ISO TC 184/SC4 N1190:2001(E)
2001-08-29

ISO TC 184/SC4 STANDING DOCUMENT

Technical Committee 184 for Industrial Automation Systems and Integration
Subcommittee 4 for Industrial Data

Guidelines for the development of mapping specifications, 2nd edition

This standing document replaces
Guidelines for the development of mapping tables (SC4 N533).

ISO TC 184/SC4 Secretariat
DISA/JECPO Center for Standards
10710 Parkridge Boulevard
Reston, Virginia 22091
USA
email: sc4sec@cme.nist.gov

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.

This standing document was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC4, *Industrial data*.

International Standards produced by ISO/TC 184/SC4 are prepared according to guidelines put forth in the following standing documents:

— ISO/TC 184/SC4 Organization handbook;

— SC4 Quality manual;

— SC4 Supplementary directives - Rules for the structure drafting of SC4 standards for industrial data.

The following standing documents provide additional guidelines for developing parts of ISO 10303 Product data representation and exchange:

— Guidelines for application interpreted construct development;

— Guidelines for application interpreted model development;

— Guidelines for the development and approval of STEP application protocols;

— Guidelines for the development of abstract test suites, 2nd edition;

— Guidelines for the development of mapping specifications, 2nd edition.

This standing document replaces Guidelines for the development of mapping tables (SC4 N533).

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 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.

ISO 10303 is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application interpreted constructs, application protocols, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1.

The purpose of this standing document is to provide methods and procedures for the development of mapping specifications that appear within ISO 10303 application protocols. The mapping specification is a pivotal component of an application protocol (AP). The mapping specification documents the traceability of the application information requirements between the specification of these requirements in clause 4 of the AP and the application interpreted model (AIM) in clause 5 of the AP. The mapping specification documents how standardized constructs are applied to satisfy the application information requirements. This document is intended to provide guidance to application protocol development teams on the documentation of mapping specifications. This document is also intended to help reviewers and implementors of APs to understand mapping specifications. Specifics on style, format, required text, and other presentation issues are provided in the *SC4 Supplementary directives – Rules for the structure and drafting of SC4 standards for industrial data*. Additional guidance on other areas of AP development is found in the *Guidelines for development and approval of STEP application protocols* and *Guidelines for application interpreted model development*.

This is the second edition of this standing document. There are three primary changes from the first edition.

— The format of the mapping specification has been changed from a table to a nested subclause structure, and the term mapping specification has replaced the term mapping table. This has caused pervasive changes throughout the document, requiring removing all mention of tables or table components. Related to this change, when mappings are clarified by the use of numbered alternatives, the variations are complete mappings as opposed to mapping fragments; each alternative specifies each component of the mapping completely within a particular subclause (see 6.2).

— A syntax for defining one time and reusing portions of a reference path, called templates, is provided in this edition (see annex A).

— An initial formal syntax for mapping specifications is provided in this edition (see annex D).

Minor changes to the use of symbols in the reference path have also been made.

— The equal sign "=" may now be used to assign a specific type to an item of an aggregate (see 9.1.3).

— When using parenthesis, square brackets and angle brackets with complex paths where only a portion of the path starts at an integrated resource entity data type, that resource entity data type must now also be included in the reference path (see 9.1.4, 9.1.5, 9.1.6).

- Rules are no longer sequentially numbered and included in a list at the end of 5.1 of the AP. They are referenced through the rule identifier and optionally the subclause number of the subclause in the AP in which they are defined (see clause 8).

The mapping specification is organized by unit of functionality (UoF) as defined in 4.1 of the AP. Each information requirement of an AP is mapped to one or more constructs in the AIM. Each of these mappings is documented in a subclause of the mapping specification for the UoF in which the application element is defined. Up to four types of information may be needed to completely specify the mapping of a given application element. These are: “AIM element”, “Source”, “Rules”, and “Reference path”. This document provides guidance on the how to populate each type of content in the mapping specification.

To aid in the understanding of this document, several examples taken from ISO 10303-201 are included within the body of this text. Additionally, a more comprehensive example is provided in annexes B and C of this document. In the example shown in the annexes, four application objects have been chosen from one of the UoFs in ISO 10303-201. Annex B contains the descriptions of the UoF, the application objects, and the application assertions as normatively documented in clause 4 of ISO 10303-201. Annex C contains the mapping specification for the application objects and assertions in annex B.

An exception to the rule that examples must appear in 10 point type has been made in various places throughout this document so that editors can see exactly how the information being presented should appear. Examples that show specific formats or layouts to be used in SC4 standards are distinguished from other text by a light grey background. Examples have been reformatted from the source AP to match the new clause structure for mapping specifications.

Throughout the body of this text, references to application elements in examples are presented with leading capitals and underscores between the words. References to AIM elements in examples are presented in bold face with underscores between words, but no leading capitals.

Guidelines for the development of mapping specifications

1 Scope

This standing document specifies procedures and practices for the development and documentation of mapping specifications within ISO 10303 application protocols.

The following are within the scope of this standing document:

- description of the structure of mapping specifications for ISO 10303 application protocols;
- instructions for detailing the contents the mapping specification;
- explanation of when reference paths are required;
- procedures for identification of the start and end entity data types for required paths;
- examples that clarify the guidance provided within this document.

The following are outside the scope of this standing document:

- specification of font sizes and other layout information for the documentation of mapping specifications;

NOTE 1 This information is found in *SC4 Supplementary directives - Rules for the structure drafting of SC4 standards for industrial data.*

- guidance on how to map the constructs of the ISO 10303 integrated resources to the information requirements of an ISO 10303 application protocol;

NOTE 2 This information is available to a small extent in *Guidelines for application interpreted model development*.

- guidelines for development of mapping specifications for documents other than ISO 10303 application protocols.

2 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this standing document. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standing document are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of the IEC and ISO maintain registers of currently valid International Standards.

ISO 10303-1:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles.*

ISO 10303-202:1996, *Industrial automation systems and integration — Product data representation and exchange — Part 202: Associative draughting.*

The following SC4 standing documents contain provisions which, through reference in this text, constitute provisions of this standing document. At the time of adoption, the revisions of the documents indicated were valid. All documents are subject to revision, and users of this standing document are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below.

ISO/TC 184/SC4 N532:1997, *Guidelines for application interpreted model development*

ISO/TC 184/SC4 N535:1998, *Guidelines for the development and approval of STEP application protocols*

ISO/TC 184/SC4 N1191:2001, *SC4 Supplementary directives - Rules for the structure drafting of SC4 standards for industrial data.*

NOTE A list of current SC4 standing documents and other material to be used by developers of SC4 standards is available from the Internet:

`<http://www.nist.gov/sc4/www/necsdocs.htm>`

3 Terms, definitions and abbreviations

3.1 Terms defined in ISO 10303-1

For the purpose of this standing document, the following terms defined in ISO 10303-1 (repeated below for convenience) apply.

3.1.1
application
a group of one or more processes creating or using product data

3.1.2
application context
the environment in which the integrated resources are interpreted to support the use of product data in a specific application

3.1.3
application interpreted model (AIM)
an information model that uses the integrated resources necessary to satisfy the information requirements and constraints of an application reference model, within an application protocol

3.1.4
application object
an 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

3.1.5
application protocol (AP)
a part of this International Standard that specifies an application interpreted model satisfying the scope and information requirements for a specific application

NOTE This definition differs from the definition used in open system interconnection (OSI) standards. However, since this International Standard is not intended to be used directly with OSI communications, no confusion should arise.

3.1.6
application reference model (ARM)
an information model that describes the information requirements and constraints of a specific application context

3.1.7
data
a representation of information in a formal manner suitable for communication, interpretation, or processing by human beings or computers

3.1.8
implementation method
a part of this International Standard that specifies a technique used by computer systems to exchange product data that is described using the EXPRESS data specification language [ISO 10303-11]

3.1.9
interpretation
the process of adapting a resource construct from the integrated resources to satisfy a requirement of an application protocol. This may involve the addition of restrictions on attributes, the addition of constraints, the addition of relationships among resource constructs

3.1.10
product data
a representation of information about a product in a formal manner suitable for communication, interpretation, or processing by human beings or by computers

3.1.11
resource construct
a collection of EXPRESS language entities, types, functions, rules and references that together define a valid description of an aspect of product data

3.1.12
unit of functionality (UoF)
a collection of application objects and their relationships that defines one or more concepts within the application context such that removal of any component would render the concepts incomplete or ambiguous

3.2 Terms defined in ISO 10303-202

For the purpose of this standing document, the following terms defined in ISO 10303-202 (repeated below for convenience) apply.

3.2.1
application interpreted construct (AIC)
a logical grouping of interpreted constructs that supports a specific function for the usage of product data across multiple application contexts

3.3 Other terms and definitions

For the purposes of this standing document, the following definitions apply.

3.3.1
application element
application object, attribute of an application object, or assertion of a relationship between two application objects

3.3.2
common resource
part of an SC4 standard that defines a group of resource constructs that can be used or interpreted in other SC4 standards

NOTE 1 The constructs defined in a common resource may be used or interpreted in all other SC4 standards or in a subset of the other SC4 standards.

NOTE 2 Common resources include the integrated resources, application interpreted constructs, and application modules parts of ISO 10303, together with a number of designated parts of other SC4 standards such as ISO 13584-20 and 15531-42.

3.3.3
mapping signature
definition of the format to be used for a template call

3.3.4
mapping specification
element of an application protocol that shows how the interpretation of integrated resources is used to meet the information requirements of the application

3.3.5
mapping template
reusable portion of a reference path that defines a commonly used part of the structure of an application interpreted model

3.3.6
root node
entity data type declared in an application interpreted construct that localizes constraints pertaining to that application interpreted construct

3.4 Abbreviations

For the purposes of this standing document, the following abbreviations apply.

AIC	application interpreted construct (see 3.2.1)

AIM	application interpreted model (see 3.1.3)
AP	application protocol (see 3.1.5)
ARM	application reference model (see 3.1.6)
UoF	unit of functionality (see 3.1.12)

4 Fundamental concepts and assumptions

The mapping specification documents the correspondence between the information requirements defined in clause 4 of an AP and how the requirements are satisfied by the objects in the AIM. This correspondence describes how the information that is created and used within the application domain of the AP can be represented using the data structures defined by the ISO 10303 integrated resources.

The mapping specification is established through analysis of the information requirements and the definition of a mapping for each application object, application object attribute, and application object assertion defined in clauses 4.2 and 4.3 of the AP. The analysis process takes the following into account:

- the scope and context of the AP, i.e., the activities that are to be supported using data exchanges based on the AIM and information that is to be communicated in those data exchanges;
- the meaning of the application objects;
- the meaning of the resource constructs defined in the ISO 10303 integrated resources;
- interpretation of general resource constructs into the specific scope and context of the AP, by inclusion of the resource construct in the AIM of the AP;
- definition of constraints on the population of the resource constructs used in the AIM; these constraints can be defined using mapping rules in the reference path element of the mapping specification, definition of subtypes of resource constructs, or definition of global rules;
- consistency of use of resource constructs within each AP;
- consistency of use of resource constructs across different APs that overlap in their scope and context.

The mapping specification should be understood as defining the correspondence between instances in the ARM (i.e., typical application data) and instances in the AIM (STEP exchange file or shared database). The primary focus of the mapping specification is to define the pattern of AIM instances and its associated constraints that corresponds to each ARM element (application object, attribute, and assertion). With suitable implementation technology, the mapping specification can also be regarded as defining queries against a population of the AIM where these queries result in the ARM "view" of the data held in an exchange file or a shared database.

Validation of the mapping should confirm that each application object, attribute, or assertion has a unique mapping to the AIM structure. In this context, the uniqueness of a mapping means that for every possible instance of the ARM there is a different (set of) instance(s) of the AIM.

There is no requirement in ISO 10303 application protocols to explicitly document the mapping from AIM to ARM. However, as part of the application interpretation process and the validation of the mapping specification, projects are encouraged to document the usage of the integrated resources in order to verify that there is no ambiguity in the AIM to ARM mapping and that the meaning of the application objects, attributes, and assertions can therefore be inferred from any valid and useful population of the AIM.

5 Application element documentation

The mapping specification consists of a series of subclauses, each of which documents the mapping for one Unit of Functionality, as described in *SC4 Supplementary directives - Rules for the structure drafting of SC4 standards for industrial data*. An application object may appear in more than one UoF within an AP. When this occurs, the subclause within each UoF mapping specification documents the mapping of that element within the context of that UoF. Application objects not listed in any UoF shall be listed in a final subclause called "Additional application objects".

The application objects (i.e. entity data types and attributes) and assertions from clause 4 of the AP become subclause headings within the UoF mapping specification in accordance with the guidance provided in the *SC4 Supplementary directives*. Each application element from the application protocol appears as the heading of at least one subclause of the specification. For EXPRESS-G ARMs, simple attributes shall be mapped as attributes, relationships shall be mapped as assertions, in accordance with the documentation of clause 4 of the AP. Each assertion shall be mapped in a separate subclause under the primary application object's mapping specification subclause.

The requirements of the application protocol may define more than one assertion between two application objects. When this occurs, each assertion is entered in a separate subclause of the mapping specification. The subclause heading is the heading of the application assertion from 4.3 of the AP along with an identifying phrase. The phrase is chosen from the normative text of the assertion and is placed in parentheses following the assertion heading. The assertions appear in the mapping specification in the order that they are defined in 4.3 of the AP.

EXAMPLE 1 In this example, three assertions between a pair of application objects are documented in 4.3 of an AP and in the mapping specification. The mappings of the multiple assertions will appear as separate subclauses in the mapping specification. The details of the mapping are omitted here.

4.3.67 Structured_dimension_callout to Text_string

Each Structured_dimension_callout has as a dimension value one or more Text_string objects. Each Text_string may be the dimension value for exactly one Structured_dimension_callout.

Each Structured_dimension_callout has as a tolerance value zero, one, or many Text_string objects. Each Text_string may be the tolerance value for exactly one Structured_dimension_callout.

Each Structured_dimension_callout has as unit text zero, one, or many Text_string objects. Each Text_string may be the unit text for exactly one Structured_dimension_callout.

5.1.2.1 Structured_dimension_callout

<mapping of application object omitted>

5.1.2.1.1 structured_dimension_callout to text_string (as dimension value)

<mapping of assertion omitted>

5.1.2.1.2 structured_dimension_callout to text_string (as tolerance value)

<mapping of assertion omitted>

5.1.2.1.3 structured_dimension_callout to text_string (as unit text)

<mapping of assertion omitted>

When an application object maps to different AIM objects in different contexts, often the mappings can be clearly documented only through the use of numbered alternative mapping specifications. In these cases, the application element subclause contains numbered descriptions that indicate when the alternative mappings apply. The complete mapping specification for that alternative follows the numbered description. In practice, when there are multiple mappings for an application object, these mappings are frequently self-explanatory. However, mappings of assertions between objects with multiple mappings generally require the numbered descriptions.

EXAMPLE 2 This is an example of alternative mappings clarified by the use of numbered alternatives.

5.1.x.y Annotation_subfigure_definition_element

#1: If the element is a curve, fill area, symbol, subfigure, or text

AIM element: (draughting_annotation_occurrence)
Source: ISO 10303-201

#2: If the element is a dimension or a draughting callout

AIM element: (draughting_elements)
Source: ISO 10303-201

In the example presented in annex C, the entity data type Organization has multiple mappings, though numbered alternatives are only provided for the mappings of the Organization_name attribute and the assertion from Approval to Organization.

The cardinalities and inheritance documented in the information requirements of clause 4 of the AP are not visible in the format of the mapping specification. Readers of the mapping specification must look to clause 4 of the AP for the cardinalities and subtype/supertype relationships among the application elements. The intent is that the inheritance in the ARM is preserved in the mapping. The mapping of an assertion from a supertype to another application element also applies to subtypes of that supertype.

Application reference models shall not contain complex supertype/subtype structures, and, in particular, shall not contain abstract supertypes. Any subtyping done in the ARM shall reflect domain taxonomies, not the analysis approach applied by the ARM developers. However, ARMs with such complex structures do exist. The following guidance is provided for mapping those ARMs.

- The structure of the mapping specification reflects the structure of the ARM: by default, application elements are mapped based on the position of their declaration in the ARM structure.

- Abstract supertypes are mapped.

- Attributes of abstract supertypes are mapped.

- Subtype and supertype mapping templates (see A.1) may be used to ensure consistency within and ARM subtype/supertype structure.

- Attributes and relationships declared in a supertype will be mapped in subtypes only if the mappings are different in the subtypes. In this case a subtype template (see A.1) shall be used in the mapping specification for the supertype.

- An AP may repeat the mappings of inherited attributes (using a supertype template, A.1) if there is a requirement or desire to do so. This must be done consistently throughout the AP.

6 AIM element documentation

The AIM element contains the description of that to which the application element listed in the subclause heading maps. Application objects map to either a single AIM element, a combination of multiple AIM elements, or a choice among multiple AIM elements. AIM elements for application objects are entity data types or attributes from resource models (i.e., integrated resources or application interpreted constructs) or entity data types that are defined in the AIM of the AP being mapped. The AIM element for an application assertion is populated with the word “PATH” or the words “IDENTICAL MAPPING”. The mapping of an application element may be asserted to be the same as that for a generalized application element (supertype) in the ARM using the predefined SUPERTYPE template (see A.1). Similarly, the mapping of an application element may be asserted to be the same as that for one or more specialized application elements (subtypes) in the ARM using the predefined SUBTYPE template (see A.1). The different types of mappings are described in the following subclauses.

The content of the AIM element is identified during the application interpretation process. This process is based on human understanding of the information requirements presented in the application protocol and the ISO 10303 integrated resources. This process is described in *Guidelines for application interpreted model development.*

The AIM element chosen for the mapping shall be as specific as possible. Mappings shall be shown to the explicit attribute that will be populated with the data rather than to the entity data type itself. Derived attributes shall not be the target of a mapping. When the mapping is to an entity data type, the entity data type name appears as the AIM element. When the mapping is to an attribute, the AIM element is the entity data type name where the attribute is defined, followed by a period, followed by the attribute name.

When a supertype is provided as the AIM element for a mapping, the intent is that the application object may map to the supertype itself or any of the subtypes that are within the scope of the AIM EXPRESS expanded listing as documented in annex A of the application protocol. If the intent is to map to a subset of the subtypes, the AIM element must be specified as a complex mapping (as described in 6.2) that indicates which subtypes are allowable. If the intent is to map the application object to the supertype only, the name of the supertype entity data type is surrounded by vertical bars “| |”.

6.1 Single AIM element

A single AIM element is provided as the AIM element when only one entity data type or attribute from the integrated resources, application interpreted constructs, or AIM short form maps to the application object.

6.2 Multiple AIM elements

Sometimes a single application object maps to more than one AIM element. When this occurs, each mapping is documented in the mapping specification. This may occur as a logical AND (both AIM elements must be present), as a logical XOR (either AIM element may be present), or as a logical OR (one or more of the AIM elements must be present). The AIM elements provided in the mapping shall be as specific as possible. For instance, if the ARM does not separate the unit and value concepts and the data that is to be populated is a combination of a unit and a value, it is more clear if the mapping is to both **measure_with_unit.unit_component** and **measure_with_unit.-value_component** than simply to **measure_with_unit**.

In an "AND" mapping, multiple AIM elements are required to satisfy the information requirement. Square brackets "[ ]" are placed around each required AIM element to denote the "AND" situation. In an "XOR" situation, multiple AIM elements are alternatives that satisfy the information requirements. Parentheses "( )" are placed around each alternative AIM element to denote the "XOR" situation. In an "OR" situation, at least one of the listed AIM elements is required to satisfy the information requirement. Angle brackets "< >" are placed around each alternative in the "OR" situation. Parentheses, square and angle brackets may be used in combination in the specification of an AIM element to indicate the mapping of complex logical situations. The source is provided for each AIM element.

EXAMPLE 1 The mapping in ISO 10303-227 for the Date attribute of the application object Change is to either a date or a date with a time. All three AIM elements are defined in ISO 10303-41.

5.1.x.y.z date	
AIM element:	(calendar_date) ([calendar_date] [local_time])
Source:	ISO 10303-41 ISO 10303-41 ISO 10303-41

The alternatives in complex mappings (typically "XOR" situations) may be numbered to reflect descriptions provided for the application element (see clause 5). These descriptions are provided to clarify the situations under which the mappings apply. If the application of the different mappings is not clear, descriptions shall be provided. For an example of an attribute with multiple mappings, see the Organization_name attribute of the Organization application object in annex C. In this mapping, how the requirement for a name is satisfied depends on whether the organization is satisfied as a person, an organization, or a person within an organization and is thus indicated by parentheses. When a person within an organization is required, the name is satisfied by both the identification of the person and the name of the organization as indicated by the square brackets. The use of numbered alternatives is not mandated, but is left to the discretion of the AP team.

6.3 Path

The AIM element of a mapping specification is “PATH” when the application element is an application assertion and the application objects in the assertion map to different AIM entity data types or to different instances of the same AIM entity data type. The reference path for an application assertion is designed to show how the relationship between the application objects is satisfied in the AIM.

There are cases where one of the application objects participating in an assertion maps to a complex logical relationship of AIM elements and the other maps to a subset of those AIM elements. If the alternatives in the path are numbered, the mapping shall include the descriptions of the alternatives relevant to the mapping. Because the reader of the mapping specification cannot tell whether this was done on purpose or is an omission, when this case occurs, the AIM element for the assertion is still “PATH”, but the mapping shall include a note so it is clear that the reference path is not incomplete. The note shall follow the word “PATH”. The note shall be of the format:

NOTE For the purpose of this mapping, only the subset of the mapping of the <“to”_application_object> specified in the reference path is applicable.

This situation is often an indication that the application reference model needs refinement in this area.

Drawing_sheet to Sheet_placed_annotation assertion is an example of such an assertion (see the example below). The relationship between Drawing_sheet and Sheet_placed_annotation does not apply to case two where the annotation is a dimension or a draughting callout.

EXAMPLE In this example a PATH AIM element requires additional clarification.

5.1.x.y Drawing_sheet

AIM element: dawing_sheet_revision

5.1.x.y.1 drawing_sheet to sheet_placed_annotation

#1: If the annotation is a curve, fill area, symbol, subfigure, or text

AIM element: PATH

NOTE For the purpose of this mapping, only the subset of the mapping of the Sheet_placed_annotation specified in the reference path is applicable.

```
Reference path:    drawing_sheet_revision <=
                   presentation_area <=
                   presentation_representation <=
                   representation
                   representation.items[i] ->
                   representation_item =>
                   styled_item =>
                   annotation_occurrence =>
                   draughting_annotation_occurrence
```

5.1.x.z Sheet_placed_annotation

#1: If the annotation is a curve, fill area, symbol, subfigure, or text

AIM element: (draughting_annotation_occurrence)
Reference path: (draughting_annotation_occurrence <=
annotation_occurrence)

#2: If the annotation is a dimension or a draughting callout

AIM element: (draughting_elements)
Reference path: (draughting_elements <=
draughting_callout)

6.4 Identical mapping

The AIM element of a mapping contains the words "IDENTICAL MAPPING" when the application element is an application assertion and both application objects in the assertion map to the same AIM element instance. In the example below, Annotation_subfigure_definition_element maps to **draughting_annotation_occurrence** or to **draughting_elements**. Draughting_annotation maps to the same two choices. The assertion between these two application objects is an "IDENTICAL MAPPING" because the intent of the mapping is that these two application objects map to the same instance. Such cases may arise when there is a high level of detail in the application reference model (ARM) or the constructs in the integrated resources are more semantically rich than the constructs in the ARM.

EXAMPLE The following is an example of an identical mapping.

5.1.x.y Annotation_subfigure_definition_element

#1: If the element is a curve, fill area, symbol, subfigure, or text

AIM element: (draughting_annotation_occurrence)
Source: ISO 10303-201

#2: If the element is a dimension or a draughting callout

AIM element: (draughting_elements)
Source: ISO 10303-201

5.1.x.y.1 annotation_subfigure_definition_element to draughting_annotation

AIM element: IDENTICAL MAPPING

5.1.x.z Draughting_annotation

#1: If the annotation is a curve, fill area, symbol, subfigure, or text

AIM element: (draughting_annotation_occurrence)
Source: ISO 10303-201

#2: If the annotation is a dimension or a draughting callout

AIM element: (draughting_elements)

Source: ISO 10303-201

When one of the application objects participating in an assertion maps to a complex logical relationship of AIM elements and the other maps to a subset of those AIM elements, the AIM element for the assertion is still IDENTICAL MAPPING, and the mapping may be clarified with a note stating that only the intersection of the AIM elements to which the two objects map are the same instance. The note clarifying the mapping, if it is deemed necessary, shall be included in as many places as it applies. The note shall be of the format:

NOTE For the purpose of this mapping, only the intersection of the mappings for each of the application objects in the assertion is applicable.

7 Source documentation

The source entry contains an ISO standard number and part number for each AIM element provided. The part identifier that is the ISO standard in which the AIM element is defined (e.g. ISO 10303-41). The part numbers referenced in a mapping specification may correspond to an SC4 common resource, an integrated resource; an application interpreted construct; or the application protocol itself, in the case where the AIM element is an AP created specialization of an integrated resource entity data type.

In the previous edition of this standing document, it was possible to list a source other than the part in which the AIM element is defined. That is no longer true. Such constraints on the mapping shall be documented as mapping rules as specified in 9.1.7.

If the AIM element is either "PATH" or "IDENTICAL MAPPING", no source document is listed.

8 Rules documentation

The mapping specification contains references to the global rules in the AIM short form that constrain the mappings. The references are the global rule identifiers as they appear in the AP. The rule reference may include the subclause number where the rule is defined in clause 5 of the AP as well. If so, the subclause references follows the rule identifier in a tail comment where the clause number is in parenthesis.

EXAMPLE The format of the rule reference is as shown below.

Rules: product_definition_has_one_shape -- (See 5.2.5.37)

There may be more than one rule constraining a given mapping. Some mappings may be unconstrained. Rules restricting instantiation of entity data types within the AIM are to be included in the mapping specification only when the mapping is to an AIM element that shall not be independently instantiable. When an entity data type constrained by an instantiability rule appears in a reference path, that mapping assumes that the entity data type will be instantiated in the context of other entity data types; therefore, the reference to the rule is not needed for that mapping. All rules that are created in the AP short form, including entity data type instantiability rules, shall be referenced in the mapping specification at least once.

Each AP contains a global rule constraining **application_context**. This rule shall be referenced in any mappings to the following AIM elements: **product**, **product_context**, **product_definition**, and **library_assignment**.

9 Reference path documentation

The reference path illustrates how the requirements and relationships stated in clause 4 of the AP are maintained as a result of the application interpretation process. It specifies the complete path of entity data type references in the AIM that is needed to represent the information requirements of the ARM. A set of symbols and formats were developed to construct a consistent syntax for documenting reference paths. Reference path syntax is consistent for each type of application element. The intent of the current syntax is to facilitate human readability of the mapping specification. In the future, this syntax may be extended to improve computer readability of the mapping specification. This clause discusses the symbology used in documenting reference paths as well as reference path requirements for each type of application element.

9.1 Symbology

A reference path specification is read from left to right and from top to bottom. Each line of the reference path specification may contain symbols to illustrate the EXPRESS structure of the AIM objects. Understanding the symbology used in the reference path specification is the key to reading and writing mapping specifications.

9.1.1 Delimiter symbols

The delimiter symbols are used to indicate the relationship of the specified entity data type or attribute preceding the symbol to the specified entity data type or attribute following the symbol. The symbol should be placed at the end of the line, so that the name of the following entity data type or attribute is at the beginning of the next line. The delimiters should be separated from the entity data type or attribute text by a single space, for readability. The meaning of these symbols can be paraphrased as:

=>	“is a supertype of”
<=	“is a subtype of”
->	“references”
<-	“is referenced by”

The “=>” and “<=” symbols indicate a supertype or subtype structure. The “=>” symbol is used to indicate that the specified entity data type preceding the symbol is the supertype of the entity data type specified on the next line. The “<=” symbol is used to indicate that the specified entity data type preceding the symbol is a subtype of the entity data type specified on the next line.

The “->” and “<-” symbols indicate the reference to an entity data type or defined data type by an attribute. The “->” symbol is used to indicate that the specified attribute preceding the symbol references the entity data type or defined data type specified on the next line. The “<-” symbol is used to indicate that the specified entity data type or defined data type preceding the symbol is referenced by the attribute specified on the next line.

When an entity data type name appears on a line that is terminated without the use of one of the above delimiter symbols, this may indicate that the specified entity data type has the attribute shown on the next line. When an attribute name appears on a line that is terminated without the use of one of the above symbols, this may indicate that the specified attribute is an attribute of the entity data type shown on the next line. A new line is used without conveying additional semantics before and after lines where an EXPRESS select type value is provided (see 9.1.3). New lines precede and follow mapping rules (braces), and the individual options for mappings with multiple AIM elements (parentheses, square and angle brackets). A new line also terminates the reference path.

Reference path statements that are too long to fit on one line can be split using the forward slash symbol "\". The symbol conveys no additional meaning within the reference path. The "\" is positioned between elements of the statement, at the end of the line, preferably in white space. The "\" may appear between an entity data type and an attribute, following the period, but this case should be avoided where possible. The "\" should not be placed inside of a text string; the entire text string should follow the forward slash on the next line.

9.1.2 Aggregation symbols

If an attribute references an aggregate cardinality, and any single instance of the aggregate is of interest, brackets and the letter i "[i]" are used to indicate this. This reflects the usual requirement that the path can go through any member of the aggregate. The use of "[n]" (where n is an integer) indicates that member n of the aggregate is of interest in the mapping. A reference to an_entity.aggregate[1] reflects the requirement that the path must go through the first element of the aggregate. In order to limit the number of elements in an instantiation of the aggregation, EXPRESS rules shall be written in the short form of the AIM. In the mapping of the Drawing to Approval assertion in annex C, the attribute **approved_items** is a set that references the select type **approved_item**.

9.1.3 Equal sign

An equal sign "=" is used in the reference path specification to indicate a member of the select list of an EXPRESS select type, an item from the enumerated list of an EXPRESS enumeration type, a specific value for an attribute, or a specific type is being assigned to an item of an aggregate. In the case of a select list, the name of the select type appears first followed on the same line by the equal sign and the member that is being selected. In the case of an enumerated list, the name of the enumeration type appears first followed on the same line by the equal sign and the enumerated item. In the case of a specific value, the attribute name appears first followed on the same line by the equal sign and the value assigned to the attribute. In the case where a specific type is being assigned to an item of an aggregate, the aggregate reference appears first followed on the same line by the equal sign and the specified item.

See the mapping of the Date attribute of the Approval object in annex C. In this reference path specification, the attribute **date_time** references the select type **date_time_select**. For this mapping, the selection is a **date**. As seen in this example, the name of the select type appears on the line before the line containing the equal sign, and the selected member appears on the line following the line containing the equal sign. The order of these lines is reversed for a reference path in which the selected member is referenced first in the path. See the mapping of the Drawing to Approval assertion in the example in annex C. In this mapping, the reference path encounters **drawing_revision** first, which is the selected member of the **approved_item** select type. In the following example, an item of an aggregate is being specified to be of a certain type; the **representation_item** must be a **measure_representation_item**.

EXAMPLE The EXPRESS defined data types are used in a reference path below to illustrate assigning a specific type to an item of an aggregate.

```
ENTITY compound_representation_item
  SUBTYPE OF (representation_item);
     item_element : compound_item_definition;
END_ENTITY;

TYPE compound_item_definition = SELECT
    (set_representation_item,
     list_representation_item);
END_TYPE;

TYPE set_representation_item = SET[1:?] OF representation_item;
END_TYPE;

TYPE list_representation_item = LIST[1:?] OF representation_item;
END_TYPE;
```

Reference path: {compound_representation_item
compound_representation_item.item_element ->
compound_item_definition
compound_item_definition = list_representation_item
list_representation_item
list_representation_item[i] = representation_item
representation_item =>
measure_representation_item}

9.1.4 Parentheses

Parentheses "( )" are used to indicate the existence of options in the reference path; a logical XOR. Each option is enclosed by a set of parentheses. The parentheses are used to indicate that a mapping has multiple reference paths or sections of the reference path. There are two reasons that the reference path may diverge: an object is mapped to multiple AIM entity data types or the reference path depends on the instantiation of the AIM. To aid understanding, the optional sections of the path may be numbered and a description provided, with the application object, that gives the reason for the divergence.

When an ARM object maps to an entity data type from the integrated resources, no reference path is typically required. In the case where an ARM object maps to an "XOR" of an entity data type from the integrated resources and an entity data type defined in the AP, the entity data type from the integrated resources shall be enclosed in parenthesis in the reference path for clarity.

EXAMPLE Fragments of a mapping specification that illustrate the "XOR" case.

AIM element: (product_definition)
(externally_defined_product_definition)
Reference path: (product_definition)
(externally_defined_product_definition <= [product_definition]
[externally_defined_item])

See the mapping of the Approval to Organization assertion in annex C. In this example, it is necessary to show the reference paths for each of the AIM elements to which the Organization maps.

9.1.5 Square brackets

Square brackets “[ ]” are used to indicate two or more required sections of the reference path; a logical AND. The square brackets indicate that there are either multiple mappings or multiple paths required to satisfy the mapping. Each mapping or path is enclosed by a set of square brackets. To fully document how the requirements are satisfied by the mapping, sections of the path may be numbered and a description giving the reason for the divergence may be provided immediately following the subclause heading.

When an ARM object maps to an entity data type from the integrated resources, no reference path is typically required. In the case where an ARM object maps to an “AND” of an entity data type from the integrated resources and an entity data type defined in the AP, the entity data type from the integrated resources shall be enclosed in square brackets in the reference path for clarity.

See the mapping of the Annotation_subfigure_definition to 2D_cartesian_coordinate_space assertion in the example below. This example shows that every **representation_context** that satisfies the requirements of the application object **2D_cartesian_coordinate_space** must be both a **geometric_-representation_context** and a **global_unit_assigned_context**.

EXAMPLE An example of a mapping that illustrate the “AND” case.

5.1.x.y Annotation_subfigure_definition

AIM element:	symbol_representation_map
Source:	ISO 10303-46

Reference path:	{symbol_representation_map <=
	representation_map
	representation_map.mapped_representation->
	representation =>
	symbol_representation =>
	draughting_subfigure_representation}

5.1.x.y.1 annotation_subfigure_definition to 2d_cartesian_coordinate_space

AIM element:	PATH
Reference path:	symbol_representation_map <= representation_map representation_map.mapped_representation -> representation {representation => symbol_representation => draughting_subfigure_representation} representation.context_of_items -> representation_context => [geometric_representation_context] [global_unit_assigned_context]

5.1.x.z 2D_cartesian_coordinate_space

AIM element:	[geometric_representation_context] [global_unit_assigned_context]
Source:	ISO 10303-42

See the mapping of the Organization_name attribute of Organization in annex C. Where the mapping is to a person within an organization, both reference paths shown are required.

9.1.6 Angle brackets

Angle brackets "< >" are placed around elements of a reference path to indicate that at least one of the sections of the reference path enclosed by the angle brackets are required; a logical OR. The angle brackets indicate that there are either multiple mappings or multiple paths required to satisfy the mapping. To fully document how the requirements are satisfied by the mapping, sections of the path may be numbered and a description giving the reason for the divergence may be provided immediately following the subclause heading.

When an ARM object maps to an entity data type from the integrated resources, no reference path is typically required. In the case where an ARM object maps to an "OR" of an entity data type from the integrated resources and an entity data type defined in the AP, the entity data type from the integrated resources shall be enclosed in angle brackets in the reference path for clarity.

Mappings and reference paths that require the use of angle brackets are an indication that the application element being mapped represents multiple concepts. In these cases, the application element should be reviewed and the creation of a separate object for each concept should be considered.

NOTE While this syntax is retained in this version of this document, its use is deprecated.

9.1.7 Braces

Braces "{ }" are used to indicate constraints placed on the mapping. The constraints documented within the braces are commonly referred to as mapping rules. Mapping rules constrain the set of valid data populations. They apply only to the mapping of the application element being mapped in that subclause of the mapping specification. Because of this limited scope, it may not be possible to also write these constraints in the AIM EXPRESS as either global or local rules. Where possible, EXPRESS constraints should be provided in the AIM to document mapping rule constraints.

NOTE 9.2 of ISO 10303-11:1994 enumerates the type of constraints that can be written for EXPRESS entity data types.

Mapping rules are used when the reference path specification would not normally contain AIM entity data types which are crucial to the mapping. Mapping rules may be used to include into the reference path specification required supertypes or subtypes, required values assigned to attributes, or AIC entity data types containing rules that constrain the mapping.

Mapping rules shall start on a new line of a reference path and shall end with a new line. The reference path shall be completely specified without the mapping rule; if the mapping rule were removed, the reference path would still be correct. If a reference path contains multiple mapping rules, all the rules apply to the mapping. If the granularity of the ARM and AIM differ such that the resulting mapping rules apply to different cases, provide numbered alternatives immediately following the subclause heading and number the mapping rules correspondingly in the reference path.

Mapping rules are also used when an assertion is mapped to a specialization of a resource entity data type and the reference path specification would not otherwise show the resource entity data type. The inclusion of these AIM elements is intended to satisfy the requirement that all mappings to a specialized entity data type shall have a reference path to a resource entity data type.

In this example, the reference path specification would not normally show the ISO 10303-201 created subtype. However, for this mapping, the only **draughting_callout_relationship** that satisfies the information requirement is the **dimension_callout_component_relationship** subtype. The second mapping rule indicates that the name of the relationship is restricted for this mapping to have the value “prefix”.

EXAMPLE The mapping of the Structured_dimension_callout to Draughting_callout assertion contains two mapping rules.

5.1.x.y Structured_dimension_callout

AIM element:	structured_dimension_callout
Source:	ISO 10303-201
Reference path:	structured_dimension_callout <= draughting_callout

5.1.x.y.1 structured_dimension_callout to draughting_callout (as prefix)

AIM element:	PATH
Reference path:	structured_dimension_callout <- draughting_callout_relationship.relating_dimension_callout draughting_callout_relationship {draughting_callout_relationship <= dimension_callout_component_relationship} {draughting_callout_relationship.name = 'prefix'} draughting_callout_relationship.related_draughting_callout -> draughting_callout

5.1.x.z Draughting_callout

AIM element:	draughting_callout
Source:	ISO 10303-101

In the case where you are mapping to an entity that may be used in multiple different contexts within the AIM, the contexts may be distinguished through the use of a mapping rule.

EXAMPLE 2 This case was handled in the first edition of this document by specifying the AIC as the source for the second case. This is the current recommendation for how to document such a mapping.

5.1.x.y Point

#1: the point is not part of a shape representation

AIM element:	point
Source:	10303-42
Reference path:	point {point <= geometric_representation_item <=

```
                    representation_item <-
                    representation.items[i]
                    representation}

#2: the point is part of an elementary brep

AIM element:        point
Source:             10303-42
Reference path:     point
                    {point <=
                    geometric_representation_item <=
                    representation_item <-
                    representation.items[i]
                    representation =>
                    shape_representation =>
                    elementary_brep_shape_representation}
```

9.1.8 Asterisk

An asterisk "*" is used in conjunction with braces to indicate that any number of relationship entity data types may be assembled in a relationship tree structure. This is shown by the use of the repeated section of the reference path enclosed in braces and followed by an asterisk. The start and end of the path segment being repeated must be instances of the same entity data type. An asterisk may only be used with the mapping rule construct, and the only currently known application of this syntax is in extending relationship trees to any number of levels. In the absence of a mapping rule including an asterisk, the path contains only the number of relationship entity data types shown. In the reference path in the example, below, the reference path shows a choice between one relationship (in first set of parenthesis) or two or more relationships (in the second set of parenthesis) in the relationship tree.

EXAMPLE A mapping rule using the * operator.

```
(draughting_callout <-
draughting_callout_relationship.related_draughting_callout
draughting_callout_relationship
draughting_callout_relationship.relating_draughting_callout ->
draughting_callout)
({draughting_callout <-
draughting_callout_relationship.related_draughting_callout
draughting_callout_relationship
draughting_callout_relationship.relating_draughting_callout ->
draughting_callout
draughting_callout <-
draughting_callout_relationship.related_draughting_callout
draughting_callout_relationship
draughting_callout_relationship.relating_draughting_callout ->
draughting_callout}*)
```

NOTE There is an issue logged against this syntax element. No end condition is provided in the syntax, therefore this operator may not be automatable. This issue is under investigation. See ISO TC 184/SC4 QC N135, issue 18.

9.1.9 Vertical bars

Vertical bars “| |” are used to indicate that the mapping is only to the supertype entity data type found between the vertical bars. If the vertical bars are not present, and the entity data type being referenced is a supertype, it is assumed that any of the subtypes of that supertype are valid in the context of that reference path. The vertical bars limit the reference path to the supertype entity data type itself. Vertical bars may appear in both the AIM element and the reference path.

EXAMPLE Use of vertical bar notation.

```
A.b ->
| only_supertype_allowed |
```

9.1.10 Single quotes

Single quotes “ ‘ ’ “ are used to indicate that the enclosed text is intended to be the value of an attribute whose underlying type is a string. Single quotes are used in this context to provide consistency with ISO 10303-11. They may appear in the reference path only.

9.2 Application objects

A reference path specification is necessary for each application object that is mapped to a specialization of an integrated resource entity data type. The reference path starts with the AIM element to which the application object is mapped. It concludes at the integrated resource entity data type of which the specialization is a subtype. See the mapping of Drawing in annex C. In this example, Drawing is mapped to **draughting_drawing_revision**, which is a subtype of **drawing_revision** in ISO 10303-101. The reference path contains these two entity data types.

A reference path can also be shown to clarify a restriction on the mapping. See the mapping of Annotation_subfigure_definition in the example in 9.1.5. This object maps to an entity data type in the integrated resources; therefore, no reference path specification is required. However, to satisfy the requirements for this mapping, the inherited attribute **mapped_representation** must be of type **draughting_subfigure_representation**. This restriction is shown by including this portion of the reference path specification within braces (see 9.1.7).

In the case where a mapping has options and/or multiple required portions where one of the selections is an entity data type from the integrated resources, the entity data type from the integrated resources shall also appear in the reference path to completely state the options in the path. See the example in 9.1.4.

9.3 Application attributes

There are different mappings of an attribute that must be considered for the documentation of the reference path. An attribute may be mapped to an attribute of the same AIM entity data type to which the application object is mapped. In this case, no reference path is necessary for the attribute. See the mapping of the Description attribute of Approval in the example in annex C.

An attribute may be mapped to an attribute of an entity data type different from the one to which the application object is mapped. The reference path for the attribute starts with the AIM element to which the application object is mapped. The path follows the entity data types and attributes of the AIM to the AIM attribute to which the application attribute is mapped.

See the mapping of the Drawing_number attribute of Drawing in the example in annex C. In this example, Drawing is mapped to **draughting_drawing_revision** so the reference path of Drawing_number begins with this entity data type. **Drawing_revision** has an attribute **drawing_identifier** that references the entity data type **drawing_definition**. The **drawing_definition** entity data type has an attribute **drawing_-number** to which the Drawing_number attribute is being mapped.

9.4 Application assertions

Application assertions specify the relationships between pairs of application objects, the cardinality of the relationships, and the rules required for the integrity and validity of the application objects.

The reference path of an assertion starts with the AIM element to which the first application object is mapped. The path concludes at the AIM element to which the second application object in the assertion is mapped. See the mapping of the Drawing to Approval assertion in annex C. In this example, Drawing is mapped to **draughting_drawing_revision** so the reference path of the assertion begins with this entity data type. **Drawing_revision** is one of the **approved_items** to which **approval** is assigned.

If the AIM element is “IDENTICAL MAPPING”, no reference path specification is required.

In rare cases, an application assertion may map to an AIM entity data type or attribute. In these cases, the mapping may be to an attribute in the reference path that connects the two identified application objects, or to an AIM entity data type that acts as the intersection entity data type connecting the two identified application objects. This attribute or entity data type is selected as the AIM element with agreement by AIM interpretation experts. The source is the number of the part containing this entity data type or attribute.

9.5 AIC considerations

AICs define entity data types where the global rules pertaining to that AIC have been localized. These entity data types are called “root node” entity data types. If the root node of the AIC includes restrictions that apply to the mapping, the root node shall be included in the reference path specification to indicate this, even if the inclusion is only within a mapping rule.

Annex A
(normative)

Mapping templates

NOTE Mapping templates are a new concept, introduced in this edition of this document. Mapping templates are experimental, the use of mapping templates is optional, and the mapping template language may be modified in the future as the concept is tested and matures.

A mapping specification may make use of mapping templates. A mapping template is a reusable portion of a reference path that defines a commonly used part of the structure of the application interpreted model. Mapping templates help to reduce the size of a mapping specification. In allowing for the reuse of both mappings and reference paths, mapping templates contribute to consistency and quality of a mapping specification. Mapping templates also facilitate maintenance of a mapping specification.

A mapping template is similar to a programming language macro. Each mapping template definition has three components, as follows:

— the template signature, that specifies the name of the template and may also specify the names and the order of the formal parameters of the template;

— description of the formal parameters of the template, if any; and

— the template body, that defines the reusable portion of a reference path and may indicate, through the use of the formal parameter names included in the template signature, the points at which the value parameters are supplied in each template application.

When a formal parameter is intended to be a string, the template definition shall include that information in the definition of the formal parameter.

Mapping templates that are predefined, such as the subtype and supertype reference templates described in A.1 below, may not follow this structure completely; they may not have a template body.

Each mapping template documented in the AP shall be used at least once in the reference paths specified in the AP. Each such template application is a reference to the template definition, based on the pattern established by the template signature, and supplies the value parameters that are to be substituted for the formal parameters specified in the template definitions. The full reference path can be derived by replacing any formal parameters in the template body by the value parameters specified in the template application and then substituting the completed template body for the template application.

EXAMPLE 1 The following is an example of a template application that invokes and supplies parameters for the GROUPS mapping template.

```
/GROUPS(shape_aspect, 'boundary index 2')/
```

The non-blank characters following the first "/" define the name of the mapping template. The name of the mapping template shall be written in uppercase. It shall not begin with a number; otherwise numbers may be included. No special characters shall be used except for underscore ("_"). The name of the template shall be followed by the list of value parameters, enclosed in parentheses. Value parameters shall be written in lowercase except in the case that the value parameter is a string literal that includes upper case letters. In general, identifiers and literals in templates, definitions and calls, shall only include characters from the EXPRESS character set that are allowed according to 7.4 and 7.5 of ISO 10303-11:1994.

The following notational conventions apply to the definitions and applications of mapping templates:

/	marks the beginning and the end of a template signature or a template application;
&	prefixes the name of a formal parameter within the definition of a template body;
( )	enclose the formal parameters in a template signature or the value parameters in a template application;
,	separates formal parameters in a template signature or value parameters in a template application;
' '	enclose a string literal that is used as a value parameter in a template application.

Value parameters are either EXPRESS data type identifiers or string literals. Value parameters that are not enclosed by quotes are EXPRESS data type identifiers.

The template bodies of mapping templates are documented in 5.1.1 of the APs that use this concept. See *SC4 Supplementary directives - Rules for the structure drafting of SC4 standards for industrial data* for details on the document structure, presentation style and required text for mapping templates in APs.

EXAMPLE 2 A mapping template definition from 5.1.1 of an AP.

5.1.1.1 APPROVES

The APPROVES mapping template specifies a reference path constraint in which instances of type entity are the **approval_item**s within instances of **applied_approval_assignment**, where the role name spedicied for the assignment is 'arm role'.

Mapping signature:

/APPROVES(entity,arm_role)/

Parameter definitions:

entity: the identifier of the AIM entity data type to which an approval is assigned

arm_role: a string containing the constrained role that is specified for the approval assignment

Template body:

```
approval_assignment.assigned_approval
approval_assignment
{approval_assignment.role ->
object_role
[object_role.name = &arm_role]
[object_role.description = '.UNUSED.']}=>
applied_approval_assignment
```

```
applied_approval_assignment.items[i] ->
approval_item
approval_item = & entity
```

EXAMPLE 3 The following application of the APPROVES mapping template states that an approval is assigned to a **product_definition_shape** in the role of 'subject'.

```
approval <-
/APPROVES(product_definition_shape,'subject')/
```

EXAMPLE 4 Another mapping template definition from 5.1.1 of an AP.

5.1.1.2 PERS_ORG_ASSGN

The PERS_ORG_ASSGN mapping template specifies a reference path constraint in which instances of type **entity** are the **approval_item**s within instances of **applied_approval_assignment**, where the role name specified for the assignment is 'arm role'.

Mapping signature:

/PERS_ORG_ASSGN(t,role)/

Parameter definitions:

t: the identifier of the AIM entity data type to which a **person_and_organization** is assigned

role: a string containing the constrained role of the person in the context of this assignment

Template body:

```
person_and_organization_item = &t
person_and_organization_item <-
applied_person_and_organization_assignment.items[i]
applied_person_and_organization_assignment <=
person_and_organization_assignment
{person_and_organization_assignment.role ->
person_and_organization_role
person_and_organization_role.name = &role}
person_and_organization_assignment
person_and_organization_assignment.assigned_person_and_organization
person_and_organization
```

EXAMPLE 5 The following application of the PERS_ORG_ASSGN mapping template states that a **person_-and_organization_assignment** is applied to a **document** entity data type to identify the author.

```
/PERS_ORG_ASSGN(document,'author')/
```

A mapping template may use other templates in its template body. The recursive use of templates is not foreseen.

A.1 Subtype and supertype cross references

When an ARM is specified as a data model that includes inheritance trees, it may happen that entity data types with an inheritance relationship map to the same AIM concepts.

NOTE 1 The Quality Committee strongly discourages ARMs that are fully attributed data models.

Thus, mappings on lower levels of the inheritance trees may be identical to mappings higher up. Also, the mapping of supertypes may in some cases be merely the one or the sum of the mappings of its subtypes. To allow a mapping specification show these types of identical mappings the following two predefined mapping templates shall be used:

— /SUBTYPE(<subtype_of_application_object>)/ -- (See <subclause containing subtype mapping>);

— /SUPERTYPE(<supertype_of_application_object>)/ -- (See <subclause containing supertype mapping>).

The parameter of each of the templates shall be replaced by the name of the referenced application object. The comment contains a reference to the subclause of the mapping of the referenced application object.

NOTE 2 The mapping templates SUBTYPE and SUPERTYPE are further documented in *SC4 Supplementary directives - Rules for the structure drafting of SC4 standards for industrial data*.

For application objects, their attributes, and application assertions, these templates may appear as the AIM element. For an application object the template replaces a common resource concept, such as an entity data type from the integrated resources. For an application assertion it replaces PATH or IDENTICAL MAPPING. If subtype or supertype cross referencing is applied, the remainder of the mapping specification that normally follows the AIM element subclause shall be omitted. The mapping specification documented under the referenced subtype or supertype is valid instead.

EXAMPLE 1 This example illustrates subclause referencing among application objects. The AIM element for this application object is the same as for its subtype, i.e., in this example the application object chamfer. Chamfer is mapped in subclause 5.1.x.y, with x denoting the UoF and y the application object within the UoF.

5.1.2. Structural_features UoF

5.1.2.4 Feature

AIM element: /SUBTYPE(chamfer)/ -- (See 5.1.x.y)

One AIM element may contain several references to subtypes or supertypes. The AND, XOR and OR operators of the mapping notation described in 9.1.3, 9.1.4, 9.1.5 shall be applied to indicate the relationships between the various mappings.

EXAMPLE 2 An AIM element may map according to which subtype it is instantiated as.

AIM element: (/SUBTYPE(corner_cutout)/ -- (See 5.1.5.2))
(/SUBTYPE(interior_cutout)/ -- (See 5.1.7.6))

A.2 Templates in reference paths

The reference paths of a mapping specification often repeat themselves. This is partly because of the limited size of the integrated resources. Also the existence of an inheritance tree in the ARM leads to repetition. The mapping templates that are introduced in annex A may be used to avoid such repetition. A piece of a reference path that appears several times within the mapping specification of an AP may be transformed into a mapping template. The bit of the reference path that shall be removed becomes the template body; this shall be documented in 5.1.1 of the AP. It will be replaced in the reference path by the template application. Templates may use parameters to increase the reusability of the mapping template.

The following examples show reference paths without and with mapping templates:

EXAMPLE 1 The reference path for the attribute **boundary_index_2** (5.1.2.1.3) does not use mapping templates:

5.1.2 Structural_features UoF

5.1.2.1 Corner_cutout_boundary_relationship

AIM element:	group
Source:	ISO 10303-41
Rules:	shape_aspect_for_border_for_corner_cutout_boundary_relationship -- (See 5.2.5.309) shape_aspect_for_corner_cutout_for_corner_cutout_boundary_relationship -- (See 5.2.5.313)

5.1.2.1.3 boundary_index_2

AIM element:	shape_aspect
Source:	ISO 10303-41
Reference path:	group <- group_assignment.assigned_group group_assignment => {group_assignment.role -> object_role [object_role.name = 'boundary index 2'] [object_role.description = '.UNUSED.']} applied_group_assignment applied_group_assignment.items[i] -> group_item group_item = shape_aspect shape_aspect

5.1.2.2 Feature

AIM element:	/SUBTYPE(chamfer)/ -- (See 5.1.x.y)

EXAMPLE 2 A mapping template definition from 5.1.1 of an AP.

5.1.1.1 GROUPS

The GROUPS mapping template specifies a reference path constraint in which instances of type entity are the **approval_item**s within instances of **applied_group_assignment**, where the role name spicied for the assignment is 'arm role'.

Mapping signature:

/GROUPS(grouped_entity,arm_role)/

Parameter definitions:

grouped_entity: the data type of the instances that are collected as grouped items by a group assignment

arm_role: a string containing the constrained role that is specified for the group assignment

Template body:

```
group_assignment.assigned_group
group_assignment
{group_assignment.role ->
object_role
[object_role.name = &arm_role]
[object_role.description = '.UNUSED.']}
applied_group_assignment
applied_group_assignment.items[i] ->
group_item
group_item = &grouped_entity
```

EXAMPLE 3 An equivalent mapping specification to Example 1, above, using the mapping template defined in Example 2, above:

5.1.2 Structural_features UoF

5.1.2.1 Corner_cutout_boundary_relationship

AIM element:	group
Source:	ISO 10303-41
Rules:	shape_aspect_for_border_for_corner_cutout_boundary_relationship -- (See 5.2.5.309) shape_aspect_for_corner_cutout_for_corner_cutout_boundary_relationship -- (See 5.2.5.313)

5.1.2.1.3 boundary_index_2

AIM element:	shape_aspect
Source:	ISO 10303-41
Reference path:	group <- /GROUPS(shape_aspect, 'boundary index 2')/ shape_aspect

5.1.2.2 Feature

AIM element:	/SUBTYPE(chamfer)/ -- (See 5.1.x.y)

Annex B
(informative)

Example information requirements

This annex contains example descriptions of a UoF, application objects, and application assertions that appear in the example mapping specification in annex C. These descriptions are extracted from clause 4 of ISO 10303-201. Some objects, attributes, and assertions from the UoF have been excluded from the example.

NOTE The numbering in this annex reflects the clause numbering as published in ISO 10303-201.

4.1 Units of functionality

4.1.4 drawing_structure_and_administration

The drawing_structure_and_administration UoF contains information about the hierarchical organization of drawings, drawing sheets, and drawing views, together with the administrative information necessary to manage drawings and drawing sheets. Drawing sheets and drawing views are defined in their specific coordinate space. Annotation may be assigned to each drawing sheet and drawing view. The administrative information supports the exchange of drawings between environments in which configuration management of drawings is used. The following application objects are used by drawing_-structure_and_administration UoF:

- Approval;
- Drawing;
- Drawing_sheet;
- Organization.

4.2 Application objects

4.2.13 Approval

An Approval is information that indicates a drawing, drawing sheet, or both have been reviewed for data content and for correctness of the presentation of that data and has been found to be acceptable. The data associated with an Approval are the following:

- Date;
- Description.

4.2.19.1 Date

The Date specifies the date on which the approval was assigned.

4.2.19.2 Description

The Description specifies the organization-specific release status or the authorized modifications for the revision of the drawing, drawing sheet, or both.

4.2.30 Drawing

A Drawing is the presentation of product data in a human-interpretable form wherein the physical and functional requirements for that product are presented pictorially and textually. The data associated with a Drawing are the following:

— Drawing_number;

— Drawing_revision_id.

4.2.30.2 Drawing_number

The Drawing_number specifies the identification of a particular drawing by an organization.

4.2.30.3 Drawing_revision_id

The Drawing_revision_id specifies the identification of a particular version of the drawing.

4.2.31 Drawing_sheet

A Drawing_sheet is a logical division of a drawing into a two-dimensional area for the presentation of product data. These divisions correspond to sheet paper sizes for plotting. A Drawing_sheet contains at least one Drawing_view or one Draughting_annotation. The data associated with a Drawing_sheet are the following:

— Sheet_number;

— Sheet_revision_id.

4.2.31.2 Sheet_number

The Sheet_number specifies the page number for a particular drawing sheet and its location in relation to other sheets of the drawing.

4.2.31.3 Sheet_revision_id

The Sheet_revision_id specifies the identification of a particular version of the drawing sheet.

4.2.57 Organization

An Organization is a number of persons or groups that designs, produces and supplies products and services. The data associated with an Organization are the following:

— Organization_name.

4.2.57.2 Organization_name

The Organization_name specifies the identification of a particular organization.

4.3 Application assertions

4.3.18 Approval to Organization

Each Approval is provided by one or more Organization objects. Each Organization provides zero, one, or many Approval objects.

4.3.32 Drawing to Approval

Each Drawing is governed by zero, one, or many Approval objects. Each Approval governs zero or one Drawing.

4.3.33 Drawing to Drawing_sheet

Each Drawing consists of one or more Drawing_sheet objects. Each Drawing_sheet belongs to exactly one Drawing.

4.3.37 Drawing_sheet to Approval

Each Drawing_sheet is governed by zero, one, or many Approval objects. Each Approval governs zero, one, or many Drawing_sheet objects.

Annex C
(informative)

Example mapping specification

This annex contains the mapping specification that corresponds to the example information requirements in annex B. See annex B for the textual descriptions of the application objects.

The AIM entity data types found in this mapping specification are defined in ISO 10303-41 [4], ISO 10303-101 and ISO 10303-201.

5.1.4 Drawing_structure_and_administration UoF

5.1.4.1 Approval

AIM element: approval
Source: ISO 10303-41

5.1.4.1.1 date

```
AIM element:     calendar_date
Source:          ISO 10303-41
Reference path:  approval <-
                 approval_date_time.dated_approval
                 approval_date_time
                 approval_date_time.date_time ->
                 date_time_select
                 date_time_select = date
                 date =>
                 calendar_date
```

5.1.4.1.2 description

AIM element: approval.level
Source: ISO 10303-41

5.1.4.1.3 approval to organization

#1: If the approval is given by only a person

```
AIM element:     PATH
Reference path:  (approval <-
                 approval_person_organization.authorized_approval
                 approval_person_organization
                 approval_person_organization.person_organization ->
                 person_organization_select
        person_organization_select = person
        person)
```

#2: If the approval is given by only an organization

AIM element: PATH
Reference path: (approval <-
approval_person_organization.authorized_approval
approval_person_organization
approval_person_organization.person_organization ->
person_organization_select
person_organization_select = organization
organization)

#3: If the approval is given by a person within an organization

AIM element: PATH
Reference path: (approval <-
approval_person_organization.authorized_approval
approval_person_organization
approval_person_organization.person_organization ->
person_organization_select
person_organization_select = person_and_organization
person_and_organization)

5.1.4.2 Drawing

AIM element: draughting_drawing_revision
Source: ISO 10303-201
Reference path: draughting_drawing_revision <=
drawing_revision

5.1.4.2.1 drawing_number

AIM element: drawing_definition.drawing_number
Source: ISO 10303-101
Reference path: draughting_drawing_revision <=
drawing_revision
drawing_revision.drawing_identifier ->
drawing_definition
drawing_definition.drawing_number

5.1.4.2.2 drawing_revision_id

AIM element: drawing_revision.revision_identifier
Source: ISO 10303-101
Reference path: draughting_drawing_revision <=
drawing_revision
drawing_revision.revision_identifier

5.1.4.2.3 drawing to approval

AIM element: PATH
Reference path: draughting_drawing_revision <=

drawing_revision
approved_item = drawing_revision
approved_item <-
draughting_approval_assignment.approved_items[i]
draughting_approval_assignment <=
approval_assignment
approval_assignment.assigned_approval ->
approval

5.1.4.2.4 drawing to drawing_sheet

AIM element: PATH
Reference path: draughting_drawing_revision <=
drawing_revision <=
presentation_set <-
area_in_set.in_set
area_in_set
{area_in_set =>
drawing_sheet_revision_usage}
area_in_set.area ->
presentation_area =>
drawing_sheet_revision

5.1.4.3 Drawing_sheet

AIM element: drawing_sheet_revision
Source: ISO 10303-101

5.1.4.3.1 sheet_number

AIM element: drawing_sheet_revision_usage.sheet_number
Source: ISO 10303-101
Reference path: drawing_sheet_revision <=
presentation_area <-
area_in_set.area
area_in_set =>
drawing_sheet_revision_usage
drawing_sheet_revision_usage.sheet_number

5.1.4.3.2 sheet_revision_id

AIM element: drawing_sheet_revision.revision_identifier
Source: ISO 10303-101

5.1.4.3.3 drawing_sheet to approval

AIM element: PATH
Reference path: drawing_sheet_revision
approved_item = drawing_sheet_revision
approved_item <-
draughting_approval_assignment.approved_items[i]

	draughting_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval

5.1.4.4 Organization

AIM element:	(person) (organization) (person_and_organization)
Source:	ISO 10303-41 ISO 10303-41 ISO 10303-41

5.1.4.4.1 organization_name

#1: If the organization is only a person

AIM element:	(person.id)
Source:	ISO 10303-41

#2: If the organization is only an organization

AIM element:	(organization.name)
Source:	ISO 10303-41

#3: If the organization is a person within an organization

AIM element:	([person.id] [organization.name])
Source:	ISO 10303-41 ISO 10303-41
Reference path:	(person_and_organization [person_and_organization.the_person -> person person.id] [person_and_organization.the_organization -> organization organization.name])

Annex D
(informative)

Mapping specification syntax

NOTE This formal mapping syntax is a new concept, introduced in this edition of this document. This syntax is experimental and may not match the guidelines set forth in the body of this document. Its use optional. The formal syntax may be modified in the future as the concept is tested and matures. For a description of the requirements for the syntax, instructions on its use and limitations, see [7] and [8].

This annex describes a formal syntax for mapping specifications that is intended to match the requirements set forth in the body of this document. An AP project wishing to provide a computer-interpretable representation of their mapping specification may use the syntax provided in this annex. They may include the resulting specification as an informative, computer-interpretable annex to the AP.

This annex defines the lexical elements of the mapping specification syntax and the grammar rules that these elements shall obey.

D.1 Grammar rules for mapping syntax

White space (white space, tab symbol or new line symbol) and/or remark(s) may appear between any two tokens in the rules defined below. To aid readability, the grammar rules are organized according to mapping syntax elements.

```
schema  schema_mapping = "schema_mapping" schema_mapping_name "("
        target_schema_name "," source_schema_name ")" ";" {include}
        ({{uof_mapping} {schema_template}} | {{entity_mapping}
        {uof_template}})  "end_schema_mapping".

        include = "include" file_name ";".

        schema_mapping_name = id.

        target_schema_name = id.

        source_schema_name = id.

        file_name = id.

uof     uof_mapping = uof_mapping_name "(" target_uof_name ")" ";"
        {{entity_mapping} {uof_template}} "end_uof_mapping".

        uof_mapping_name = id.

        target_uof_name = id.

entity  entity_mapping = "entity_mapping" entity_mapping_name
        (defined_in_subtypes | local_entity_mapping |
        reference_to_entity_mapping) "end_entity_mapping" ";".

        local_entity_mapping = "(" target_entity_name ","
```

source_entity_name ")" [inheritance] ";"
[entity_mapping_constraints] {{attribute_mapping}
{entity_template}}.

entity_mapping_constraint = "mapping_constraints"
entity_reference_path "end_mapping_constraints" ";".

reference_to_entity_mapping = "references"
qualified_entity_mapping_name [({attribute_mapping_name})]
";"[entity_mapping_constraints] {{attribute_mapping} {template}}.

defined_in_subtypes = "(" target_entity_name ")".

inheritance = "inherit" "(" entity_mapping_name {","
entity_mapping_name} ")".

entity_mapping_name = id.

target_entity_name = id.

source_entity_name = id.

qualified_entity_mapping_name = uof_mapping_name "."
entity_mapping_name.

attribute_maping = general_attribute_mapping |
mapping_of_value_of_attribute.

attribute **general_attribute_mapping** = "attribute_mapping"
attribute_mapping_name "(" target_attribute_name ","
source_attribute_element ["," target_attribute_type_name] ")" ";"
attribute_reference_path "end_attribute_mapping" ";".

attribute_mapping_name = id.

source_attribute_element = source_entity_name |
qualified_source_attribute_name.

qualified_source_attribute_name = source_entity_name "."
source_attribute_name.

target_attribute_type_name = target_type_name |
nested_target_types.

nested_target_types = target_type_name "(" (target_type_name |
nested_target_types) ")".

mapping_of_value_of_attribute = "attribute_value_mapping"
attribute_mapping_name "(" target_attribute_name "," mapped_value
")" ";" entity_reference_path "end_attribute_mapping" ";".

type **type_mapping** = "type_mapping" type_mapping_name "("
target_type_name "," source_element ")" ";" entity_reference_path
"end_type_mapping" ";".

template **template** = template_definition | template_usage.

template_definition = entity_mapping_template_def |
attribute_mapping_template_def | reference_path_template_def.

template_usage = entity_mapping_template_usage |
attribute_mapping_template_usage.

schema_template = template_definition.

uof_template = template_definition |
entity_mapping_template_usage.

entity_template = attribute_mapping_template_def |
reference_path_template_def | attribute_mapping_template_usage.

entity template

entity_mapping_template_def = "entity_mapping_template"
entity_mapping_template_name "/" template_parameters "/"
(defined_in_subtypes | local_entity_mapping |
reference_to_entity_mapping) "end_entity_mapping" ";".

template_paremters = template_parameter {"," template_parameter}.

template_parameter = id.

entity_mapping_template_name = id.

entity_mapping_template_usage = "entity_mapping_template_usage"
entity_mapping_name entity_mapping_template_name "/"
template_parameters_values "/"(defined_in_subtypes |
local_entity_mapping | reference_to_entity_mapping)
"end_entity_mapping" ";".

attribute template

attribute_mapping_template_def =
general_attribute_mapping_template |
attribute_value_mapping_template.

general_attribute_mapping_template = "attribute_mapping_template"
attribute_mapping_template_name "/" template_parameters "/" "("
target_attribute_name "," source_attribute_element [","
target_attribute_type_name] ")" ";" attribute_reference_path
"end_attribute_mapping" ";".

attribute_mapping_template_name = id.

attribute_value_mapping_template =
"attribute_value_mapping_template"
attribute_mapping_template_name "/" template_parameters "/" "("
target_attribute_name "," mapped_value ")" ";"
entity_reference_path "end_attribute_mapping" ";".

attribute_mapping_template_usage =
"attribute_mapping_template_usage"
attribute_mapping_template_name "/" template_parameters_values
"/".

reference path template

reference_path_template_def = "reference_path_template"
reference_path_template_name "/" template_parameters "/" ";"
reference_path "end_reference_path_template" ";".

reference_path_template_name = id.

reference path

reference_path = entity_reference_path |
attribute_reference_path.

enity_reference_path = [constraint] ".".

attribute_reference_path = [path_elements] ".".

path

path_elements = path_element {";" path_element}.

constraint = "{" [path_elements ";"] path_element | constraining_element) "}".

path_element = constraint | path.

path = or_element | backward | forward | repeat | strict_entity.
strict_entity = "|" entity_name "|".

or_element = "(" path_elements ")" "(" path_elements ")" {"(" path_elements ")"}.

attribute relations

backward = id_backward "<-" attribute_backward.

forward = attribute [aggregate] "->" id_forward {subtype}.

id_backward = entity_name {select_backward | nested_aggregate_backward}.

select_backward = "=" select_type.

nested_aggregate_backward = "=" aggregate_type_name aggregate.

aggregate = "[" ("i" | digits) "]".
attribute_backward = attribute aggregate {subtype}.

subtype = "=>" entity_name.
id_forward = {select_forward | nested_aggregate_forward} type.

select_forward = select_type "=".
nested_aggregate_forward = aggregate_type_name aggregate "=".

constraining_element = attribute [aggregate] ("=" | ("->" (select_forward | nested_aggregate_forward) {select | nested_aggregate}) value.

value = enumeration_value | integer_value | string_value | logical_value | real_value.

repeat

repeat = "(" path_elements ")*".

template usage

template_usage = "/" template_name "(" [parameter {"," parameter}] ")/".

parameter = entity_name | attribute_name | value.

remark

remark = "(*" {letter | digit | remark} "*)".

D.2 Lexical elements

The following rules specify how certain combinations of characters are interpreted as lexical elements within the mapping specification syntax. Only letters, numbers underscores, and plus signs "+" are allowed in lexical elements.

```
digit = "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" | "0"

digits = digit {digit}

entity_name = complex_entity_name | simple_entity_name.

simple_entity_name = id.

complex_entity_name = id "+" id {"+" id}.

select_type = id.

attribute = entity_name "." attribute_name.

attribute_name = id.

type = id.

id = letter {digit | letter | "_"}.

letter = "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" |
"l" | "m" | "n" | "o" | "p" | "q" | "r" | "r" | "s" | "t" | "u" | "v" | "w"
| "x" | "y" | "z".
```

Annex E
(informative)

Revision history

This edition may be immediately implemented by any project interested in doing so. The use of this edition is mandated only for documents submitted for stage 30 ballot one year after the approval of this document by SC4. Projects that have not yet produced a stage 30 document at the time of this documents approval by SC4 are strongly encouraged to use this edition.

Projects using this edition of the document must use this edition alone, and in its entirety.

Within this edition, the use of templates (annex A) and the formal mapping syntax (annex D) is optional.

Bibliography

[1] ISO 10303-1:1994, *Industrial automation systems and integration —Product data representation and exchange —Part 1: Overview and Fundamental Principles.*

[2] ISO 10303-11:1994, *Industrial automation systems and integration — Product data representation and exchange —Part 11: Description methods: The EXPRESS language reference manual.*

[3] ISO 10303-41:1994, *Industrial automation systems and integration — Product data representation and exchange —Part 41: Integrated generic resources: Fundamentals of product description and support.*

[4] ISO 10303-101:1994, *Industrial automation systems and integration — Product data representation and exchange —Part 101: Integrated application resources: Draughting.*

[5] ISO 10303-201:1994, *Industrial automation systems and integration — Product data representation and exchange —Part 201: Application protocol: Explicit draughting.*

[6] ISO 10303-202:1996, *Industrial automation systems and integration — Product data representation and exchange —Part 202: Application protocol: Associative draughting.*

[7] STONIS, Alfonsas; *Mapping syntax extensions,* ISO TC 184/SC4/QC N203, 2001-06-11.

[8] STONIS, Alfonsas; *Reference path syntax of mapping tables*, ISO TC 184/SC4/QC N163, 2000-09-21

[9] *Compilation of issues and proposals for Guidelines for the development of mapping tables, SC4 N533,* ISO TC 184/SC4/QC N135, 2000-01-29.

[10] *SC4 organization handbook.* ISO TC 184/SC4 N1087, 2000-10-10.