2026 European Conference on Computing in Construction
Corfu, Greece
July 12-15, 2026

PRESENTING THE LOINxml SCHEMA: DISCUSSING DESIGN DECISIONS AND INTENDED APPLICATION

Martina Mellenthin Filardo[1,2], Paul Debus[1], Sven Zentgraf[3], Štefan Jaud[4], Christian Clemen[5], Ghang Lee[6], Jakob Beetz[7] and André Borrmann[8]
[1]Bauhaus-Universität Weimar, Weimar, Germany
[2]Erfurt University of Applied Sciences, Erfurt, Germany
[3]Ruhr-Universität Bochum, Bochum, Germany
[4]Jaud IT GmbH, Moorenweis, Germany
[5]Universtiy of Applied Sciences Dresden, Dresden, Germany
[6]Yonsei University, Seoul, South Korea
[7]RWTH Aachen University, Aachen, Germany
[8]Technical University of Munich, Munich, Germany

Abstract

The Level of Information Need introduced in ISO 19650 gained a standardized machine-readable data model with the LOINxml (ISO 7817-3). This paper presents the normative context of this standard with a short outline of the specification process of the LOINxml data model. Through systematic analysis of schema elements and design decisions, this work demonstrates how LOINxml enables structured, transaction-based requirement specifications supporting alphanumerical, documentation, and geometrical information across project lifecycles. The contribution establishes LOINxml as foundational infrastructure for digital information requirements while identifying threshold conditions and future research directions essential for industry implementation.

Introduction

Precise information requirements are critical for Building Information Modeling (BIM) information delivery and conformant project deliverables in the architecture, engineering, construction, and operation (AECO) industry (Sacks et al., 2025). Machine-readable representation of these requirements enables not only vendor-neutral definition, but also more precise requirements and higher quality of project deliverables. A framework for such requirement definition has been established in the Level of Information Need (LOIN) standards. While fundamentals have been specified in part 1 of ISO 7817, application guidelines are presented in part 2 (ISO/TS 7817-2, 2025), and a corresponding data model is defined in part 3. (ISO 7817-3, 2025). Of these three interrelated standards, this paper discusses the design decisions and intended applications of the ISO 7817-3 LOINxml schema.

This paper documents the context and design decisions of the data model presented in ISO 7817-3 (2025) to promote practitioner adoption. The context in which the data model is embedded is explained in the normative setting. Subsequently, the development process is explained. The data model published in ISO 7817-3 is then presented and discussed. The paper finishes with the conclusions and outlook.

Normative setting

The Level of Information Need (LOIN) concept was introduced in ISO 19650-1 (2018). However, the concept had to be elaborated into precise normative specifications, which has been done in the ISO 7817 series, where fundamentals and principles are described in ISO 7817-1 (2024), guidelines and application in ISO/TS 7817-2 (2025), and an XML-based data model in ISO 7817-3 (2025). ISO 7817-3, specifying a schema for the Level of Information Need, aligns closely with ISO 23387 (2025), which establishes a data template structure for construction objects based on the methodology for describing, creating, and maintaining properties in interconnected data dictionaries (ISO 23386, 2020) described in ISO standard 12006-3. This approach provides the concepts and relationships required to describe information about building objects (ISO 23387, 2025).

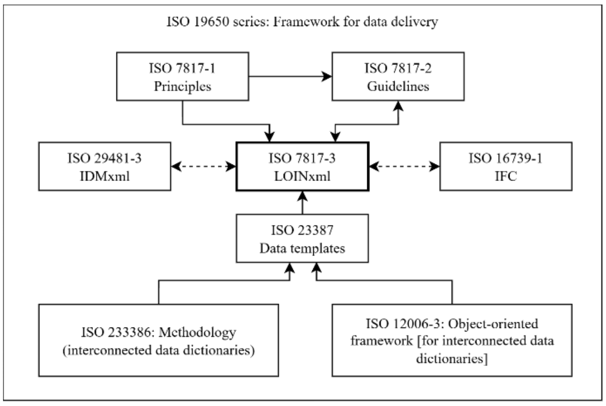

Figure 1: Normative context of ISO 7817-3.

In this context, a data template is a subset of the model ISO 23387 extends the ISO 23386 methodology, emphasizing template reusability and both standards harmonize with ISO 12006-3 (2023), thus enabling an object-oriented approach – a strong argument for adoption of the data templates elements in the LOINxml data model. This approach was implemented by importing

XML schema elements provided by ISO 23387 into the LOINxml schema, which represents the strongest form of integration. ISO 7817-3 coexists with ISO 29481-3 (2022) on the Information Delivery Manual (IDM). While both standards provide machine-readable XML structures for information requirements, IDM focuses on information requirements exchanged within specific use cases, whereas LOIN focuses on the data model for the level of information need. While being agnostic to specific data models for actual model contents, the LOINxml is compatible with the Industry Foundation Classes (IFC), among other formats and breakdown structures.

A strong principle implemented by LOINxml is the use of references to information provided in other data schema and resources. This may include references to data dictionaries, such as bSDD, for specifying ObjectTypes and Properties, or references to IDMxml data for referencing project milestones. The references are expressed by International Resource Identifiers (IRI) to allow easy lookup and retrieval, while providing a maximum of flexibility. The context of ISO standards relevant to the LOINxml data model are outlined in Figure 1. The LOINxml data schema can be used for file-based exchange (including multiple files referencing each other) but can also act as a blueprint for defining database schemas and providing access through REST (Representational State Transfer) APIs (Application Programming Interfaces).

Methodology

The development of the schema for ISO 7817-3 was conducted within the framework of standardization work (Vienna Agreement, 1991) for cooperation between the European Committee for Standardization (CEN) and the International Organization for Standardization (ISO) (European Committee for Standardization and International Organization for Standardization, 2016). Delegated experts from all interested countries participate in standard development. In the case of ISO 7817-3, experts came from Belgium, Canada, Czech Republic, France, Germany, Norway, South Korea, and the UK. Meetings were conducted virtually. Over the course of six years of development work, the meeting frequency varied between biweekly and approximately monthly (accounting for breaks and holidays). Figure 3 outlines the process. Other works aided in the development of this standard, such as Jeon et al. (2021); Jaud, Gregory and Borrmann (2021); Bolpagni et al. (2022); Abualdenien and Borrmann (2022); Son et al. (2022); Mellenthin Filardo et al. (2024a); Mellenthin Filardo et al. (2024b); Akbas et al. (2025); Zentgraf et al. (2025) among others.

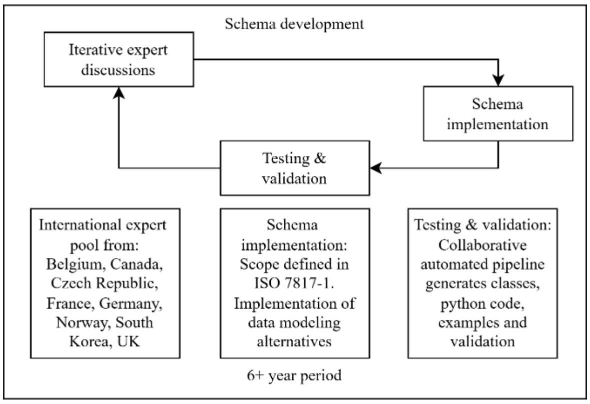

Figure 2: Overview of the schema development procedure

The schema scope derives from ISO 7817-1. Guidelines in ISO/TS 7817-2, developed parallel to ISO 7817-3, informed LOIN elaborations subsequently discussed, tested, and eventually incorporated into LOINxml. Substantive congruence between ISO 23387 and ISO 7817-3 necessitated parallel mechanism implementation. Collaboration through integration avoided duplicate work and contradictory approaches. Individual elements and mechanisms were iteratively discussed among experts.

Discussion material alternated between UML diagrams, XSD files, and XML examples using a collaborative repository. Automated actions generated classes from XSD files using xsData libraries. Python scripts produced and validated XML examples against the schema. XSD and example generation proceeded collaboratively through expert-proposed improvements across multiple iterations. Both XSD and XML files underwent expert review. Drafts circulated through designated internal committees until the standard entered its final phase.

Presenting the LOINxml

The root element of the LOINxml data model is the *LevelOfInformationNeed* element, which acts as a container for the *Specification* element(s), as shown in Figure 2. The *Specification* aggregates the elements *Prerequisites* (mandatory), for context information, *GeoReferencing* (optional), for coordinate system information, and *SpecificationPerObjectType* (optional, zero to many), for specifying requirements.

The *Prerequisites* element (Figure 4) establishes the transaction framework by specifying the purpose, information delivery milestone, and actors of the

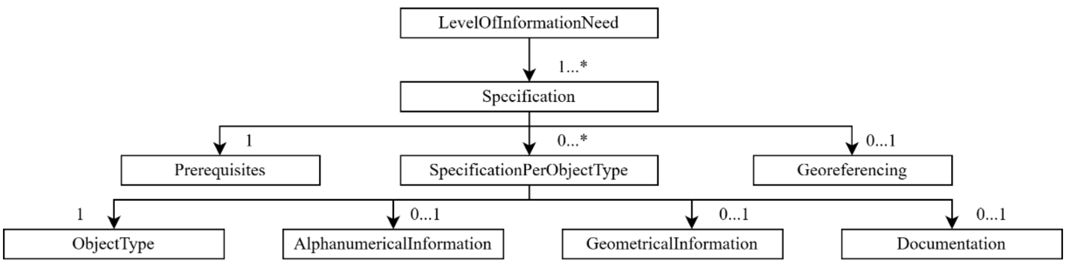

Figure 3: Simplified overview of the LOINxml data model. A detailed version of the schema can be found in ISO 7817-3.

LOINxml. The LOINxml is understood as a transaction limited to a receiving and a providing actor.

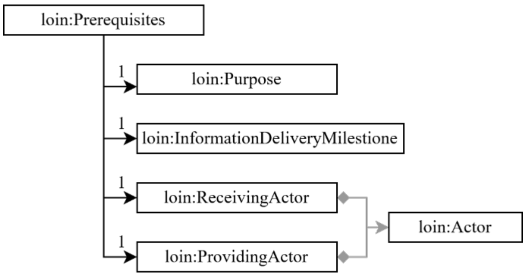

Figure 4: Structure of the Prerequisites element

The *Purpose* element provides the rationale for information need and includes name, optional definition, references, description, language, region, and dictionary links. The *InformationDeliveryMilestone* specifies delivery timing through name, optional description, references, and date/time. The *ProvidingActor* identifies the information provider through role, optional name, affiliation, description, and email. The *ReceivingActor* identifies the information recipient using the same structure as *ProvidingActor*. Varying actors or purposes require separate *Specification* instances rather than consolidated prerequisites. Multiple specifications with different prerequisites can coexist within one document.

The *GeoReferencing* element, shown in Figure 5, specifies the coordinate reference system and transformation parameters from model to coordinate reference system (CRS) coordinates. It comprises the elements *CoordinateReferenceSystem*, defining CRS type and datum, and *ModelCoordinateSystem*, establishing model origin and orientation within the CRS where multiple instances are possible.

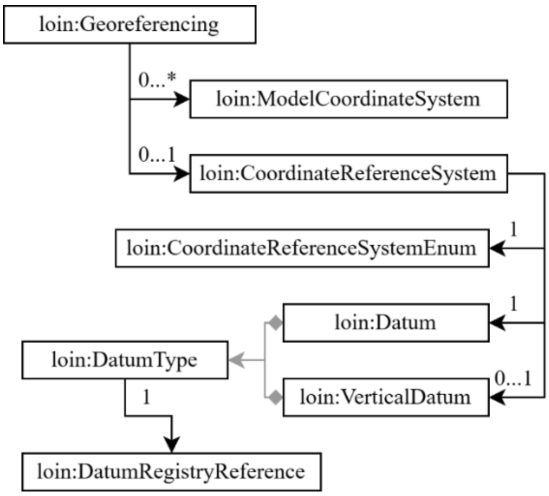

Figure 5: Georeferencing element structure

The *CoordinateReferenceSystem* defines the reference system through Type specifying CRS type as *ProjectedCRS*, *EngineeringCRS*, *GeographicCRS*, or *NotRequired, Datum* providing required horizontal datum, and *VerticalDatum* offering optional vertical datum. Each datum includes Name such as "EPSG:25832" or WKT2 string and Type as *DatumRegistryReference*. The *DatumRegistryReference* enables flexible geodetic registry specification per ISO 19127:2019 through *Name*, optional *Description*, and optional *RegistryReference* provided through a Unified Resource Identifier (URI).

The *ModelCoordinateSystem* element defines model positioning within CRS using generic coordinate names. *IsProjected* indicates whether the model inherits CRS projection distortion. *FirstCoordinate* specifies the first horizontal coordinate, typically Easting or Latitude. *SecondCoordinate* defines the second horizontal coordinate, typically Northing or Longitude. *Height* establishes the vertical coordinate. *XAxisAbscissa* and *XAxisOrdinate* provide optional model X-axis direction components. *UnitScale* offers optional model-to-CRS unit conversion. *HorizontalScale* provides optional horizontal distance scale factor.

Avoiding to “reinvent the wheel”, the LOINxml schema uses several elements from the data templates schema (namespace *dt*) defined in ISO 23387 (2025), in particular when it comes to the definition of properties, data types, and units. The schema employs Globally Unique Identifiers (GUIDs) and *ReferenceType* for element referencing within documents and across external resources. The schema definition of the used reference mechanisms is shown in Figure 6.

```
<xs:complexType name="ReferenceType">
    <xs:attribute ref="referenceURI" use="optional" />
    <xs:attribute ref="GUID" use="optional" />
</xs:complexType>
<xs:attribute name="referenceURI" type="xs:anyURI">
    <xs:annotation>
        <xs:documentation>
            Defines a referenceURI to identify a concept
            using URI
        </xs:documentation>
    </xs:annotation>
</xs:attribute>
<xs:attribute name="GUID" type="GuidType">
    <xs:annotation>
        <xs:documentation>
            Defines the GUID of a concept
        </xs:documentation>
    </xs:annotation>
</xs:attribute>
<xs:simpleType name="GuidType">
    <xs:annotation>
        <xs:documentation>
            The representation of a GUID, generally the id
            of an element.
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:pattern value="[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-
                           [a-fA-F0-9]{4}-[afA-
                           F0-9]{4}-[a-fA-F0-9]{12}" />
    </xs:restriction>
</xs:simpleType>
```

Figure 6: Schema definition of the elements from ISO 23387 used for linking in ISO 7817-3.

GUIDs provide unique element identification serving two purposes: establishing element identity and enabling internal cross-references. Most elements support the *dt:GUID* attribute for same-document referencing.

The *dt:ReferenceType* creates references via two attributes. The *dt:GUID* attribute references elements within the same document. The *dt:referenceURI* attribute references external resources. When both attributes are present, the reference points to a specific element identified by GUID within an external resource identified by URI. Referencing guidelines recommend using *dt:GUID* alone for same-file references, *dt:referenceURI* alone for external resources, and both attributes for specific elements within external resources.

The *SpecificationPerObjectType* element defines information requirements for specific object types, such as walls, doors, or bridge abutments, binding object definitions to three requirement aspects: alphanumerical information, documentation, and geometrical information. Each Specification may contain multiple *SpecificationPerObjectType* elements, with one per deliverable object type. The structure comprises *ObjectType* defining the target object type as a required element, *AlphanumericalInformation* specifying required properties, quantities as optional, *Documentation* defining required documents as optional, and *GeometricalInformation* specifying geometrical requirements as optional.

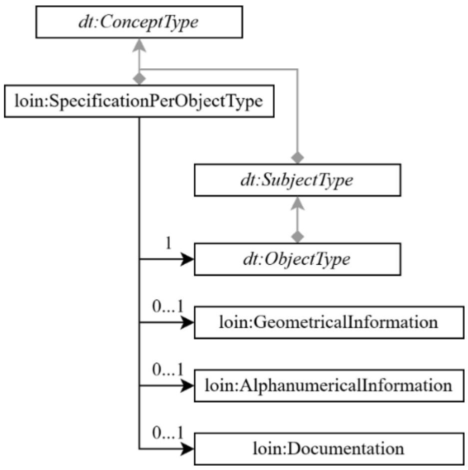

Figure 7: Structure of the SpecificationPerObjectType element

Since *loin:SpecificationPerObjectType* extends *dt:ConceptType* from ISO 23387, it includes Name and Definition elements separate from the *ObjectType* it addresses. Figure 7 shows the *dt:ConceptType* element and how it is incorporated in the LOINxml.

The *ObjectType* element identifies requirement targets using *dt:ObjectTypeType* from ISO 23387. Each object type requires a *dt:GUID* for internal referencing. Two main definition approaches exist: inline definition with Name, Definition, and metadata, or external reference via *dt:ReferenceDocumentRef* linking to libraries or other external resources. Multiple language variants support international projects as specified in ISO 23387. Version tracking occurs through *dt:MajorVersion*, *dt:MinorVersion*, and *dt:Status* as defined in ISO 23387. External references enable linkage to established classifications such as IFC or UniClass while maintaining internal GUIDs and optional local metadata.

A reduced instance example of a specification per object type element with emphasis on an object type definition using CityGML is shown in Figure 8. The *dt:ObjectType* is a child of the *dt:SubjectType* element, thus covering all the metadata of the *dt:ConceptType* element and allowing references defined in the subject type (reference for '*hasParts*' and '*isSubtypeOf*'). By aggregating specific information and associating it with a specific object type, the specification per object type container acts as a 'has' relation inherently (XML does not support relations in that sense).

```xml
<SpecificationPerObjectType dt:GUID="2c34312at-6h81-4907-
827fdc5eb9b2f28d" dateOfCreation="2026-01-07T10:30:00">
    <dt:Name language="en">CityGML abstract relief component
        properties</dt:Name>
    <dt:Definition language="en">CityGML abstract relief
        component to aggregate inheritance properties for terrain
        elements</dt:Definition>
    <ObjectType dt:about="https://schemas.opengis.net/citygml/relief/
1.0/relief.xsd" dt:GUID="ab294g80-2c6c-6f13-2adf-
46ed115369e9" dateOfCreation="2026-01-07T10:30:00">
        <dt:Name language="en">CityGML abstract relief
            component</dt:Name>
        <dt:Definition language="en">CityGML abstractrelief
            component</dt:Definition>
        <dt:Description language="en">abstract component to
            aggregate inheritance attributes</dt:Description>
        <dt:LanguageOfCreator>en-US</dt:LanguageOfCreator>
        <dt:CountryOfOrigin>Germany</dt:CountryOfOrigin>
        <dt:MajorVersion>3</dt:MajorVersion>
        <dt:MinorVersion>2</dt:MinorVersion>
        <dt:Status>Released</dt:Status>
    </ObjectType>
</SpecificationPerObjectType>
```

Figure 8: XML instance example of an object type definition for a CityGML element

The *AlphanumericalInformation* element specifies required properties through several components. Its structure is presented in Figure 9.

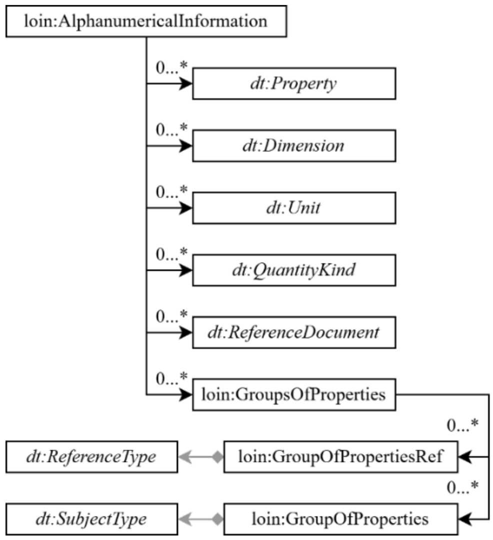

Figure 9: Structure of the AlphanumericalInformation element

Property defines individual property requirements either inline or referenced, with data types, constraints, and measurement references. *QuantityKind* provides quantity type definitions such as Length, Mass, or Pressure for inline properties or by referencing *Dimension*. *Dimension* expresses physical dimensionality (i.e., $L^1M^0T^0$) as SI base unit exponents, supporting inline units and quantity kinds. *Unit* specifies measurement units with conversion information including Scale, Base, Coefficient, and Offset. *DataType* specifying constraints. External definitions via *dt:referenceURI* on reference elements can point to standards like the Quantity, Unit, Dimension, and Type (QUDT) ontology (Hodgson, 2024).

Figure 10 shows an XML example of a self-contained LOINxml definition (inline in one file) while Figure 11 demonstrates the definition in separate files (e.g., one LOINxml and a separate Library with referenceable property definitions).

```xml
<AlphanumericalInformation dt:GUID="94eb4cfa-46a4-4ea6-
    bab5-
    c6711b1ae35a">
    <Property dt:GUID="8ef0be47-929f-428a-9016-
      ec1e1f1f7616" dateOfCreation="2026-01-07T10:30:00">
        <dt:Name language="en">relativeToTerrain</dt:Name>
        <dt:Definition language="en">Property to filter
               elements without
            checking geometry</dt:Definition>
        <dt:DataType name="STRING">
            <dt:PossibleValues>
                <dt:ValueList language="en">
                    <dt:Value order="1">
                          entirelyAboveTerrain </dt:Value>
                    <dt:Value order="2">
                          substantiallyBelowTerrain
                    </dt:Value>
                </dt:ValueList>
            </dt:PossibleValues>
        </dt:DataType>
    </Property>
    <dt:Property dt:GUID="f1090232-9199-22ab-23gh-
      4a88b6d6b86d" dateOfCreation= "2026-01-07T10:30:00">
        <dt:Name language="en"> FrostResistanceClasses
        </dt:Name>
        <dt:Name language="de">Frostschutzklassen</dt:Name>
        <dt:Definition language="en">Exposure classes
            related to freezethaw
            cycles according to DIN 1054</dt:Definition>
        <dt:LanguageOfCreator>en-US</dt:LanguageOfCreator>
        <dt:DataType name="STRING">
            <dt:PossibleValues>
                <dt:ValueList language="en">
                    <dt:Value order="1">XF1</dt:Value>
                    <dt:Value order="2">XF2</dt:Value>
                    <dt:Value order="3">XF3</dt:Value>
            </dt:PossibleValues>
        </dt:DataType>
    </dt:Property>
</AlphanumericalInformation>
```

Figure 10: XML instance example of an inline (self-contained) property definition for the specified CityGML object type

```xml
  <AlphanumericalInformation dt:GUID="82cd6dfc-24a7-1ed9-cda8-
a6727a1be28a">
    <GroupOfProperties dt:GUID="b2120e0b-47b1-4f66-b0f7-70a9e0c559ce" d
        ateOfCreation="2026-01-07T10:30:00">
        <dt:Name language="en">Structural Analysis Properties</dt:Name>
        <dt:Definition language="en">Collection of exemplary properties
            <dt:LanguageOfCreator>en-US</dt:LanguageOfCreator>
            <dt:HasPropertyRef
        dt:referenceURI="file:structural_properties_
        library.xml" dt:GUID="55492688-2348-4986-85b3-
        b1a3fb68b31b" />
            <dt:HasPropertyRef
        dt:referenceURI="file:structural_properties_
        library.xml" dt:GUID="f9660631-7197-41cc-95bc-
        8a98a5d5b89c" />
    </GroupOfProperties>
    </GroupsOfProperties>
</AlphanumericalInformation>
and

<dt:Property dt:about="https://ifc43-
docs.standards.buildingsmart.org/IFC/
RELEASE/IFC4x3/HTML/lexical/IfcPropertySingleValue.htm"
dt:GUID="55492688-2348-4986-85b3-b1a3fb68b31b" dateOfCreation="2024-01-
10T09:00:00">
    <dt:Name language="en">LoadBearingCapacity</dt:Name>
    <dt:Name language="de">Tragfähigkeit</dt:Name>
    <dt:Definition language="en">Maximum axial load the column can
        support
        under design conditions</dt:Definition>
    <dt:Description language="en">Ultimate load capacity considering
        safety
        factors and material properties</dt:Description>
    <dt:LanguageOfCreator>en-US</dt:LanguageOfCreator>
    <dt:CountryOfOrigin>Germany</dt:CountryOfOrigin>
    <dt:DataType name="REAL">
        <dt:MinInclusive value="100.0" />
        <dt:MaxInclusive value="50000.0" />
        <dt:DataFormat value="[0-9]+\.[0-9]{1,2}" />
    </dt:DataType>
    <dt:Symbol>F_Rd</dt:Symbol>
    <dt:DimensionRef dt:GUID="63f6299d-6ef4-4371-947f-4d3cef25fcdb" />
    <dt:UnitRef dt:GUID="e3bf386d-6baa-473e-aa8b-d95b20dace1b" />
    <dt:QuantityKindRef dt:GUID="43daf97c-ee29-4800-b010-
        c5f2ab4f05be" />
</dt:Property>
```

Figure 11: XML instance example outlining the linking mechanism (yellow highlight) of properties in separate files

GroupsOfProperties contains collections of related properties through inline *GroupOfProperties* or external *GroupOfPropertiesRef* elements. *ReferenceDocument* provides optional links to external dictionaries or standards.

The *GeometricalInformation* element defines geometrical requirements through five aspects, as shown in Figure 12.

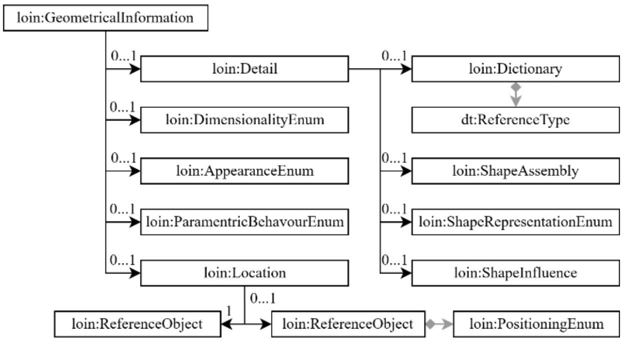

Figure 12: Structure of the GeometricalInformation element

The *Detail* aspect specifies geometry detailing information through a *Dictionary*, providing optional external definition references, *ShapeAssembly*, describing object assembly per ISO/TS 7817-2, *ShapeRepresentation* defining shape representation method per ISO/TS 7817-2, and *ShapeInfluence* establishing geometric feature requirements per ISO/TS 7817-2. The *Dimensionality* aspect describes dimensional representation as *0D, 1D, 2D, 3D*, or *NotRequired* per ISO/TS 7817-2.

The *Appearance* aspect specifies visual appearance requirements per ISO/TS 7817-2. The *ParametricBehaviour* aspect defines parametric behavior specification per ISO/TS 7817-2. The *Location* aspect establishes positioning method as *Absolute, Relative*, or *NotDefined* per ISO/TS 7817-2, where *RelativeOrAbsolute* specifies the positioning method as required and *ReferenceObject* provides the reference for relative positioning through GUID, filter, XML path, or natural language as optional.

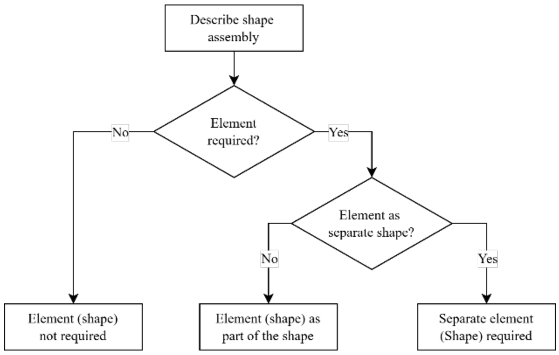

Figure 13: Simplified decision tree showing shape composition

The *Placeholder* attribute indicates placeholder-only geometry. *ShapeInfluence* specifies geometric feature representation with enumerated values according to the decision trees presented in ISO/TS 7817-2. A simplified version of the decision tree logic is outlined in Figure 13. The resulting enumerations (*NotRequired, No[Element], [Element]AsPartOfShape,* or *SeparateShapes*), and optional *ThresholdDimension* for size thresholds.

An XML example is given in Figure 14 to show application of the geometrical information.

```xml
<GeometricalInformation dt:GUID="5a807595-d9a9-41ce-868d-
3912fa16b19c" placeholder="false">
    <Detail>
```

```
        <Dictionary dt:referenceURI=
      "https://www.buildingsmart.co.at/glossarentries/
      level-of-detail-200-lod-200/" />
        <ShapeAssembly> SingleObjectSingularShape
        </ShapeAssembly>
        <ShapeRepresentation> SingleBoundingPrimitive
                </Shape Representation>
            <ShapeInfluence>
                <InsideGeometry>
      InsideGeometryAsPartOfShape </Inside Geometry>
                    <Connections> ConnectionsAsPartOfShape
                    </Connections>
                    <Openings>SeparateShapes</Openings>
                    <OperatingAndClearanceZones>
      SeparateShapes </Operating AndClearanceZones>

      <Features>FeaturesAsPartOfShape</Features>
            </ShapeInfluence>
    </Detail>
    <Dimensionality>3D</Dimensionality>
    <Appearance>SingularMaterial</Appearance>
    <ParametricBehaviour>Requested</ParametricBehaviour>
    <Location>
        <RelativeOrAbsolute>Relative</RelativeOrAbsolute>
        <ReferenceObject>Relative to the bridge
            bearing</ReferenceObject>
    </Location>
</GeometricalInformation>
```

Figure 14: XML instance example showing a geometrical information definition

The *Documentation* element specifies required documents via one or more *Document* elements. Each *Document* defines name as the document identifier, description specifying content as optional, *ReferenceDocument* providing external specification references as optional, *Format* establishing required file format with name and version, and optional classification attributes including type, form, and content.

Discussion of design decisions

The LOINxml data model prioritized reusing established principles and respecting existing frameworks. A major consideration was the breakdown structure foundation.

While IFC is widely established in AECO industry, not all requirements can be defined within the IFC data schema due to receiving actors' limited IFC knowledge or requirement nature. Projects requiring alternative breakdown structures are not uncommon. To enable broad application, the ISO 7817-3 (2025) data model remains agnostic, permitting any breakdown structure. The design supports multiple breakdown structures, enabling requirements spanning various disciplines. Though this decision introduces known challenges, the primary objective was allowing requirement definition as receiving actors deem necessary. Translating formalized requirements into IFC or other formats constitutes a subsequent step.

LOIN specifications require prerequisites elements while georeferencing and specification per object type remain optional. This ensures LOINxml specifications always include contextual information. The transaction-based model restricting specifications to one receiving and one providing actor was deliberate. Complex actor relationships such as sub-contractors or multiple providing actors were considered. However, such transactions vary either in purpose, warranting separate specifications, or in requirements themselves as with sub-contractors or separate providing actors delivering partial models. Given LOINxml's extensive linking options for alphanumerical information and flexible property structuring, the two-actor solution balances clearly framed requirements with lightweight definition.

The *GeoReferencing* structure separates CRS definition from model positioning through distinct *CoordinateReferenceSystem* and *ModelCoordinateSystem* elements. This separation acknowledges that multiple model coordinate systems may exist within a single CRS, supporting scenarios where different project components use different local origins or orientations within a shared global reference frame. The use of generic coordinate names (*FirstCoordinate, SecondCoordinate, Height*) rather than CRS-specific terminology (Easting, Northing, Latitude, Longitude) enables the schema to accommodate projected, geographic, and engineering coordinate systems uniformly. The *IsProjected* parameter addresses the fundamental distinction between models that inherit CRS projection distortion and those maintaining true-to-reality geometry, reflecting practical considerations in infrastructure versus building modeling workflows. The optional transformation parameters (*UnitScale, HorizontalScale, XAxisAbscissa, XAxisOrdinate*) provide flexibility for complex coordinate transformations but increase specification complexity and potential for misapplication.

The integration of multilingual capabilities through ISO 23387's *dt:ConceptType* enables specifications to function across language boundaries without translation or duplication. This design decision reflects the international nature of construction projects and standards development but introduces complexity in specification authorship and maintenance where multiple language variants must remain synchronized. The inclusion of versioning metadata (*MajorVersion, MinorVersion, Status*) at the object type level enables tracking of definition evolution over time, supporting change management and specification maturity assessment. However, the schema does not enforce versioning constraints or define version compatibility rules, leaving version management protocols to implementation conventions rather than schema-level enforcement.

The distinction between *ObjectType* definition and *SpecificationPerObjectType* implements a separation of concerns where object identity remains independent of requirement specification. This architecture permits the same object type to appear in multiple specifications with varying requirements while maintaining consistent identity through GUIDs. The extension of *dt:ConceptType* from ISO 23387 enables the specification itself to carry metadata distinct from the object it addresses. This dual-layer approach supports requirement versioning and modification without altering object type definitions, facilitating reuse across different project contexts and specification documents.

The schema accommodates both inline definition and external referencing across multiple element types, reflecting a fundamental tension between self-contained specifications and distributed definition management. Inline definitions create complete, autonomous documents requiring no external dependencies but introducing redundancy when similar definitions recur across specifications. External referencing through *dt:referenceURI* reduces redundancy and enables centralized maintenance of definitions but creates dependencies on external resources whose availability and versioning must be managed. The property definition chain exemplifies this design where inline properties require complete definition of supporting *Units, Dimensions,* and *QuantityKinds*, while externally referenced properties inherit these definitions from their source dictionaries. This flexibility accommodates varying organizational capabilities and project requirements but places responsibility for consistency management on specification authors.

The decomposition of requirements into alphanumerical, documentation, and geometrical aspects reflects established information modeling paradigms where object attributes, associated documents, and geometric representations constitute distinct information categories. This separation enables independent specification of each aspect's requirements, supporting scenarios where different aspects require different levels of detail (for lack of a better word) or where responsibility for different information types resides with different project participants. The optional nature of all three aspects acknowledges that not all specifications require all information types. However, this structure assumes these three categories comprehensively cover information needs, potentially constraining representation of emerging requirement types such as behavioral specifications or process-related information.

The universal application of GUIDs for element identity establishes a stable referencing mechanism independent of element names, positions, or hierarchical relationships. This approach supports requirement traceability and change management by providing persistent identifiers that remain valid across document modifications. The distinction between *dt:GUID* for internal references and *dt:referenceURI* for external references, with their combination enabling references to specific elements within external resources, implements a two-tier resolution strategy. Internal resolution operates within document scope without external dependencies, while external resolution requires access to referenced resources. The schema's guidance to prefer GUID-only for internal references and URI-only for external resources suggests optimization for reference resolution performance and reliability, though the combined approach enables precise external element targeting when necessary.

The five-aspect geometric requirements structure (*Detail, Dimensionality, Appearance, ParametricBehaviour, Location*) aligns with ISO/TS 7817-2 guidelines for geometric specification. This decomposition treats geometric information as multidimensional rather than monolithic, acknowledging that different project phases or use cases may require different geometric characteristics. The *ShapeInfluence* mechanism within *Detail* enables fine-grained specification of individual geometric features with threshold-based rules, supporting complexity management where minor features may be omitted while significant features require explicit modeling. However, the reliance on enumerated values for each aspect constrains expressiveness to predefined options, potentially limiting specification of novel geometric requirements not anticipated in the enumeration design.

The practical implementation of the proposed schema necessitates a shift from purely geometric modeling toward robust semantic integrity, requiring rigorous alignment between proprietary parametric environments and standardized data structures. This integration typically demands the development of bespoke middleware or API-driven scripts to facilitate the mapping of local attributes to schema-specific fields, thereby ensuring that data fidelity is maintained throughout the project lifecycle. Furthermore, the operationalization of the schema is contingent upon the precise calibration of the Level of Information Need, which prevents computational overhead by restricting data density to the specific requirements of defined project milestones. In practice, the schema finds utility i.e. in the automation of quantity takeoffs and the enhancement of multi-disciplinary coordination through automated data validation; this allows for the identification of "data gaps" where physical components lack the requisite metadata for facility management.

Conclusions

The schema represents a significant advancement in formalizing information requirements specification across different dimensions of detail within the AEC industry, providing technical mechanisms for precise, machine-readable requirement definition. The schema's utility depends not primarily on its technical merits but on ecosystem development including software tool support, reference resource availability, practitioner capability building, and governance framework establishment. These ecosystem elements remain underdeveloped, suggesting near-term adoption will be limited to advanced organizations and pilot projects rather than achieving broad industry penetration. The schema provides necessary technical infrastructure for digital information delivery but proves insufficient without surrounding social and technical systems.

Future success depends on recognizing schema implementation as a socio-technical challenge requiring coordinated development of technical, organizational, and human capabilities rather than purely a standards adoption problem.

Acknowledgments

The authors would like to thank ISO, CEN and national standardization bodies and collaborators for the support throughout the development of the mentioned standards, especially CENT/TC 442/WG 2 and JWG ISO/TC 59/SC 1

References

Abualdenien, J. and Borrmann, A. (2022). Levels of detail, development, definition, and information need: a critical literature review. Journal of Information Technology in Construction, 27:363–392.

Akbas, E., Bolpagni, M., Borrmann, A., Boeykens, S., Mellenthin Filardo, M., Liu, L., and Beetz, J. (2025). A holistic approach to information requirements: Integration of level of information need and information delivery specification. J. Inf. Technol. Constr., 30:731–744.

Bolpagni, M., Bosché, F., de Boissieu, A., Akbarieh, A., Shaw, C., Mêda, P., Puust, R., Medineckiene, M., Popov, V., and Sacks, R. (2022). An explorative analysis of European standards on building information modelling. Proceedings of the 2022 European Conference on Computing in Construction, volume 3 of Computing in Construction.

European Committee for Standardization and International Organization for Standardization (2016). Guidelines for the implementation of the Agreement on Technical Cooperation between ISO and CEN. ISO/CEN, 7 edition.

Hodgson, R. (2024). Quantities, Units, Dimensions and Types (QUDT) Schema - Version 2.1.35. last accessed: 14.01.2026.

ISO 12006-3 (2023). Building construction— Organization of information about construction works: Part 3: Framework for object-oriented information. International Organization for Standardization, Geneva, CH, 1 edition.

ISO 19650-1 (2018). Organization of information about construction works - Information management using building information modelling. Part 1: Concepts and principles. International Organization for Standardization, Geneva, CH, 1 edition.

ISO 23386 (2020). Building information modelling and other digital processes used in construction: Methodology to describe, author and maintain properties in interconnected data dictionaries. International Organization for Standardization, Geneva, CH, 1 edition.

ISO 23387 (2025). Building information modelling (BIM) — Data templates for construction objects used in the life cycle of built assets: Concepts and principles. International Organization for Standardization, Geneva, CH, 1 edition.

ISO 29481-3 (2022). Building information models — Information delivery manual: Part 3: Data schema. International Organization for Standardization, Geneva, CH, 1 edition.

ISO 7817-1 (2024). Building information modelling—Level of information need: Part 1: Concepts and principles. International Organization for Standardization, Geneva, CH, 1 edition.

ISO/TS 7817-2 (2025). Building information modelling—Level of information need: Part 2: Guidelines for Application. International Organization for Standardization, Geneva, CH, 1 edition.

ISO 7817-3 (2025). Building information modelling—Level of information need: Part 3: Data model and schema. International Organization for Standardization, Geneva, CH, 1 edition.

Jaud, Š., Gregory, L. J., Borrmann, A. (2021). Georeferencing. T. H. Kolbe, R. Bill, A. Donaubauer (eds), Geoinformationssysteme 2021: Beiträge zur 8. Münchener GI-Runde, Runder Tisch GIS e.V., Munich, Germany, 49-50.

Jeon, K., Lee, G., Kang, S., Roh, H., Jung, J., Lee, K. and Baldwin, M. (2021). A relational framework for smart information delivery manual (IDM) specifications. Advanced Engineering Informatics, vol. 49, p. 101319.

Mellenthin Filardo, M., Liu, L., Hagedorn, P., Zentgraf, S., Melzner, J., and König, M. (2024a). A standard-based ontology network for information requirements in digital construction projects. Proceedings of the 12th Linked Data in Architecture and Construction Workshop (LDAC 2024).

Mellenthin Filardo, M., Debus, P., Melzner, J., Bargstädt, H.-J. (2024b). Enhancing Information Requirement Compliance Through Automated Integration in a Modelling Environment. Proceedings of the 31st International Workshop on Intelligent Computing in Engineering (EG-ICE), Vigo, Spain.

Sacks, R., Lee, G., Burdi, L., and Bolpagni, M. (2025). BIM handbook: A guide to building information modeling for owners, designers, engineers, contractors, and facility managers. John Wiley & Sons, Hoboken, New Jersey, 4 edition.

Son, S., Lee, G., Jung, J., Kim, J., and Jeon, K. (2022). Automated generation of a model view definition from an information delivery manual using idmXSD and buildingSMART data dictionary. Advanced Engineering Informatics, 54:101731.

Zentgraf, S., Mellenthin Filardo, M., Liu L., Hagedorn, P., Melzner J., König, M. (2025). Update of the standard-based ontology network for information requirements in digital construction projects, LDAC 2025: 13th Linked Data in Architecture and Construction Workshop, July 09-11, 2025, Porto, Portugal.