the WHU-RS19 datasets following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/WHU-RS19-quality.xml.

B.3. TDChangeset Encoding Example

B.3.1. DOTA-v1.5 Changeset

DOTA-v1.5 uses the same images as DOTA-v1.0, but the extremely small instances (less than 10 pixels) are also annotated. Moreover, a new category “container crane” is added. It contains 403,318 instances in total. The number of images and dataset splits are the same as DOTA-v1.0. This version was released for the DOAI Challenge 2019 on Object Detection in Aerial Images in conjunction with IEEE CVPR 2019.

An encoded changeset example between the DOTA-v1.0 and DOTA-v1.5 datasets following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/DOTA-v1.5-changeset.xml.

B.4. Non-EO Imagery TrainingDataset Encoding Examples

B.4.1. ERA5 Dataset

The ERA5 dataset is derived from in-situ observational data (Copernicus product), and we limit its usage scenario to the autoregression problem of time series data. Therefore, its label is the data itself. Similar to unsupervised learning, the autoregression task for time series data does not require additional labeled data. For this dataset, inheritance classes for AI_AbstractLabel are not defined, although this class is required in the existing standard (please note that these test cases are for future versions of the standard). In addition, additional attributes to support the complete representation of dataset information were added.

An example of XML encoding of the ERA5 dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/blob/main/use-cases/examples/1.0/ERA5_hourly_data.xml.

B.4.2. SCIERC Dataset

The SCIERC dataset is derived from textual data, and its labels are the classification of the text. This dataset is a text classification problem, with the goal of information extraction and entity recognition. For this textual dataset, the Abstract class is inherited and `AI_TextTrainingDataset`, `AI_TextTrainingData`, `AI_TextTask`, and `AI_EntityLabel` respectively are defined. In addition, additional attributes to support the complete representation of dataset information were added.

An example of XML encoding of the SCIERC dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/blob/main/use-cases/examples/1.0/SCIRec.xml.

B.4.3. nuScenes Dataset

The nuScenes dataset is a public large-scale dataset for autonomous driving developed by the team at Motional (formerly nuTonomy). The full dataset includes approximately 1.4M camera images, 390k LIDAR sweeps, 1.4M RADAR sweeps and 1.4M object bounding boxes in 40k keyframes. Although the training data may come from different domains, the 3D annotation boxes captured by numerous sensors in the same keyframe are targeted at the same object and are unique. Based on this, a 3D annotation box is used to organize each 3D object using `AI_ObjectLabel`. Since each training data and each 3D object require many additional attributes to be fully described, many additional attributes to provide a detailed description of the training dataset, training data, labels, etc. were added.

An example of XML encoding of the nuScenes dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/blob/main/use-cases/examples/1.0/nuScenes.xml.



ANNEX C (INFORMATIVE) REVISION HISTORY



ANNEX C

(INFORMATIVE)

REVISION HISTORY

Table — Revision history

DATE	RELEASE	AUTHOR	PARAGRAPH MODIFIED	DESCRIPTION
2023-12-15	0.1	Peng Yue, Ruixiang Liu, Kaixuan Wang, Yongyi Mi	All	Draft for internal review.
2024-02-26	0.2	Peng Yue, Ruixiang Liu, Carl Reed, Yongyi Mi	Most	Merge edits and comments from Carl Reed.
2024-06-09	0.3	Peng Yue, Ruixiang Liu, Yongyi Mi	Chapter 2, 4, Annex A	Revisions after OAB review and public comments.
2026-02-28	1.0.1	Peng Yue, Baoxin Teng	Most	Revisions based on the DIS comments on ISO 19178-1.



BIBLIOGRAPHY



BIBLIOGRAPHY

- [1] How Our Compliance Program Works, n.d. Open Geospatial Consortium [online]. Available from: <https://www.ogc.org/how-our-compliance-program-works/> (accessed 11.3.24).