TECHNICAL SPECIFICATION

ISO/TS 10303-15

First edition

Industrial automation systems and integration — Product data representation and exchange —

Part 15: Description methods: SysML XMI to XSD transformation

PROOF/ÉPREUVE

Reference number
ISO/TS 10303-15:2021(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 184, *Automation systems and integration*, Subcommittee SC 4, *Industrial data.*

A list of all parts in the ISO 10303 series can be found on the ISO website at https://standards.iso.org/iso/10303/tech/step_titles.htm.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product and independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

This document is a member of the description methods series. This document specifies a mapping of SysML XMI to the XSD. This document supports the STEP extended architecture.[17][18][19] This document specifies the transformation from a STEP data model in SysML XMI to a STEP XSD.

The object management group (OMG) has standardized the XML metadata interchange specification (XMI) that integrates the OMG systems modeling language (SysML), the OMG unified modeling language (UML), and the World Wide Web Consortium (W3C) extensible markup language (XML). SysML inherits the XMI interchange capability from UML. XMI is a mechanism for the interchange of metadata between UML-based modeling tools. OMG has also standardized an XMI compliant interchange format for the SysML thus specifying a lexical representation of SysML models based on a standardized metamodel of the SysML.

The W3C has standardized the XML schema definition (XSD). XSD allows to define shared vocabularies and allow machines to carry out rules made by developers. They provide a means for defining the structure, content and semantics of XML documents.

This document specifies a description method of the STEP parts family, which defines the transformation of SysML constructs to the XSD constructs. Because the XMI standard specifies the XML representation of SysML metamodel constructs, standardizing the binding of SysML constructs into XSD constructs supports the representation of SysML models as XML schemas.

The specified mapping is a one-way transformation from SysML information model represented in XMI into an XML schema. These limitations make the mapping unsuitable for the transformation of arbitrary SysML models to XSD.

A detailed knowledge of the W3C XML and XSD languages, and the OMG systems modelling language is useful.

The main components of this document are:

— the structure, conventions and concepts of the XSD;

— the specification of the transformation from SysML XMI to XSD for each STEP element modelled in SysML.

Industrial automation systems and integration — Product data representation and exchange —

Part 15: Description methods: SysML XMI to XSD transformation

1 Scope

This document specifies the transformation of SysML (ISO/IEC 19514:2017) constructs to XSD (World Wide Web Consortium's XML schema definition language) constructs for the purpose of representing the SysML model represented in XMI (ISO/IEC 19509:2014) as XML (World Wide Web Consortium's XML) schemas. The specified mapping is a one-way transformation from SysML information model represented in XMI into an XML schema. These limitations make the mapping unsuitable for the transformation of arbitrary SysML models to XML schemas.

The following are within the scope of this document:

— the specification of the structure, components, and conventions of the XSD for the STEP (ISO 10303-1) XML implementation method;

— the transformation of SysML metamodel constructs represented in XMI to XSD constructs for the purpose of representing SysML information models as XML schemas.

The following are outside the scope of this document:

— the transformation of SysML metamodel constructs into XSD constructs that are not used in the STEP extended architecture;

— the transformation of SysML metamodel constructs into XSD constructs for other purposes than representing SysML constructs as STEP concepts;

— codes and scripts to transform SysML XMI to XSD schema;

— the transformation of SysML constraints (OCL, see ISO/IEC 19507) into Schematron (see ISO/IEC 19757-3).

2 Normative references

There are no normative references in this document.

ISO 10303-11:2004, *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual*

ISO/IEC 19505-1:2012, *Information technology — Object Management Group Unified Modeling Language (OMG UML) — Part 1: Infrastructure*

ISO/IEC 19514:2017, *Information technology — Object management group systems modeling language (OMG SysML)*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 10303-11, ISO/IEC 19505-1, ISO/IEC 19514 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1.1 Terms and definitions for generic concepts

3.1.1.1
EXPRESS
language by which aspects of product data can be defined

3.1.1.2
application object
atomic element of an application reference model that defines a unique concept of the application and contains attributes specifying the data elements of the object

[SOURCE: ISO 10303-1:1994 2.1.11]

3.1.1.3
data
representation of information in a formal manner suitable for communication, interpretation, or processing by human beings or computers

[SOURCE: ISO 10303-1:2021, 3.1.29]

3.1.1.4
data model
description of the organization of data in the management information system of an enterprise.

[SOURCE: ISO/IEC 2382:2015, 2121422]

3.1.1.5
implementation method
technique used by computer systems to exchange product data

[SOURCE: ISO 10303-1:2021, 3.1.39, modified — In the definition, "part of ISO 10303" has been replaced with "technique" and the text after "data" has been removed.]

3.1.1.6
information
facts, concepts, or instructions

[SOURCE: ISO 10303-1:2021, 3.1.41]

3.1.1.7
resulting XSD
XSD based on the transformation specification

3.1.1.8
information model
conceptual model of product data

Note 1 to entry: In ISO 10303, an information model is based on the object-relationship modeling technique that organizes the product data as represented in different system aspects.

Note 2 to entry: In ISO 10303, information models may be developed using EXPRESS modeling language.

EXAMPLE Application resource model for ISO 10303-242 managed model-based 3D engineering

[SOURCE: ISO 10303-1:2021, 3.1.42, modified — In the definition, "formal" has been replaced with "conceptual"; in Note 2 to entry, "are" has been replaced with "may be"; the Example has been changed.]

3.1.1.9
uniform resource identifier
URI
string of characters that unambiguously identifies a particular resource

[SOURCE: RFC 3986]

3.1.2 Terms and definitions for SysML constructs

3.1.2.1
canonical XMI
specific constrained format of XMI that minimizes variability and provides more predictable identification and ordering

Note 1 to entry: A canonical XMI file is itself a valid XMI file.

Note 2 to entry: The full definition is provided in ISO/IEC 19509:2014, Annex B.

3.1.2.2
association
association classifies a set of tuples representing links between typed model elements

Note 1 to entry: The full definition is provided in ISO/IEC 19505-1:2012, 11.5.

3.1.2.3
auxiliary
stereotype applied to an abstract block that has no properties

Note 1 to entry: The full definition is provided in ISO/IEC 19505-1:2012, Clause 22.

3.1.2.4
block
modular construct used for defining an entity

Note 1 to entry: Used for defining application activity model concepts, application data planning objects, application domain model business objects, core model objects and ARM in SysML entities. They can include: reference, part, and value properties; constraints. They can be specializations of other blocks.

Note 2 to entry: The full definition is provided in ISO/IEC 19514:2017, Clause 8.

3.1.2.5
composite aggregation
responsibility for the existence of composed object

Note 1 to entry: If a composite object is deleted, all of its part instances that are objects are deleted with it.

Note 2 to entry: The full definition is provided in ISO/IEC 19505-1:2012, 11.4.1.

3.1.2.6
directed association
association between a collection of source model elements and a collection of target model elements that is said to be directed from the source elements to the target elements

Note 1 to entry: The full definition is provided in ISO/IEC 19505-1:2012, 7.2.3.3.

3.1.2.7
enumeration
value type whose values are enumerated

Note 1 to entry: The full definition is provided in ISO/IEC 19505-1:2012, 10.2.3.3.

3.1.2.8
enumeration literal
named value for an *enumeration* (3.1.2.7)

Note 1 to entry: The full definition is provided in ISO/IEC 19505-1:2012, 10.2.3.3.

3.1.2.9
data type
type whose instances are identified only by their value

Note 1 to entry: The full definition is provided in ISO/IEC 19505-1:2012, 10.2.3.1.

3.1.2.10
generalization
directed relationship between a more general supertype and a more specific subtype

Note 1 to entry: Each generalization relates a specific classifier to a more general classifier. Given a classifier, the transitive closure of its general classifiers is often called its generalizations, and the transitive closure of its specific classifiers is called its specializations. The immediate generalizations are also called the classifier's subtype, and where the classifier is a class, its supertype.

Note 2 to entry: The full definition is provided in ISO/IEC 19505-1:2012, C.1.1.

3.1.2.11
primitive type
definition of a predefined data type, without any substructure

Note 1 to entry: The full definition is provided in ISO/IEC 19505-1:2012, Clause 21.

3.1.2.12
part property
property that specifies a part with strong ownership and coincidental lifetime of its containing block

Note 1 to entry: It describes a local usage or a role of the typing block in the context of the containing block. Every part property has composite aggregation and is typed by a block.

Note 2 to entry: The full definition is provided in ISO/IEC 19514:2017, 8.3.2.3, paragraph 6.

3.1.2.13
reference property
property that specifies a reference of its containing block to another block

Note 1 to entry: The full definition is provided in ISO/IEC 19514:2017, 8.3.2.3, paragraph 6.

3.1.2.14
stereotype
limited kind of metaclass that cannot be used by itself but must always be used in conjunction with one of the metaclasses it extends

Note 1 to entry: The full definition is provided in ISO/IEC 19505-1:2012, 12.3.3.4.

3.1.2.15
value property
property of a block that is typed with a value type

Note 1 to entry: The full definition is provided in ISO/IEC 19514:2017, 8.3.2.3, paragraph 6.

3.1.2.16
value type
stereotype of UML data type that is used to define types of values that may be used to express information but cannot be identified as the target of any reference

Note 1 to entry: The full definition is provided in ISO/IEC 19514:2017, 8.3.2.14.

3.1.3 Terms and definitions for XSD constructs

3.1.3.1
schema definition language
language for XML schemas

Note 1 to entry: The purpose of an XSD schema is to define and describe a class of XML documents by using schema components to constrain and document the meaning, usage and relationships of their constituent parts: datatypes, elements and their content and attributes and their values.

[SOURCE: World Wide Web Consortium's XML schema definition language (W3C XSD)]

3.1.3.2
global complex type
complex type (3.1.3.3) that is defined globally in an XML schema

Note 1 to entry: A xsd:complexType can also defined globally and given a name. Named xsd:complexTypes can then be re-used throughout the schema, either referenced directly or used as the basis to define other xsd:complexTypes. This makes it possible to build more object-oriented data structures that are easier to work with and manage. Only complex types defined globally (as children of the <xsd:schema> element can have their own name and be re-used throughout the schema). If they are defined inline within an <xsd:element> they cannot have a name (anonymous) and cannot be reused elsewhere.

Note 2 to entry: The full definition is provided in World Wide Web Consortium's XML schema definition language.

3.1.3.3
complex type
set of attribute definitions and content type for an element in an XML schema

Note 1 to entry: A xsd:complexType provides the definition for an XML element. It specifies which element and attributes are permitted and the rules regarding where they can appear and how many times. They can be used in-place within an element definition or named and defined globally.

Note 2 to entry: the full definition is provided in World Wide Web Consortium's XML schema definition language.

3.1.3.4
attribute type
name, type and occurrence for a property in an XML schema

Note 1 to entry: An attribute provides extra information within an element. Attributes have name and type properties. An Attribute can appear 0 or 1 times within a given element in the XML document. Attributes are either optional or mandatory (by default they are optional). The "use" property in the XSD definition is used to specify if the attribute is optional or mandatory. An attribute is specified within a xsd:complexType, the type information for the attribute comes from a xsd:simpleType (either defined inline or via a reference to a built in or user defined xsd:simpleType definition). The type information describes the data the attribute can contain in the XML document, such as string, integer, date. Attributes can also be specified globally and then referenced.

Note 2 to entry: The full definition is provided in World Wide Web Consortium's XML schema definition language.

3.1.3.5
compositor
rules for ordering in an XML schema

Note 1 to entry: Compositors provide rules that determine how and in what order their children can appear within XML document. There are three types of compositors <xsd:sequence>, <xsd:choice> and <xsd:all>.

— Sequence: The child elements in the XML document shall appear in the order they are declared in the XSD schema.

— Choice: Only one of the child elements described in the XSD schema can appear in the XML document.

— All: The child elements described in the XSD schema can appear in the XML document in any order.

Note 2 to entry: The full definition is provided in World Wide Web Consortium's XML schema definition language.

3.1.3.6
extension
complex type (3.1.3.3) that is inherited

Note 1 to entry: It is possible to take an existing <xsd:complexType> and extend it using <xsd:extension> and the "base" attribute. The introduced construct <xsd:extension> indicates that an existing type is extended and specifies a new type. The construct <xsd:complexContent> shall be used to as container for the extension.

Note 2 to entry: The full definition is provided in World Wide Web Consortium's XML schema definition language.

3.1.3.7
group
reusable collection of elements and attributes in an XML schema

Note 1 to entry: Elements and attributes can be grouped together using <xsd:group> and <xsd:attributeGroup>. These groups can then be referred to elsewhere within the schema. Groups shall have a unique name and be defined as children of the <xsd:schema> element. When a group is referred to, it is as if its contents have been copied into the location it is referenced from.

Note 2 to entry: The full definition is provided in World Wide Web Consortium's XML schema definition language.

3.1.3.8
mixed content
complex type (3.1.3.3) that may contain, attributes elements and text

Note 1 to entry: Elements can also contain a combination of element types, complex types and compositors. Elements and data can be mixed.

Note 2 to entry: The full definition is provided in World Wide Web Consortium's XML schema definition language.

3.1.3.9
namespace
scope for named elements in an XML file

Note 1 to entry: Namespaces are a mechanism for breaking up your schemas. XSD standard allows to structure XSD schemas by breaking them into multiple files. These child schemas can then be included into a parent schema. Breaking schemas into multiple files can have several advantages. One can create re-usable definitions that can be used across several projects. They make definitions easier to read and version as they break down the schema into smaller units that are simpler to manage.

Note 2 to entry: The full definition is provided in World Wide Web Consortium's XML schema definition language.

3.1.3.10
restriction
definition of acceptable values for elements in an XML schema

Note 1 to entry: The usage of extensions, mixed contents, namespaces, groups, provides the capability to restrict the definition of a type.

Note 2 to entry: The full definition is provided in World Wide Web Consortium's XML schema definition language.

3.1.3.11
simple type
element type with text-only attributes in an XML schema

3.2 Abbreviated terms

CXMI canonical XMI

ID identifier

OCL object constraint language

OMG object management group

STEP standard for the exchange of product model data

SysML systems modeling language

UML unified modeling language

UoS unit of serialization

UUID universal unique identifier

URI uniform resource identifier

XMI XML meta-data interchange

XML extensible markup language

XSD XML schema definition

4 Structure and components of the XSD

4.1 General

This clause describes the structure and components of the resulting XSD. The resulting XSD shall conform to World Wide Web Consortium's XML schema definition language (W3C XSD).

This document shall be unambiguously identified in an open information system by the code defined in Annex A.

The chapter 5 SysML XMI to XSD provides the mapping specification of each mentioned XSD constructs in this clause.

4.2 Presentation conventions

For ease of identification, the fragments of XSD are presented in boxes.

EXAMPLE 1 XSD fragment presented in a box

```
XSD extract
```

The items significant to support the explanations are formatted using bold text effect to aid identification of the items in the XSD fragment.

EXAMPLE 2 Usage of bold text effect to support the explanation

An XML **attribute** shall be contained in the XML element:

```
<Element attribute="...">
...
```

4.3 Main components of the XSD

The resulting XSD shall be composed of the following main components:

— the XSD header, located at the beginning of the file;

— the import of the common definitions contained in common.xsd;

— the specification of the *DataContainer* as a choice of base root objects;

— the Key-KeyRef references;

— the list of application object specifications in XSD;

— the groups and simple types corresponding to selects and enumerations.

Each of these components are described in 4.4 to 4.9.

4.4 XSD header

The header of the XSD (xsd:schema) defines:

— the namespaces of the XSD schema,

— resulting XML schema version.

— Regarding the namespace conventions, the namespace prefixes are used throughout this part to refer to the namespaces identified by the corresponding URI. The prefixes and associated URIs are the following:

— xmlns:targetNamespace: https://standards.iso.org/iso/ts/10303/-4442/-ed-1/tech/xml-schema/domain_model;

— xmlns: https://standards.iso.org/iso/ts/10303/-4442/-ed-1/tech/xml-schema/domain_model;

— xsd: http://www.w3.org/2001/XMLSchema;

— xsi: http://www.w3.org/2001/XMLSchema-instance;

— cmn: https://standards.iso.org/iso/ts/10303/-3000/-ed-2/tech/xml-schema/common;

A valid resulting XSD header is provided in the following XSD fragment:

```
XSD:
<xsd:schema
   xmlns="https://standards.iso.org/iso/ts/10303/-4442/-ed-1/tech/xml-schema/domain_model"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:cmn="https://standards.iso.org/iso/ts/10303/-3000/-ed-2/tech/xml-schema/common"
   targetNamespace="https://standards.iso.org/iso/ts/10303/-4442/-ed-1/tech/xml-schema/domain_model"
   version="N10475;2019-06-07">
```

4.5 Common definitions from common.xsd

The header of the XSD shall be followed by the import of the external XSD. This external XSD specifies that the resulting XSD shall contain a unit of serialization (UoS) by declaring an xsd:element named UoS. This external XSD is provided at https://standards.iso.org/iso/ts/10303/-3000/-ed-2/tech/xml-schema/common/common.xsd.

The xsd:ComplexType UoS specifies that an XML shall include a UoS object that includes a header (described below) and one or more DataContainer (see 4.6 below).

The resulting XSD shall add the Key and KeyRef (see 4.7) declarations to this xsd:element.

The declaration of the **common.xsd** XSD import shall be as follows:

XSD:

<xsd:import namespace=https://standards.iso.org/iso/ts/10303/-3000/-ed-1/tech/xml-schema/common

schemaLocation="https://standards.iso.org/iso/ts/10303/-3000/-ed-1/tech/xml-schema/common/**common.xsd**"/>

NOTE 2 The use of “cmn:” in the examples in this document imply the namespace for the 10303 common schema. The contents of the common.xsd is in Annex B.

The mandatory header element that contains administrative information that characterizes the content of the data package. The header elements are described in ISO 10303-28:2007, 5.2, and are as follows:

— Name: human readable identifier for the XML resource;

— TimeStamp: date and time when the XML resource was created;

— Author: identifies the person or group of persons who created the XML resource;

— Organization: identifies the organization that created, or is responsible for, the XML resource;

— PreprocessorVersion: identifies the software system that created the XML resource itself, including platform and version identifiers;

NOTE 3 The preprocessor_version identifies the system that was used to produce the XML resource. This can be distinct from the software system that created or captured the original information.

— OriginatingSystem: identifies the software system that created or captured the information contained in the XML resource, including platform and version identifiers;

— Authorization: specifies the release authorization for the XML resource and the signa-tory, where appropriate;

NOTE 4 The authorization can be distinct from the authorizations for various information units contained within the document.

— Documentation: free text field for information.

The UoS XSD also defines the following structural features:

— BaseObject: This is the generic object from which all entities are extended. This element type is abstract. This object specifies that all XML elements may have the XML attribute “uid” typed by the standardized xsd:ID type. In XML dataset ruled by this XSD, its element shall have unique “uid” XML attributes;

EXAMPLE 1 Extract of the **BaseObject** specification