INTERNATIONAL STANDARD

ISO 15745-1

First edition
2003-03-01

Industrial automation systems and integration — Open systems application integration framework —

Part 1: Generic reference description

Systèmes d'automatisation industrielle et intégration — Charpente d'intégration d'application de systèmes ouverts —

Partie 1: Description de la référence générale

Reference number
ISO 15745-1:2003(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

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

ISO 15745-1 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 5, *Architecture, communications and integration frameworks*.

ISO 15745 consists of the following parts, under the general title *Industrial automation systems and integration — Open systems application integration framework*:

— *Part 1: Generic reference description*

— *Part 2: Reference description for ISO 11898-based control systems*

— *Part 3: Reference description for IEC 611158-based control systems*

— *Part 4: Reference description for Ethernet-based control systems*

Introduction

Real world application systems are developed from application specifications (i.e. specifications that describe the functionality and performance that are required for the application). Such application specifications typically contain textual descriptions, drawings, diagrams, and references to other specifications. Many system integrators and end users who frequently operate in specific market sectors either generate multiple similar application specifications (one for each project), or generate a master application specification with variants for each project.

The application integration framework (AIF) defines elements and rules that facilitate:

— the systematic organization and representation of the application integration requirements using integration models;

— the development of interface specifications in the form of application interoperability profiles (AIPs) that enable both the selection of suitable resources and the documentation of the "as built" application.

Figure 1 depicts the relationship between the AIF (specified in ISO 15745), the integration models and AIP (developed by an AIP developer), and a real world application system.

The left section of Figure 1 shows a generic AIF that is specified in part 1 of ISO 15745 and is extended in subsequent parts to cover specific technologies.

The middle section of Figure 1 shows the AIP (which can contain one or more other AIPs) consisting of one process profile, one or more resource profiles, and one or more information exchange profiles. Underlying the AIP are the relevant integration models which represent the application requirements.

The right section of Figure 1 shows the real world application system consisting of:

— resources (devices, communication networks, equipment, humans, materials);

— processes;

— exchanges of information.

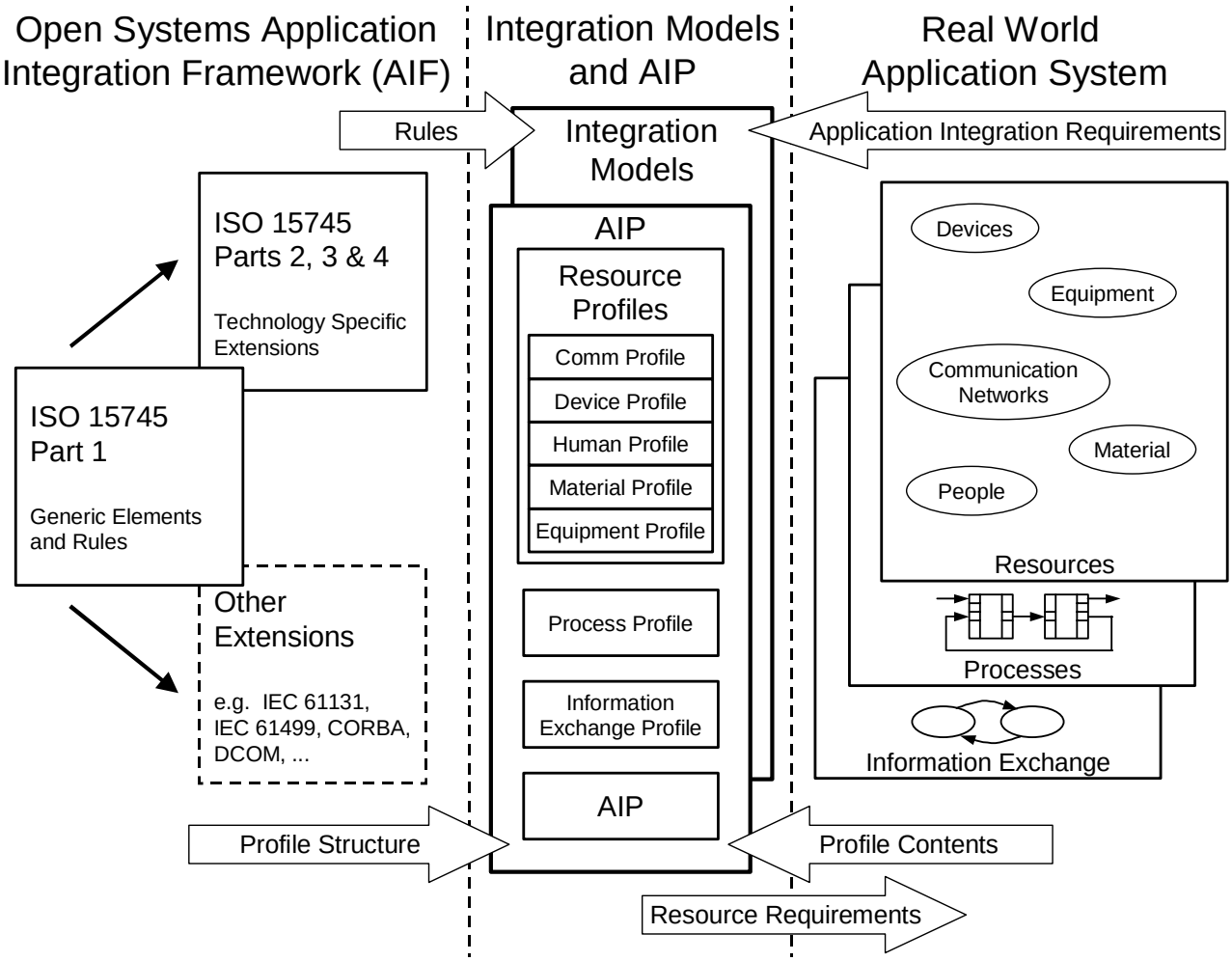


Figure 1 – Context of ISO 15745

Some of the benefits to be gained are that:

— end users can facilitate the specification and procurement of open systems by referencing pre-defined AIPs;

— system integrators can reduce the time to develop a solution based on open systems by using generic tools based on the AIF;

— automation vendors can provide and develop new products using generic tools based on the AIF e.g. an automation vendor can demonstrate that his products support the application requirements by referencing the resource profiles of an AIP.

The primary users of this International Standard will be developers of AIPs, using a variety of system platforms and product technologies in application domains such as:

— continuous process control systems;

— batch process control systems;

— machine control systems;

— discrete control systems;

— diagnostic systems.

Industrial automation systems and integration — Open systems application integration framework — Part 1: Generic reference description

1 Scope

This International Standard defines an application integration framework - a set of elements and rules for describing integration models and application interoperability profiles.

This part of ISO 15745 defines the generic elements and rules for describing integration models and application interoperability profiles, together with their component profiles - process profiles, information exchange profiles, and resource profiles.

NOTE Parts 2, 3 and 4 of this International Standard define the technology specific elements and rules for describing both communication network profiles and the communication related aspects of device profiles based upon particular fieldbus technologies - these parts can be used in conjunction with this part to form an application integration framework for a specific fieldbus technology.

This International Standard is applicable to industrial automation applications such as discrete manufacturing, process automation, electronics assembly, semiconductor fabrication, and wide-area material handling. It may also be applicable to other automation and control applications such as utility automation, agriculture, off-road vehicles, medical and laboratory automation, and public transport systems.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

ISO/IEC 7498-4:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 4 : Management framework*

IEC 61069-1:1991, *Industrial-process measurement and control – Evaluation of system properties for the purpose of system assessment – Part 1: General considerations and methodology*

REC-xml-20001006, *Extensible Markup Language (XML) 1.0 Second Edition – W3C Recommendation 6 October 2000*

REC-xmlschema-1-20010502, *XML Schema Part 1: Structures – W3C Recommendation 02 May 2001*

REC-xmlschema-2-20010502, *XML Schema Part 2: Datatypes – W3C Recommendation 02 May 2001*

UML V1.4, *OMG - Unified Modeling Language Specification (Version 1.4, September 2001)*

3 Terms and definitions

NOTE The UML terminology and notation used in this document is described in Annex A.

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

3.1
action
something which happens [ISO/IEC 10746-2]

NOTE Every action of interest for modelling purposes is associated with at least one object (see ISO/IEC 10746-2).

3.2
actor
coherent set of roles that users of use cases play when interacting with these use cases [UML]

NOTE An actor has one role for each use case with which it communicates.

3.3
aggregation
special form of association that specifies a whole-part relationship between the aggregate (whole) and a component part [UML]

3.4
AIP developer
person (or persons) who uses the application integration framework to develop integration models and/or profiles

3.5
association
semantic relationship between two or more classifiers that specifies connections among their instances [UML]

3.6
base specification
reference document containing information that is referenced by a profile

3.7
class
description of a set of objects that share the same attributes, operations, methods, relationships, and semantics [UML]

3.8
classifier
mechanism that describes behavioural and structural features [UML]

NOTE Classifiers include interfaces, classes, data types, and components.

3.9
communication network profile
representation of the integration aspects of a communication network supported by a networked device

EXAMPLE Examples of integration aspects are communication object types and the associated operating relationships (client-server, producer-consumer, etc.), services and attributes for the object types, data types for the object types and services, and encoding rules used.

3.10
compliance
relation between two specifications, A and B, that holds when specification A makes requirements which are all fulfilled by specification B (when B complies with A) [ISO/IEC 10746-1]

3.11
device
entity that performs control, actuating and/or sensing functions and interfaces to other such entities within an automation system

NOTE Devices alone do not perform material processing, material transport or material storage functions. (See 3.15).

3.12
device profile
representation of the integration aspects of a device

EXAMPLE Examples of integration aspects are functions provided, configuration across the network, behaviour on the network, and communication of I/O data.

3.13
element
atomic constituent of a model [UML]

3.14
entity
any concrete or abstract thing of interest [ISO/IEC 10746-2]

3.15
equipment
entity that is stand alone, or interfaces to an automation system, and that performs material processing, material transport, or material storage functions

EXAMPLE Conveyor, vessel, pump.

NOTE 1 Equipment can contain devices. (See 3.11).

NOTE 2 Equipment cannot have a direct connection to the communication network – only devices can be directly connected to the communication network.

3.16
equipment profile
representation of the integration aspects of an item of equipment

EXAMPLE Examples of integration aspects are conveyor speed, vessel capacity, pump delivery rate.

3.17
generalization
taxonomic relationship between a more general element and a more specific element [UML]

NOTE The more specific element is fully consistent with the more general element and contains additional information. An instance of the more specific element may be used where the more general element is allowed.

3.18
human profile
representation of the integration aspects of a person

EXAMPLE Examples of integration aspects are level of responsibility, level of competency, availability.

3.19
information
any kind of knowledge, that is exchangeable amongst users, about things, facts, concepts and so on, in a universe of discourse [ISO/IEC 10746-2]

3.20
instance
entity that has unique identity, a set of operations that can be applied to it, and state that stores the effects of the operations [UML]

3.21
interface
named set of operations that characterize the behaviour of an element [UML]

3.22
material
matter used in manufacturing the product

EXAMPLE Raw materials, consumables, catalysts.

3.23
material profile
representation of the integration properties of the material

EXAMPLE Examples of integration properties are dimensions, mass, density, shelf life, required storage temperature and humidity, hardness, formability, and viscosity.

3.24
message
specification of the conveyance of information from one instance to another, with the expectation that activity will ensue [UML]

3.25
name
term which, in a given naming context, refers to an entity [ISO/IEC 10746-2]

3.26
object
entity with a well-defined boundary and identity that encapsulates state and behaviour [UML]

NOTE State is represented by attributes and relationships, behaviour is represented by operations, methods, and state machines. An object is an instance of a class.

3.27
operation
service that can be requested from an object to effect behaviour [UML]

3.28
profile
set of one or more base specifications and/or sub-profiles, and, where applicable, the identification of chosen classes, conforming subsets, options and parameters of those base specifications, or sub-profiles necessary to accomplish a particular function, activity, or relationship

NOTE Adapted from ISO/IEC TR 10000-1

3.29
relationship
semantic connection among model elements [UML]

EXAMPLE Associations and generalizations.

3.30
resource
device, communication network, equipment, human or material used in a process

3.31
state
condition or situation during the life of an object during which it satisfies some condition, performs some activity, or waits for some event [UML]

3.32
system
something of interest as a whole or as comprised of parts [ISO/IEC 10746-2]

NOTE Therefore a system may be referred to as an entity. A component of a system may itself be a system, in which case it may be called a subsystem. (See ISO/IEC 10746-2).

3.33
<X> template
specification of the common features of a collection of <X>s in sufficient detail that an <X> can be instantiated using it [ISO/IEC 10746-2]

NOTE <X> can be anything that has a type

3.34
type (of an <X>)
predicate characterizing a collection of <X>s [ISO/IEC 10746-2]

3.35
use case
<class> specification of a sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system [UML]

3.36
view
projection of a model, which is seen from a given perspective or vantage point and omits entities that are not relevant to this perspective [UML]

4 Abbreviated terms

4.1 General

AIF	Application Integration Framework
AIP	Application Interoperability Profile
ASCII	American Standard Code for Information Interchange
CORBA	Common Object Request Broker Architecture
DCOM	Distributed Component Object Model
FDA	Food and Drug Administration
HMI	Human Machine Interface
IAS	Industrial Automation Systems
OSI	Open Systems Interconnection
UML	Unified Modelling Language
XML	eXtensible Markup Language

4.2 IAS interfaces types

NOTE See Annex B for details of IAS interface types.

API	Application Program Interface
CMI	Configuration Management Interface
CSI	Communication Services Interface
ESI	Engineering Support Interface
FSI	Facility Services Interface
HCI	Human/Computer Interface
ISI	Information Services Interface
MTI	Material Transport Interface
SEI	Safety and Environmental Interface
USI	Utility Services Interface

5 AIF concept

5.1 Elements and rules

The AIF specifies elements and rules. The AIP developer uses the elements and applies the rules in order to develop the integration models and profiles needed for an application.

The AIF elements defined in ISO 15745 are:

— the master profile template structure and header defined in 7.2;

— the generic profile templates defined in 7.3;

— the technology specific profile templates defined in parts 2, 3 and 4 of ISO 15745;

— the IAS interface types defined in Annex B.

The AIP developer will also use elements which are developed independently of ISO 15745 e.g. base specifications which are identified by the AIP developer.

The AIF rules constrain the AIP developer with respect to:

— creating integration models (see 6);

— the development of profiles (see 7), including the use of the master profile template (see 7.2), the use of the generic profile templates (see 7.3), and the development of technology specific profiles (see parts 2, 3 and 4 of ISO 15745);

— the profile exchange language (see 7.2.6).

5.2 Using the AIF to develop an AIP

5.2.1 Overview

Real world application systems are developed from application specifications (i.e. specifications that describe the functionality and performance that are required for the application). Such application specifications typically contain textual descriptions, drawings, diagrams, and references to other specifications. Many system integrators and end users who frequently operate in specific market sectors either generate multiple similar application specifications (one for each project), or generate a master application specification with variants for each project.

The AIF focuses on the integration aspects of an application system, and provides elements and rules for the development of integration models and profiles based on the process, information exchange, and resource views of the application (see Figure 2). Integration models represent the application requirements, and profiles are interface specifications that enable both the selection of suitable resources and the documentation of the "as built" application.

Integration models (see 6) are in the form of UML diagrams while profiles (see 7) are XML documents (REC-xml-20001006).

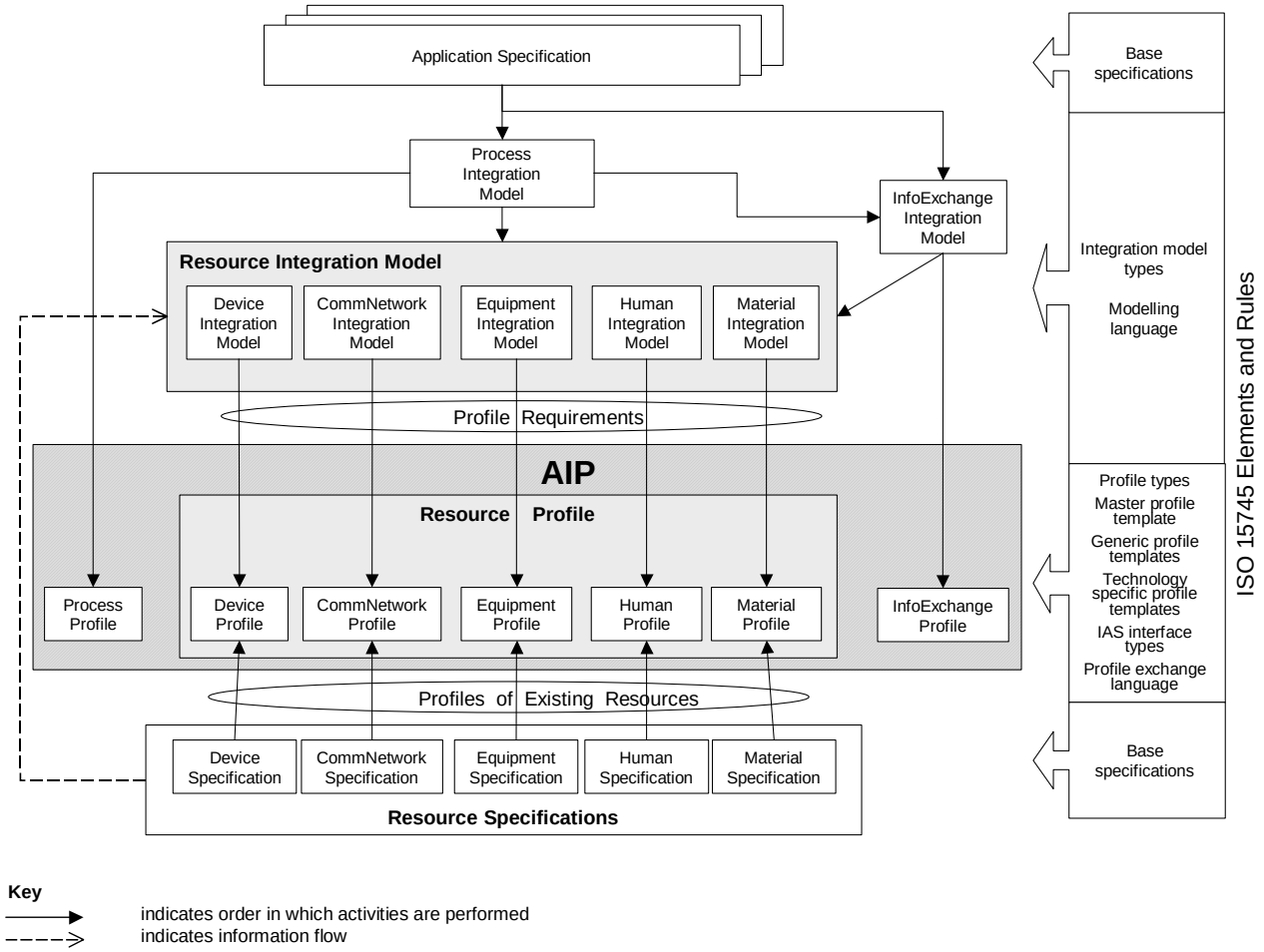


Figure 2 – Profile development using ISO 15745

5.2.2 Integration models

The three integration models (process, information exchange, and resource) are developed by the AIP developer and are a formal representation of the integration requirements contained in the application specification. Each

particular application can have its own application specification, but the AIP developer may choose whether an integration model is specific to an individual application or applies to a group of similar applications.

Each integration model consists of classes of collaborating objects. Relationships (associative, hierarchical) between these abstractions provide an overall representation of the application requirements. Each integration model is developed using a set of visual representations (i.e. UML diagrams as described in Annex A) of the classes and objects and their static and dynamic relationships.

The process integration model describes the control, material and information flows within the application, and determines many of the requirements for both the resources and the information to be exchanged.

The information exchange integration model describes the information that is exchanged among the resources.

The resource integration model describes the resources involved in the process being modelled and the interfaces among them. Separate integration models focus on each resource type (device, communication network, equipment, human and material).

5.2.3 Profiles

The integration models identify the interfaces necessary to support the application requirements - a profile is a precise representation of a particular such interface. Profiles can be concise by referencing selected options identified in the base specifications, thus avoiding the need to repeat the full text contained in the base specifications. Profiles are XML documents that are developed using profile templates (see 7.1) that are XML schemas (REC-xmlschema-1-20010502 and REC-xmlschema-2-20010502) (see 7.2.6).

An AIP is a single specification referencing a group of profiles (process profile, information exchange profile(s), resource profile(s), and sometimes other AIPs) that reference parts of base specifications (which may themselves be profiles).

The process profile is a record of the "as built" status of the process specified by the process integration model – it contains the "as built" information and/or references to base specifications that contain the "as built" information.

Similarly, the information exchange profile is a record of the "as built" status of the information exchange specified by the information exchange integration model - it contains the "as built" information and/or references to base specifications that contain the "as built" information.

Resource profiles (device, communication network, equipment, human and material) representing the requirements for the interfaces between the resources can be derived from the resource integration model (this is indicated by "Profile Requirements" in Figure 2). Resource profiles for existing resources are developed from the corresponding resource specifications (this is indicated by "Profile of Existing Resources" in Figure 2). Comparison of these two resource profiles (both in XML format) enables the AIP developer to match a particular resource to the application integration requirements.

6 Integration model types

6.1 Process integration model

The process integration model describes the control, material and information flows within the application, and describes the functions, steps, and activities involved in setting up and operating the application.

The AIP developer shall create the process integration model using the UML (see Annex A) – typical UML diagrams used are the use case diagram (showing the actors, use cases, and the application system boundary), activity diagrams (showing the flow of the process), class diagrams (showing the classes that exist and the relationships between them), and sequence diagrams (showing the process interactions arranged in time sequence).

6.2 Information exchange integration model

The information exchange integration model describes the information that is exchanged among the resources involved in the process being modelled.

This information exchanged typically includes the following:

— process information (such as recipes, geometric data, schedules, or other process parameters needed to perform the application);

— status information (such as faults, equipment status reports, alerts, and quality information);

— control information (such as commands and requests for service) identifying the source, destination, and type of information.

The AIP developer shall create the information exchange integration model using the UML (see Annex A) – typical UML diagrams used are class diagrams (showing the classes that exist and the relationships between them), collaboration diagrams (showing the information exchanged between the various objects), and sequence diagrams (showing the information exchanges arranged in time sequence).

6.3 Resource integration model

6.3.1 General

The resource integration model describes the devices, communication network, equipment, humans, and material involved in the process being modelled. The resource integration model also identifies the main interfaces between these resources that enable them to inter-operate and to provide the necessary functions and services to support the automation and control of the process.

The AIP developer shall create the resource integration model using the UML (see Annex A) – typical UML diagrams used are class diagrams (showing the classes that exist and the relationships between them), collaboration diagrams (showing the interactions between the various resources), and deployment diagrams (showing the hardware configuration of both the devices and the equipment, the software objects that reside on each piece of hardware, and the humans that interact with the hardware and software).

The resource integration model can be further detailed using integration models to describe each type of resource. The interfaces denoted in each integration model will determine the profile associated with each model.

6.3.2 Device integration model

The device integration model describes the properties of the device that are necessary to support the application requirements defined in the process integration model and the information exchange integration model e.g.:

— function(s) performed by the device;

— input and output data exchanged with the device;

— configuration parameters and runtime variables stored by the device.

The AIP developer shall create the device integration model using the UML (see Annex A) – typical UML diagrams used are class diagrams (showing the classes that exist and the relationships between them).

6.3.3 Communication network integration model

The communication network integration model describes the properties of the communication network that are necessary to support the application requirements defined in the process integration model and the information exchange integration model e.g.:

— topology, transmission media, number of nodes;

— transaction types;

— data throughput.

The AIP developer shall create the communication network integration model using the UML (see Annex A) – typical UML diagrams used are class diagrams (showing the classes that exist and the relationships between them), and collaboration diagrams (showing the transactions between the objects).

6.3.4 Equipment integration model

The equipment integration model describes the properties of the equipment that are necessary to support the application requirements defined in the process integration model and the information exchange integration model e.g.:

— conveyor speed;

— vessel capacity;

— pump delivery rate.

The AIP developer shall create the equipment integration model using the UML (see Annex A) – typical UML diagrams used are class diagrams (showing the classes that exist and the relationships between them), and deployment diagrams (showing the hardware configuration of the equipment).

6.3.5 Human integration model

The human integration model describes the properties of the personnel that are necessary to support the application requirements defined in the process integration model and the information exchange integration model e.g.:

— level of responsibility;

— level of competency;

— availability.

The AIP developer shall create the human integration model using the UML (see Annex A) – typical UML diagrams used are class diagrams (showing the human classes that exist and the relationships between them), collaboration diagrams (showing the interactions between the humans and other objects), and sequence diagrams (showing the human interactions arranged in time sequence).

6.3.6 Material integration model

The material integration model describes the properties of the material that are necessary to support the application requirements defined in the process integration model and the information exchange integration model e.g.:

— physical properties (dimensions, mass, density, etc.);

— storage properties (shelf life, required storage temperature and humidity, etc.);

— processing properties (hardness, formability, viscosity, etc.).

The AIP developer shall create the material integration model using the UML (see Annex A) – typical UML diagrams used are class diagrams (showing the material classes that exist and the relationships between them), collaboration diagrams (showing the interactions between the material and other objects), and sequence diagrams (showing the material interactions arranged in time sequence).

7 Profile templates and types

7.1 Profile template relationships

All generic profile templates are sub-classes of the master profile template and inherit all properties of the master profile template (see Figure 3).

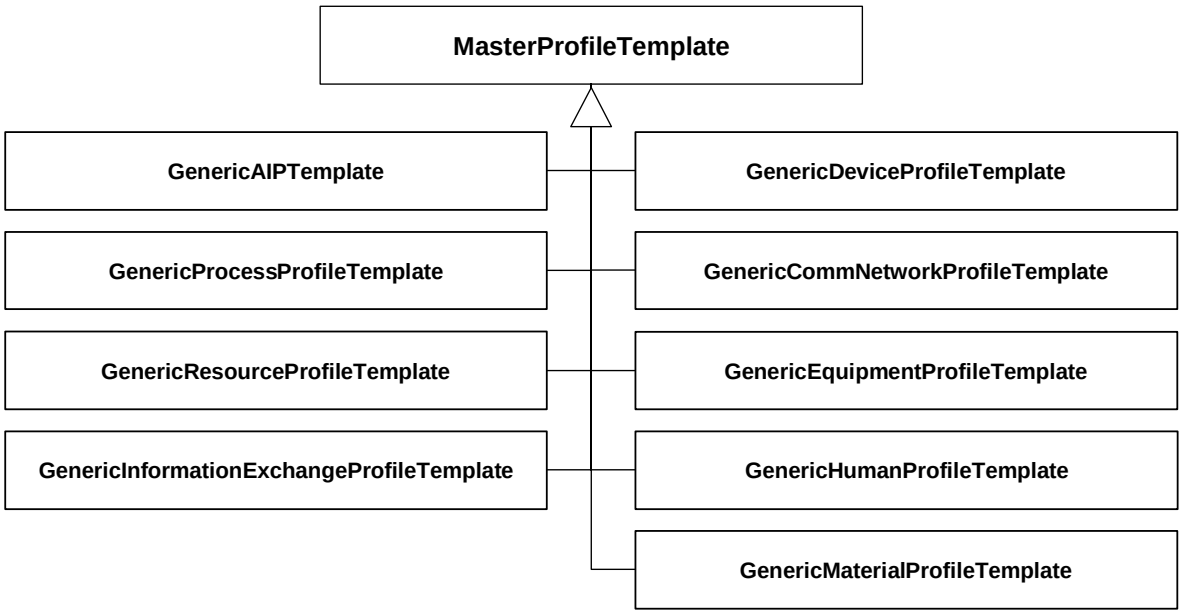


Figure 3 – Master / generic profile template class diagram

Parts 2, 3 and 4 of ISO 15745 define technology specific communication network profile templates and communication related aspects of device profile templates that are based upon particular fieldbus technologies. These technology specific profile templates inherit the structure of the generic communication network profile template defined in 7.3.4.3 and the structure of the generic device profile template defined in 7.3.4.2 respectively (see Figure 4).

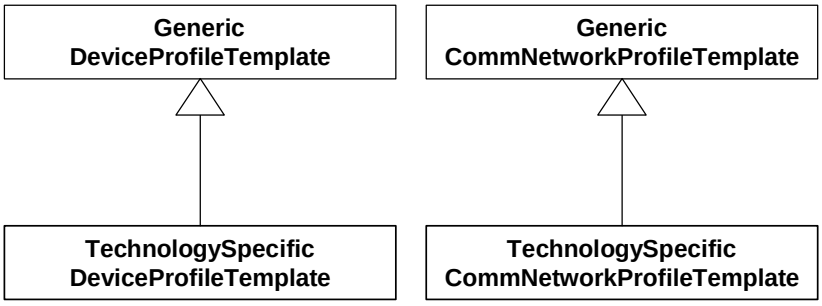


Figure 4 – Generic / technology specific profile template class diagram

Depending on their needs, AIP developers may:

— create their own profile templates (belonging to one of the generic profile template classes shown in Figure 3);

— create their own profile templates (belonging to one of the technology specific profile template classes shown in Figure 4 and defined in parts 2, 3 and 4 of ISO 15745);

— use existing profile templates that have already been created by other AIP developers (and which belong to one of the generic profile template classes shown in Figure 3);

— use existing profile templates that have already been created by other AIP developers (and which belong to one of the technology specific profile template classes shown in Figure 4 and defined in parts 2, 3 and 4 of ISO 15745).

When creating profile templates, AIP developers may add information to that inherited from the relevant generic or technology specific profile template, but may not modify or delete such information. AIP developers create profiles by adding profile specific information to the profile templates.

7.2 Master profile template

7.2.1 Structure

The master profile template consists of a header section and a body section (see Figure 5).

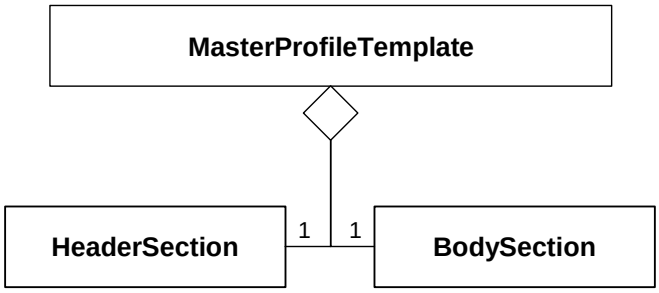


Figure 5 – Master profile template class diagram

7.2.2 Header section

The header section contains the attributes specified in Table 1. The attribute values shall be assigned by the AIP developer when constructing the profiles using the profile templates that have been derived from the master profile template.

Table 1 — Master profile template header attributes

Attribute	Description
ProfileIdentification	Profile identification. XML data type : string EXAMPLE : ABC-123-XX
ProfileRevision	Revision of the profile. XML data type : string EXAMPLE : 2.34
ProfileName	Descriptive name of the profile. XML data type : string EXAMPLE : AIP Material Handling
ProfileSource	Identification of the AIP developer. XML data type : string EXAMPLE : Profiles'R'Us

ProfileClassID	Identification of the profile class. XML data type : ProfileClassID_DataType (based on "string") – see 7.2.4. Valid profiles classes are: AIP Process InformationExchange Resource Device CommunicationNetwork Equipment Human Material EXAMPLE : AIP
ProfileDate	The release date of this revision of the profile in CCYY-MM-DD format. This field is optional. XML data type : date EXAMPLE : 2002-10-25
AdditionalInformation	Location of diagrams / additional information for the profile. This field is optional. XML data type : anyURI EXAMPLE : http://www.profilesrus.net
ISO15745Reference	Identifies the part of ISO 15745 (see ISO15745Part), together with its edition (see ISO15745Edition) and the profile technology (see ProfileTechnology). XML data type : ISO15745Reference_DataType – see 7.2.4. Multiple references are allowed e.g. for a device with more than one communication interface.
ISO15745Part	The part of the International Standard ISO 15745 with which the profile complies. XML data type : positiveInteger EXAMPLE : 1 (indicating ISO 15745-1)
ISO15745Edition	Edition of the referenced part of ISO 15745. XML data type : positiveInteger EXAMPLE : 1
ProfileTechnology	Name of the referenced technology within the previously specified part of ISO 15745 (see ISO15745Part field). XML data type : string The name associated with each technology is specified in the relevant part of ISO 15745. If no ISO 15745 technology is applicable, then the value “None” shall be used. EXAMPLE : None

IASInterfaceType	The IAS interface type. XML data type : IASInterfaceType_DataType (based on "string") – see 7.2.4. This field is optional. Valid IAS interface types are listed below and described in Annex B. Any combination of the following is permitted: a) IAS interface types defined in ISO/IEC TR 14252 (see B.1): CSI Communication Services Interface HCI Human/Computer Interface ISI Information Services Interface API Application Program Interface b) IAS interface types defined in ISO 15745 (see B.2): CMI Configuration Management Interface ESI Engineering Support Interface FSI Facility Services Interface MTI Material Transport Interface SEI Safety And Environmental Interface USI Utility Services Interface c) User defined IAS interface types (see B.3). EXAMPLE 1 : ISI ESI EXAMPLE 2 : CMI 37X6

7.2.3 Body section

The particular profile template determines the structure of the body section and the type of information it contains.

7.2.4 Header data types

The data types ProfileClassID_DataType, ISO15745Reference_DataType, and IASInterface_DataType are defined in this section of the profile template and are used in the header section.

7.2.5 ISO 15745 defined data types

A particular profile template may contain elements that reference other profile documents. Such elements shall use the data type "ProfileHandle_DataType" which is defined in Table 2.

Table 2 — ProfileHandle_DataType

Attribute	Description
ProfileHandle_DataType	Used to specify elements that contain information to identify a XML instance of a profile. It is composed of one ProfileIdentification, one ProfileRevision and an optional ProfileLocation. EXAMPLE 1 : ABC-123-XX 2.34 EXAMPLE 2 : ABC-123-XX 2.34 http://myserver.de/abc-123-xx.xml
ProfileIdentification	See Table 1.
ProfileRevision	See Table 1.
ProfileLocation	Address of a profile document file. XML data type : anyURI EXAMPLE : http://myserver.de/abc-123-xx.xml

7.2.6 XML Representation

Profiles are XML documents that can then be distributed, processed, and displayed as required. Profile templates can be represented as a skeletons for XML schemas. The skeleton XML schema (excluding target namespace information) for the master profile template is shown in Figure 6:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<!-- Target namespaces are not specified in this master template -->

<xsd:element name="ISO15745Profile">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="ProfileHeader" />
            <xsd:element ref="ProfileBody" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:annotation>
    <xsd:documentation>* HEADER SECTION *</xsd:documentation>
</xsd:annotation>

<xsd:element name="ProfileHeader">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="ProfileIdentification" type="xsd:string" />
            <xsd:element name="ProfileRevision" type="xsd:string" />
            <xsd:element name="ProfileName" type="xsd:string" />
            <xsd:element name="ProfileSource" type="xsd:string" />
            <xsd:element name="ProfileClassID" type="ProfileClassID_DataType" />
            <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0" maxOccurs="1" />
            <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0"
    maxOccurs="1" />
            <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"  />
            <xsd:element name="IASInterfaceType" type="IASInterface_DataType" minOccurs="0"
    maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:annotation>
    <xsd:documentation>* BODY SECTION *</xsd:documentation>
</xsd:annotation>

<xsd:element name="ProfileBody">

<!-- Profile body details are not specified in this master profile template -->

</xsd:element>

<xsd:annotation>
    <xsd:documentation>* HEADER DATA TYPES *</xsd:documentation>
</xsd:annotation>

<xsd:simpleType name="ProfileClassID_DataType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="AIP" />
        <xsd:enumeration value="Process" />
```

```
        <xsd:enumeration value="InformationExchange" />
        <xsd:enumeration value="Resource" />
        <xsd:enumeration value="Device" />
        <xsd:enumeration value="CommunicationNetwork" />
        <xsd:enumeration value="Equipment" />
        <xsd:enumeration value="Human" />
        <xsd:enumeration value="Material" />
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="ISO15745Reference_DataType">
    <xsd:sequence>
        <xsd:element name="ISO15745Part" type="xsd:positiveInteger" />
        <xsd:element name="ISO15745Edition" type="xsd:positiveInteger" />
        <xsd:element name="ProfileTechnology" type="xsd:string" />
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="IASInterface_DataType">
    <xsd:union>
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="CSI" />
                <xsd:enumeration value="HCI" />
                <xsd:enumeration value="ISI" />
                <xsd:enumeration value="API" />
                <xsd:enumeration value="CMI" />
                <xsd:enumeration value="ESI" />
                <xsd:enumeration value="FSI" />
                <xsd:enumeration value="MTI" />
                <xsd:enumeration value="SEI" />
                <xsd:enumeration value="USI" />
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:length value="4" />
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:union>
</xsd:simpleType>

<xsd:annotation>
    <xsd:documentation>* ISO 15745 DEFINED DATA TYPES *</xsd:documentation>
</xsd:annotation>

<xsd:complexType name="ProfileHandle_DataType">
    <xsd:sequence>
        <xsd:element name="ProfileIdentification" type="xsd:string" />
        <xsd:element name="ProfileRevision" type="xsd:string" />
        <xsd:element name="ProfileLocation" type="xsd:anyURI" minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
</xsd:complexType>

</xsd:schema>
```

Figure 6 – Master profile template XML schema

The master profile template XML schema shown in Figure 6 is the basis for all profile template XML schemas.

Parts 2, 3 and 4 of ISO 15745 provide XML schemas for certain technology specific profile templates that are derived from the generic communication network profile template defined in 7.3.4.3 and the generic device profile template defined in 7.3.4.2. The AIP developer can use these technology specific XML schemas as required.

In all other cases, the AIP developer shall develop his own XML schemas that are based on the master profile template XML schema shown in Figure 6.

AIP developers shall represent a profile as a valid XML document i.e. an XML document that is both well-formed as defined in the XML specification, and is in accordance with the constraints defined in the associated XML schema.

EXAMPLE Figure 7 shows an example of a header section of an XML document for an AIP based on an XML schema that was created from the master profile template XML schema shown in Figure 6, and using the example data given in Table 1.

```
<ProfileHeader>
    <ProfileIdentification>ABC-123-XX</ProfileIdentification>
    <ProfileRevision>2.34</ProfileRevision>
    <ProfileName>AIP Material Handling</ProfileName>
    <ProfileSource>Profiles'R'Us</ProfileSource>
    <ProfileClassID>AIP</ProfileClassID>
    <ProfileDate>2002-10-25</ProfileDate>
    <AdditionalInformation>http://www.profilesrus.net</AdditionalInformation>
    <ISO15745Reference>
        <ISO15745Part>1</ISO15745Part>
        <ISO15745Edition>1</ISO15745Edition>
        <ProfileTechnology>None</ProfileTechnology>
    </ISO15745Reference>
    <IASInterfaceType>ISI</IASInterfaceType>
    <IASInterfaceType>ESI</IASInterfaceType>
</ProfileHeader>
```

Figure 7 – Example of a header section of an XML document for an AIP

7.3 Generic profile templates

7.3.1 AIP

The AIP consists of one process profile, one or more resource profiles, and one or more information exchange profiles (see Figure 8). An AIP can also contain other AIPs.

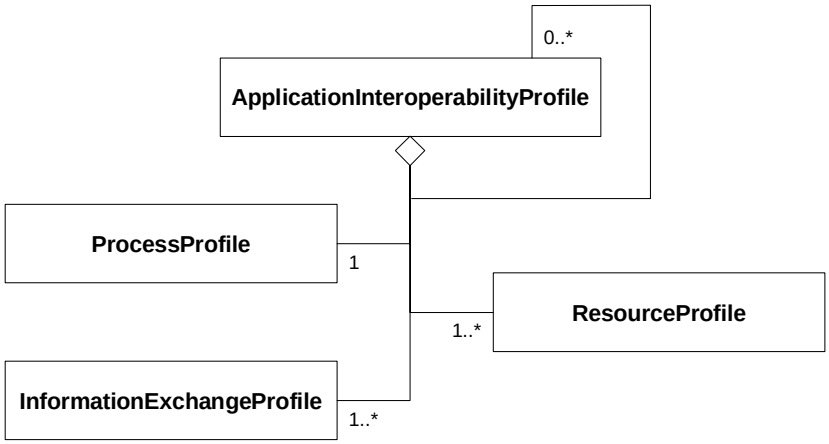


Figure 8 – Generic application interoperability profile class diagram

The header section of the generic AIP template XML schema is as shown in Figure 6.

The body section of the generic AIP template XML schema is shown in Figure 9:

```
<xsd:element name="ProfileBody">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="ApplicationInteroperabilityProfileHandle" minOccurs="0"
                    maxOccurs="unbounded" type="ProfileHandle_DataType" />
            <xsd:element name="ProcessProfileHandle" type="ProfileHandle_DataType" />
            <xsd:element name="InformationExchangeProfileHandle" minOccurs="1"
                    maxOccurs="unbounded" type="ProfileHandle_DataType" />
            <xsd:element name="ResourceProfileHandle" minOccurs="1" maxOccurs="unbounded"
                    type="ProfileHandle_DataType" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

Figure 9 – Body section of the generic AIP template XML schema

7.3.2 Process profile

The process profile is a record of the "as built" status of the process specified by the process integration model.

The process profile is an XML document that contains the "as built" information and/or references to base specifications that contain the "as built" information. Examples of such information include:

— the associated process integration model;

— operator instructions (e.g. faceplates, HMI screen menu hierarchy);

— process operation (e.g. recipe for brewing beer, engine assembly sequence, pipe and instrumentation diagrams);

— communication network diagrams (e.g. network topology, network configuration);

NOTE Communication network diagrams may also appear in the resource profile.

— test specifications and results (e.g. FDA approval, customer acceptance test);

— software (e.g. executable code, software design documentation);

— maintenance procedures (e.g. lubrication schedule, vibration monitoring);

— product type information (e.g. engine model number, batch identification);

— quality information (e.g. plant utilisation);

The header section of the generic process profile template XML schema is as shown in Figure 6.

NOTE The contents of the ProfileBody element of the generic process profile template XML schema are not specified in this Standard.

7.3.3 Information exchange profile

The information exchange profile is a record of the "as built" status of the information exchange specified by the information exchange integration model.

The information exchange profile is an XML document that contains the "as built" information and/or references to base specifications that contain the "as built" information. Examples of such information include:

— references to formats for data types e.g. IEC 61131-3, IEC 61158-5

— references to formats for data exchange e.g. IEC 61158-3/4/5/6

— references to data display formats e.g. language used, ISO 10646

— references to data storage formats e.g. XML, ASCII

— function blocks e.g. IEC 61499, IEC 61804

— programmable controller software coding e.g. IEC 61131-3

— interface definition language e.g. CORBA

The header section of the generic information exchange profile template XML schema is as shown in Figure 6.

NOTE The contents of the ProfileBody element of the generic information exchange profile template XML schema are not specified in this Standard.

7.3.4 Resource profile

7.3.4.1 General

The components of an automation system for a specific application type, and the relationships between the components shall be described using the resource profile shown in Figure 10. Each component of the resource profile shall be constructed using the IAS interfaces between the resources identified in each of the associated resource models.

The resource profile is an XML document that contains information about the relationships among the components of the resource profile. It can also contain common information such as communication network diagrams (e.g. network topology, network configuration).

NOTE Communication network diagrams may also appear in the process profile.

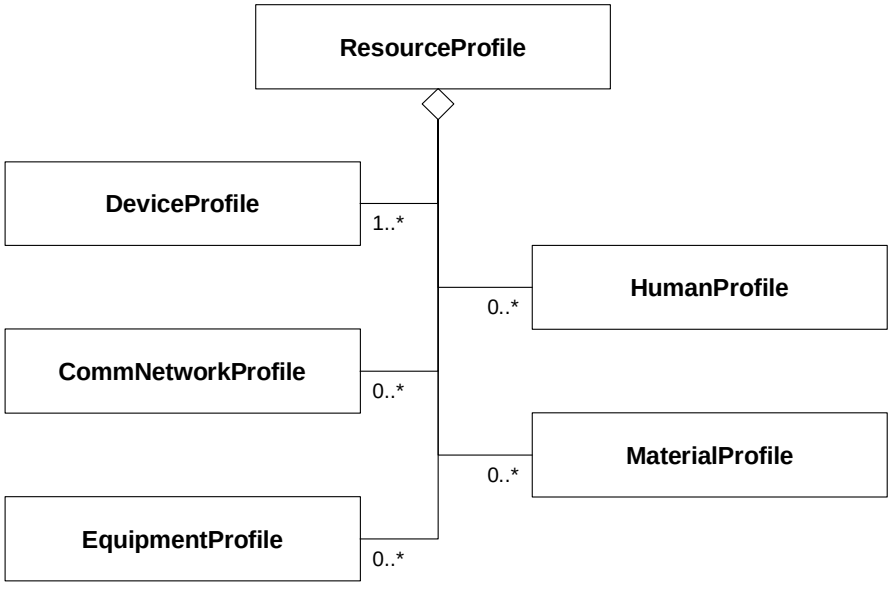


Figure 10 – Generic resource profile class diagram

The header section of the generic resource profile template XML schema is as shown in Figure 6.

The body section of the generic resource profile template XML schema is shown in Figure 11:

```
<xsd:element name="ProfileBody">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="DeviceProfileHandle" minOccurs="1" maxOccurs="unbounded"
                type="ProfileHandle_DataType" />
            <xsd:element name="CommNetworkProfileHandle" minOccurs="0"
                maxOccurs="unbounded" type="ProfileHandle_DataType" />
            <xsd:element name="EquipmentProfileHandle" minOccurs="0" maxOccurs="unbounded"
                type="ProfileHandle_DataType" />
            <xsd:element name="HumanProfileHandle" minOccurs="0" maxOccurs="unbounded"
                type="ProfileHandle_DataType" />
            <xsd:element name="MaterialProfileHandle" minOccurs="0" maxOccurs="unbounded"
                type="ProfileHandle_DataType" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

Figure 11 – Body section of the generic resource profile template XML schema

7.3.4.2 Device profile

7.3.4.2.1 General

Device profiles for existing devices are developed from the corresponding device specifications.

Device profiles representing the requirements for the interfaces between the device and other resources (including other devices) can be derived from the device integration model.

The AIP developer shall format the device profile representing the requirements for the interfaces using the same XML template as the device profile for the existing device - evaluation of these two profiles (both in XML format) allows comparison of the device to the device integration requirements.

If the device satisfies the device integration requirements, then it is suitable for the application. If not, then either another device should be considered, or the device integration requirements modified (e.g. using a combination of devices), or a new device can be developed to suit the device integration requirements.

Figure 12 shows the structure of the generic device profile. The technology specific device profile templates defined in parts 2, 3 and 4 of ISO 15745 are based on this generic device profile structure, and may extend it to accommodate legacy systems.

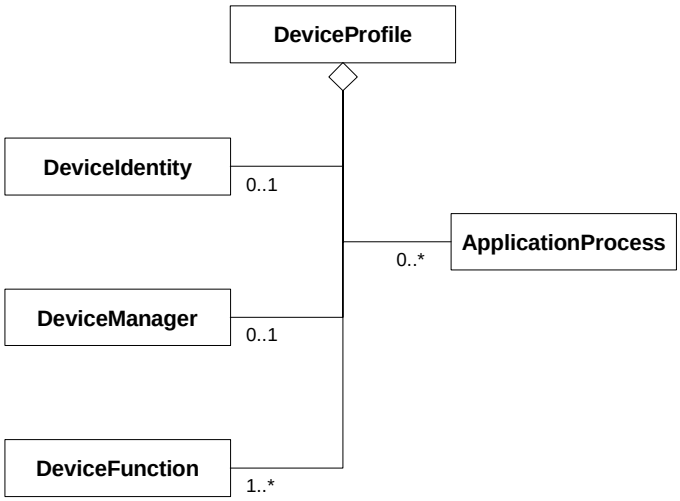


Figure 12 – Generic device profile class diagram

The header section of the generic device profile template XML schema is as shown in Figure 6.

The body section of the generic device profile template XML schema is shown in Figure 13:

```
<xsd:element name="ProfileBody">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="DeviceIdentity" minOccurs="0" maxOccurs="1" />
            <xsd:element ref="DeviceManager" minOccurs="0" maxOccurs="1" />
            <xsd:element ref="DeviceFunction" minOccurs="1" maxOccurs="unbounded" />
            <xsd:element ref="ApplicationProcess" minOccurs="0" maxOccurs="unbounded" />
            <xsd:element name="ExternalProfileHandle" type="ProfileHandle_DataType" minOccurs="0"
    maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

NOTE The additional ExternalProfileHandle element shown in Figure 13 enables external non-XML data to be referenced. The ProfileIdentification, ProfileRevision, and ProfileLocation elements of the ProfileHandle_DataType refer to the external non-XML data file. This can be used by legacy systems that are in the process of migrating to XML.

Figure 13 – Body section of the generic device profile template XML schema

7.3.4.2.2 Device identity

The device identity object contains attributes which uniquely identify the device. Examples of such attributes are the manufacturer's identification, part number, revision, location of storage of additional information, and indication of the number and type of additional objects within the device.

7.3.4.2.3 Device manager

The device manager object represents the set of attributes (e.g. revision of the device identity object) and services (e.g. reset, configure/run mode, retrieval of device manager object attributes) used to configure and to monitor a device integrated into the application system.

7.3.4.2.4 Device function

The device function object describes the intrinsic function of a device in terms of its technology (e.g. mechanical limit switch, proximity sensor, ultrasonic sensor). The device function object differentiates the technology of the device from the application of the device. Examples of device function objects are analog current input in milliamps, and discrete voltage output in volts.

7.3.4.2.5 Application process

The application process object represents a set of attributes and services that correspond to the application requirements captured in the attributes and services of the associated process profile. The application process object therefore describes the behaviour of the device in terms of the application, independent of the device technology.

EXAMPLE An example of an application process object is a section of code within a device that detects, validates, and reports the presence (or absence) of a part, independent of the device technology being used. An infrared photoelectric sensor, a capacitive proximity switch, or a piezoelectric pressure device can meet the application requirement represented by the same application process object.

A simple device may contain one application process object. A complex device may contain one or more application process objects. In a distributed system, one application process object may span a number of devices.

7.3.4.3 Communication network profile

7.3.4.3.1 General

Communication network profiles for existing devices with communication capability are developed from the corresponding communication network specifications.

EXAMPLE Examples of device types with network-oriented device profiles include programmable controllers, personal computers, valves, drives, motor starters, block I/O modules, limit switches, and photoelectric switches.

Communication network profiles representing the requirements for the interfaces between the communication network and other resources (including other communication networks) can be derived from the communication network integration model.

The AIP developer shall format the communication network profile representing the requirements for the interfaces using the same XML template as the communication network profile for the existing device - evaluation of these two profiles (both in XML format) allows comparison of the communication network configuration with the communication network integration requirements.

If the communication network configuration satisfies the communication network integration requirements, then it is suitable for the application. If not, then either another communication network should be considered, or the communication network integration requirements modified (e.g. using a combination of communication networks), or a new communication network can be developed to suit the communication network integration requirements.

Figure 14 shows the structure of the generic communication network profile which is based on the OSI reference model defined in ISO/IEC 7498-1. The technology specific communication network profile templates defined in parts 2, 3 and 4 of ISO 15745 are either based on this generic communication network profile structure, or alternatively reference external data in order to accommodate legacy systems.

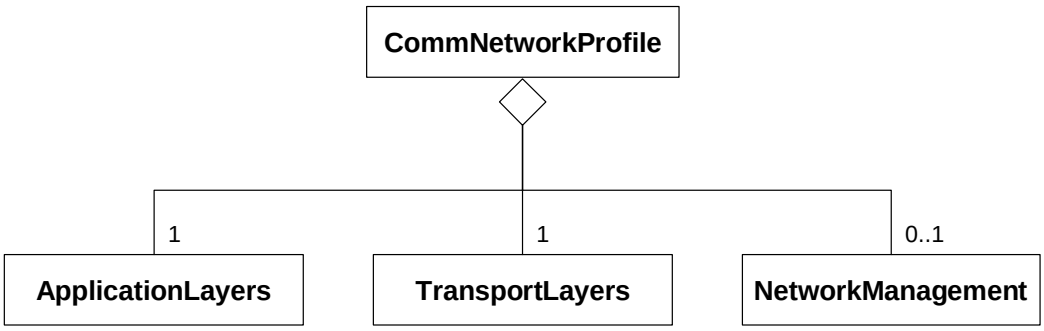


Figure 14 – Generic communication network profile class diagram

The header section of the generic communication network profile template XML schema is as shown in Figure 6.

The body section of the XML schema representing the generic communication network profile template is shown in Figure 15:

```
<xsd:element name="ProfileBody">
    <xsd:complexType>
        <xsd:choice>
            <xsd:sequence>
                <xsd:element ref="ApplicationLayers" />
                <xsd:element ref="TransportLayers" />
                <xsd:element ref="NetworkManagement" minOccurs="0" maxOccurs="1" />
            </xsd:sequence>
            <xsd:element name="ExternalProfileHandle" type="ProfileHandle_DataType" />
        </xsd:choice>
    </xsd:complexType>
</xsd:element>
```

NOTE The alternate ExternalProfileHandle element shown in Figure 15 enables external non-XML data to be referenced. The ProfileIdentification, ProfileRevision, and ProfileLocation elements of the ProfileHandle_DataType refer to the external non-XML data file. This can be used by legacy systems that are in the process of migrating to XML.

Figure 15 – Body section of the generic communication network profile template XML schema

7.3.4.3.2 Application layers

The application layers object represents OSI layers 5 to 7 (Session, Presentation, Application) as defined in ISO/IEC 7498-1.

7.3.4.3.3 Transport layers

The transport layers object represents OSI layers 1 to 4 (Physical, Data Link, Network, Transport) as defined in ISO/IEC 7498-1.

7.3.4.3.4 Network management

The network management object represents the following network management functional areas (see ISO/IEC 7498-4):

— fault management;

— configuration management;

— performance management.

The following network management functions should also be considered:

— security management (see ISO/IEC 7498-4);

— dependability (see IEC 61069-1).

7.3.4.4 Equipment profile

Equipment profiles for existing equipment are developed from the corresponding equipment specifications.

Equipment profiles representing the requirements for the interfaces between the equipment and other resources (including other equipment) can be derived from the equipment integration model.

The AIP developer shall format the equipment profile representing the requirements for the interfaces using the same XML template as the equipment profile for the existing equipment - evaluation of these two profiles (both in XML format) allows comparison of the equipment with the equipment integration requirements.

If the equipment satisfies the equipment integration requirements, then it is suitable for the application. If not, then either another piece of equipment should be considered, or the equipment integration requirements modified (e.g. using a combination of equipment), or new equipment can be developed to suit the equipment integration requirements.

The header section of the generic equipment profile template XML schema is as shown in Figure 6.

NOTE The contents of the ProfileBody element of the generic equipment profile template XML schema are not specified in this Standard.

7.3.4.5 Human profile

Human profiles for existing people are developed from the corresponding job specifications.

Human profiles representing the requirements for the interfaces between the human and other resources (including other humans) can be derived from the human integration model.

The AIP developer shall format the human profile representing the requirements for the interfaces using the same XML template as the human profile for the existing person - evaluation of these two profiles (both in XML format) allows comparison of the person with the human integration requirements.

If the person satisfies the human integration requirements, then the person is suitable for the application. If not, then either another person should be considered, or the human integration requirements modified (e.g. using a combination of people to perform a task), or an existing person could undergo training in order to meet the human integration requirements.

The header section of the generic human profile template XML schema is as shown in Figure 6.

NOTE The contents of the ProfileBody element of the generic human profile template XML schema are not specified in this Standard.

7.3.4.6 Material profile

Material profiles for existing material are developed from the corresponding material specifications.

Material profiles representing the requirements for the interfaces between the material and other resources (including other material) can be derived from the material integration model.

EXAMPLE An example of an interface specification that could be referenced in a material profile is the product data exchange specifications in ISO 10303.

The AIP developer shall format the material profile representing the requirements for the interfaces using the same XML template as the material profile for the existing material - evaluation of these two profiles (both in XML format) allows comparison of the material with the material integration requirements.

If the material satisfies the material integration requirements, then it is suitable for the application. If not, then either an alternative material should be considered, or the material integration requirements modified (e.g. using a combination of different materials), or a new material can be developed to suit the material integration requirements.

The header section of the generic material profile template XML schema is as shown in Figure 6.

NOTE The contents of the ProfileBody element of the generic material profile template XML schema are not specified in this Standard.

8 Compliance

AIP developers are responsible for ensuring that any integration models, profile templates, and/or profiles developed and/or used satisfy the requirements contained in the relevant part(s) of ISO 15745.

Annex A
(informative)

UML terminology and notation

A.1 General

This annex explains the UML terminology and notation used in this part of ISO 15745. It is provided to aid the reader in understanding ISO 15475 – it is not intended to provide an introduction to UML.

A.2 UML diagrams

A.2.1 Use case diagram

A use case diagram shows the relationships among actors (see 3.2) and use cases (see 3.35) within a system.

A.2.2 Class diagram

A class diagram describes the static structure of a system – it shows the classes, their contents and relationships.

A.2.3 Behaviour diagrams

A.2.3.1 Statechart diagram

A statechart diagram shows how an object behaves - it shows the relationships among the various states of the object, and how the object responds to events.

A.2.3.2 Activity diagram

An activity diagram is a special case of a statechart diagram in which all (or most) of the transitions are triggered by completion of the previous state i.e. the activity flows are driven by internal processing, as opposed to external events.

A.2.3.3 Interaction diagrams

NOTE Sequence diagrams and collaboration diagrams express similar information, but show it in different ways.

A.2.3.3.1 Sequence diagram

A sequence diagram shows object interactions arranged in time sequence. In particular, it shows the objects participating in the interaction and the sequence of messages exchanged.

NOTE Unlike a collaboration diagram, a sequence diagram includes time sequences but does not include object relationships.

A.2.3.3.2 Collaboration diagram

A collaboration diagram shows the interactions and the relationships among the objects.

NOTE Unlike a sequence diagram, a collaboration diagram shows the relationships among the objects.

A.2.4 Implementation diagrams

A.2.4.1 Component diagram

A component diagram shows the organizations and dependencies among components (physical parts of a system).

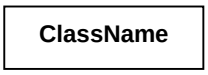
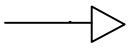
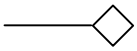
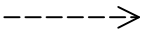
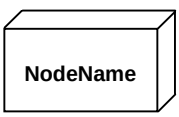
A.2.4.2 Deployment diagram

A deployment diagram shows the organization of the processing resources (hardware topology) and the binding of the software to the various resources.

A.3 UML notation

The following notation described in Table A.1 is used in the UML class diagrams (see 7), UML component diagrams (see Annex B), and UML deployment diagrams (see Annex B).

Table A.1 — UML notation

Symbol	Description
	Class (see 3.7) – used in class diagrams. Class attributes and operations are not shown in this part of ISO 15745.
	Association (see 3.5). Indicates a relationship between two model elements.
	Generalization (see 3.17). The more specific element (sometimes known as the child or subclass) inherits the attributes and operations of the more general element (sometimes known as the parent or superclass). The hollow triangle points toward the more general element.
	Aggregation (see 3.3). The aggregate is made up of its component parts. The hollow diamond is attached to the class that is the aggregate.
	Dependency. The model element at the tail of the arrow (the client) depends on the model element at the arrow head (the supplier).
	Note. Contains textual information.
	Component - used in component diagrams. Physical part of a system (e.g. software code) that resides on a node.
	Node - used in a deployment diagrams. A physical object that represents a processing resource, generally having at least a memory and often processing capability as well.

Annex B
(informative)

IAS interface types

B.1 IAS interface types (ISO/IEC TR 14252)

The following IAS interface types are defined in ISO/IEC TR 14252 and are reproduced here for convenience:

— application program interface (API) : interface between the application software and the application platform, across which all services are provided [ISO/IEC TR 14252]

— communication services interface (CSI) : boundary across which access to services for interaction between internal application software entities and application platform external entities is provided [ISO/IEC TR 14252]

— human / computer interface (HCI) : boundary across which physical interaction between a human being and the application platform takes place [ISO/IEC TR 14252]

— information services interface (ISI) : boundary across which external, persistent information storage service is provided [ISO/IEC TR 14252]

B.2 IAS interface types (ISO 15745)

The following IAS interface types are defined in this annex:

— configuration management interface (CMI) : interface between an automation system and a provider of system configuration services

EXAMPLE 1 An automation system that provides means for dynamic re-configuration may utilize a CMI to manage and/or change the system configuration and interconnections. For example, a system supporting this type of interface may allow replacement of component B with component C.

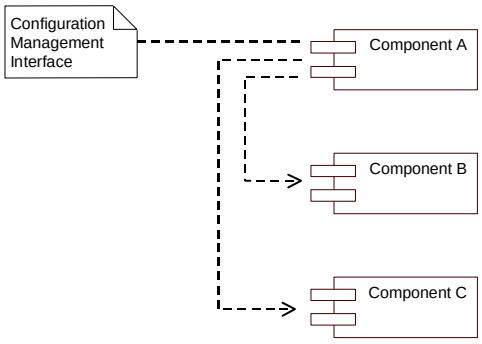


Figure B1 – Component diagram showing an example of a CMI

— engineering support interface (ESI) : interface between an automation system and an external system that provides design support, documentation, diagnostic, and simulation functions

EXAMPLE 2 An automation run-time system that has an interface to a repository of information that documents the "as installed" or "as commissioned" state of the system.

Figure B2 – Deployment diagram showing an example of an ESI

— facility services interface (FSI) : interface between an automation system and an external system that provides services (such as heating, ventilation, air conditioning, access control, and security) within a facility

EXAMPLE 3 An automation system that has an interface to a water heating system.

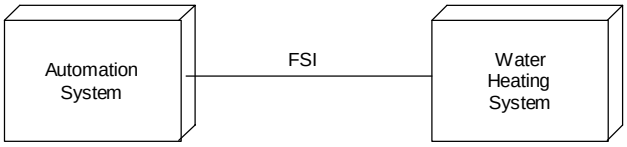


Figure B3 – Deployment diagram showing an example of a FSI

— material transport interface (MTI) : interface across which material transfer activity between the manufacturing or production infrastructure and the automation equipment can be abstracted

EXAMPLE 4 An automation system that has an interface to a robot that performs the task of transferring a part from a machine tool and placing it on a conveyor. The MTI interface could specify requirements such as number of parts/hour, size constraints, etc.

Figure B4 – Deployment diagram showing an example of a MTI

— safety and environmental interface (SEI) : interface between an automation system and an external system that provides safety and environmental management services

EXAMPLE 5 An automation system that has an interface to a perimeter guarding system to prevent unprotected intrusion into a dangerous area of a manufacturing system.

Figure B5 – Deployment diagram showing an example of a SEI

— utility services interface (USI) : interface across which power (electrical, mechanical, fluid, etc.) is delivered between a power source and an automation system

EXAMPLE 6 An automation system that has an interface to an electrical power source. The USI interface could specify requirements such as voltage regulation and conditioning, current, uptime, etc.

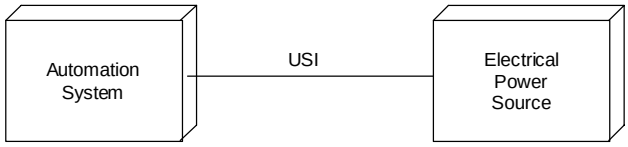


Figure B6 – Deployment diagram showing an example of a USI

B.3 User defined IAS interface types

If none of the above IAS interface types are applicable, then users may define their own IAS interface types together with a 4 character string mnemonic.

Bibliography

[1] ISO 8879:1986, *Information processing – Text and office systems – Standard Generalized Markup Language (SGML)*

[2] ISO/IEC TR 10000-1:1998, *Information technology – Framework and taxonomy of International Standardized Profiles – Part 1: General principles and documentation framework*

[3] ISO/IEC TR 10000-2:1998, *Information technology – Framework and taxonomy of International Standardized Profiles – Part 2: Principles and Taxonomy for OSI Profiles*

[4] ISO/IEC TR 10000-3:1998, *Information technology – Framework and taxonomy of International Standardized Profiles – Part 3: Principles and Taxonomy for Open System Environment Profiles*

[5] ISO/IEC 10646 (all parts), *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*

[6] ISO/IEC 10746-1:1998, *Information technology – Open Distributed Processing – Reference Model: Overview*

[7] ISO/IEC 10746-2:1996, *Information technology – Open Distributed Processing – Reference Model: Foundations*

[8] ISO/IEC 10746-3:1996, *Information technology – Open Distributed Processing – Reference Model: Architecture*

[9] ISO/IEC TR 14252:1996, *Information technology – Guide to the POSIX Open System Environment (OSE)*

[10] ISO 15704:2000, *Industrial automation systems – Requirements for enterprise-reference architectures and methodologies*

[11] IEC 61131-3:1993, *Programmable controllers – Part 3: Programming languages*

[12] IEC 61158 (all parts), *Digital data communications for measurement and control – Fieldbus for use in industrial control systems*

[13] IEC/PAS 61499 (all parts), *Function blocks for industrial-process measurement and control systems*

[14] IEC 61804 (all parts)[1)], *Function blocks for process control*

[15] IEC TS 61915:—[1)], *Low-voltage switchgear and controlgear – Representation of networked industrial devices*

1) To be published.

ICS 25.040.40

Price based on 31 pages