INTERNATIONAL STANDARD

ISO 16100-6

First edition
2011-06-15

Industrial automation systems and integration — Manufacturing software capability profiling for interoperability —

Part 6: Interface services and protocols for matching profiles based on multiple capability class structures

Systèmes d'automatisation industrielle et intégration — Profil d'aptitude du logiciel de fabrication pour interopérabilité —

Partie 6: Services d'interface et protocoles pour la correspondance des profils fondés sur les structures de classe d'aptitude multiple

Reference number
ISO 16100-6:2011(E)

COPYRIGHT PROTECTED DOCUMENT

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

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.

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 16100-6 was prepared by Technical Committee ISO/TC 184, *Automation systems and integration*, Subcommittee SC 5, *Interoperability, integration, and architectures for enterprise systems and automation applications*.

ISO 16100 consists of the following parts, under the general title *Industrial automation systems and integration — Manufacturing software capability profiling for interoperability*:

— *Part 1: Framework*

— *Part 2: Profiling methodology*

— *Part 3: Interface services, protocols and capability templates*

— *Part 4: Conformance test methods, criteria and reports*

— *Part 5: Methodology for profile matching using multiple capability class structures*

— *Part 6: Interface services and protocols for matching profiles based on multiple capability class structures*

Introduction

The motivation for ISO 16100 stems from the industrial and economic environment, in particular:

a) a growing base of vendor-specific solutions;

b) user difficulties in applying standards;

c) the need to move to modular sets of system integration tools;

d) recognition that application software and the expertise to apply that software are assets of the enterprise.

ISO 16100 is an International Standard for the computer-interpretable and human-readable representation of a capability profile. Its goal is to provide a method to represent the capability of manufacturing application software relative to its role throughout the life cycle of a manufacturing application, independent of a particular system architecture or implementation platform. This can lead to reduced production and information management costs to users and vendors/suppliers of manufacturing applications.

Certain diagrams in this part of ISO 16100 are constructed following UML conventions. Because not all concepts embodied in these diagrams are explained in the text, some familiarity with UML on the part of the reader is assumed.

Industrial automation systems and integration — Manufacturing software capability profiling for interoperability —

Part 6:
Interface services and protocols for matching profiles based on multiple capability class structures

WARNING — This part of ISO 16100 provides a specification intended to be implemented in software. Incompatibilities may result in machine-to-machine communication in the case of software developed on the basis of translations of this part of ISO 16100 into languages other than the official ISO languages. Accordingly, it is strongly recommended that any implementations be developed only on the basis of the texts in the official ISO languages.

1 Scope

This part of ISO 16100 defines the detailed interface services and protocols used in a matching method based on multiple capability class structures. This part of ISO 16100 also defines a Capability Profile Template Interface (CPTI) Service Group, an Extended Capability Profile Interface (CPI) Service Group and an Extended Matcher Interface Service Group, which are extensions of the Type 1, Type 2 and Type 3 services, respectively, as specified in ISO 16100-3:2005, 5.4.

This part of ISO 16100 also defines the Capability Class Structure Interface (CCSI) Service Group, an additional service group used to create, register, access and modify a capability class structure for the reference manufacturing domain models, as specified in ISO 16100-5:2009, Clause 6.

This part of ISO 16100 also specifies detailed contents of the specific part of a capability profile template as defined in ISO 16100-5:2009, Clause 7.

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 16100-1, *Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 1: Framework*

ISO 16100-2, *Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 2: Profiling methodology*

ISO 16100-3, *Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 3: Interface services, protocols and capability templates*

ISO 16100-4:2006, *Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 4: Conformance test methods, criteria and reports*

ISO 16100-5:2009, *Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 5: Methodology for profiling matching using multiple capability class structures*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 16100-1, ISO 16100-2, ISO 16100-3, ISO 16100-4 and ISO 16100-5 and the following apply.

3.1
capability class
〈software unit capability class〉 element within the capability profiling method that represents software unit functionality and behaviour with regard to the software unit's role in a manufacturing activity

NOTE 1 The role of a Manufacturing Software Unit (MSU) changes when used in different manufacturing activities; however, the MSU's corresponding capability class is positioned uniquely in an inheritance structure, but it can assume different positions in an aggregation structure.

NOTE 2 In this part of ISO 16100, a capability class template is identical to a capability profile template (see ISO 16100-2:2003, 6.3, for requirements for capability templates).

NOTE 3 Adaped from ISO 16100-2:2003, definition 3.3.

NOTE 4 In general, a capability class maps to an activity. The capability class is distinct within a capability inheritance structure and can form a capability aggregation structure with other capability classes.

3.2
capability class structure
hierarchy of capability classes

NOTE This structure is intended for modeling capability aggregation hierarchies in the target domains of ISO 16100-1:2009, Figure 2.

3.3
capability class structure template

extensible markup language (XML) schema representing a capability class structure

NOTE Adapted from ISO 16100-5:2009, definition 3.2.

3.4
capability profile template
schema for a manufacturing software capability profile

3.5
extended service interface
set of service access points defined in this part of ISO 16100 that handle manufacturing domain data, manufacturing domain models, capability class structures, capability profiles and capability profile templates

NOTE "Extended" refers to both the services specified in this part of ISO 16100 and the "basic" services specified in ISO 16100-3.

3.6
manufacturing domain data
information, represented by a unified modeling language (UML) class, about manufacturing resources, manufacturing activities, or items exchanged among manufacturing resources within a particular manufacturing domain

NOTE Adapted from ISO 16100-5:2009, definition 3.3.

3.7
manufacturing domain data template
extensible markup language (XML) schema representing a manufacturing domain data

[ISO 16100-5:2009, definition 3.4]

3.8
manufacturing domain model
particular view of a manufacturing domain, consisting of manufacturing domain data and relationships among them, corresponding to the domain's applications

[ISO 16100-5:2009, definition 3.5]

3.9
parts library
⟨manufacturing⟩ collection of part descriptions or catalogue

NOTE The term "parts library" also refers to a dictionary such as a PLIB dictionary in ISO 13584 or OTD in ISO 22745.

4 Symbols and abbreviated terms

BSU Basic Semantic Unit

CCS Capability Class Structure

CCSI Capability Class Structure Interface

CPI Capability Profile Interface

CPTI Capability Profile Template Interface

CSI Conformance Statement for the Implementation

ESI Extended Service Interface

ESP Extended Service Provider

ICD International Code Designator

MDD Manufacturing Domain Data

MDM Manufacturing Domain Model

MSU Manufacturing Software Unit

OTD Open Technical Dictionary

PLIB Parts Library (as specified in ISO 13584)

UML Unified Modeling Language

URL Uniform Resource Locator

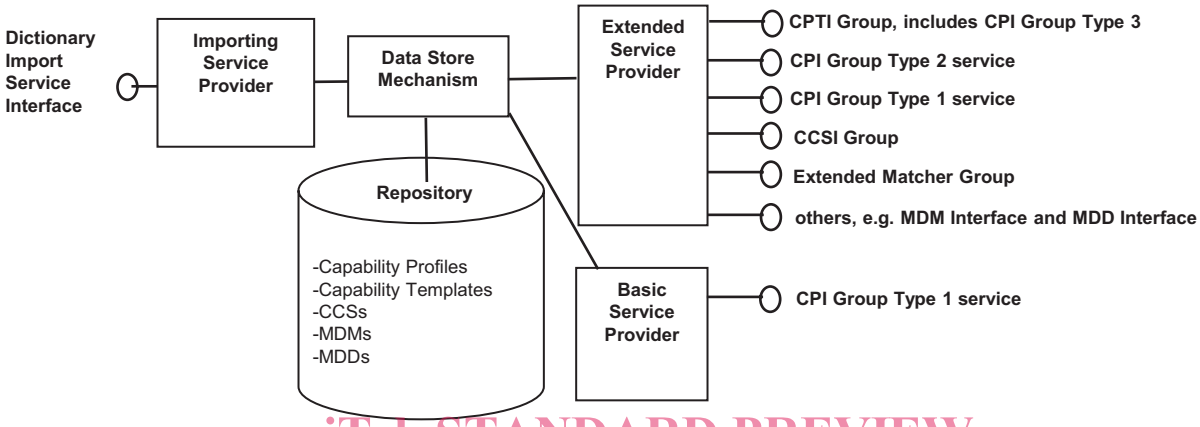
URN Uniform Resource Name

XML eXtensible Markup Language

5 Service provider interface services

5.1 Service sets

Figure 1 shows all the services, and their relations to Extended Service Providers and Basic Service Providers, for handling capability profiles, capability profile templates, CCSs, MDMs, MDDs and MDD objects. Basic Service Providers handle CPI group Type 1 services. Extended Service Providers handle CPTI, CPI and CCSI services. In addition, Extended Service Providers support extended matcher services and other service interfaces for handling MDMs and MDDs.



NOTE 1 This figure is not in accordance with UML conventions. The line between the Data Store Mechanism and the Repository represents the rules for adding, removing and changing contents of the Repository. The line between the Data Store Mechanism and the Extended Service Provider represents a mapping of the extended services to the Data Store services. The mapping is typically implementation-specific and therefore not part of the scope of this part of ISO 16100.

NOTE 2 The boldfaced elements in this figure are specifically addressed in this part of ISO 16100.

NOTE 3 The contents in the Repository are stored as XML files.

NOTE 4 The ESI access point is represented elsewhere in this part of ISO 16100 by the object *ServiceAccessPoint*.

NOTE 5 The Type 1 CPI service group, which is briefly described in ISO 16100-3:2005, 5.4, includes Type 1 matcher service.

NOTE 6 The Type 2 CPI service group, which is briefly described in ISO 16100-3:2005, 5.4, does not include Type 2 matcher services, which are part of the Extended Matcher Group.

NOTE 7 The Type 3 CPI service group is briefly described in ISO 16100-3:2005, 5.4.

Figure 1 — Extended Service Provider service sets

All services have the following characteristics:

a) when a service is conducted, there is one service provider and one service user, and no other third party is involved;

b) the service user initiates all service invocations, which are distinct from the lower communications layer service invocations;

c) a service user invocation is always accompanied by a response from the service provider;

d) any service user invocation and the corresponding response(s) are conducted in a bounded time frame, as determined by the service user or the service provider or both;

e) a service invocation at a service access point is processed when a response to a prior service invocation is completed;

f) an invocation is made for a single service; there is no invocation for a service group;

g) a service to register additions and updates to the repository uses the data store mechanism in Figure 1;

h) the state of an object in the repository is one of the following:

1) stored: an object is stored in the repository after a creation request;

2) registered: an object is registered into the repository after it is conformance tested;

3) deleted: an object is deleted from the repository after a deletion request.

5.2 ESI service set

The generic ESI services provided by an ESP can be organized in specific combinations into a CPI Group Type 1 (see ISO 16100-3) and the four service groups listed below, which are described in more detail in Clause 6. Other service groups can exist, e.g. MDM, MDDs and MDD objects, but are not defined in this part of ISO 16100.

a) The CPTI Group, which includes CPI Group Type 3, allows the following services:

1) create a new capability profile template;

2) access a capability profile template;

3) modify a capability profile template;

4) conformance test a capability profile template;

5) register an MSU capability profile;

6) delete an MSU capability profile.

b) The CPI Group Type 2 (see ISO 16100-3) allows the following services:

1) create a new MSU capability profile or a new requirement capability profile;

2) access an MSU capability profile or a requirement capability profile;

3) modify an MSU capability profile or a requirement capability profile;

4) conformance test a capability profile;

5) register a requirement capability profile or a requirement capability profile;

6) delete a requirement capability profile or a requirement capability profile.

c) The CCSI Group allows the following services:

1) create a new capability class structure;

2) access a capability class structure;

3) modify a capability class structure;

4) conformance test a capability class structure;

5) register a capability class structure;

6) delete a capability class structure.

d) The Extended Matcher Group allows the following services:

1) access an MSU capability profile or a requirement capability profile;

2) match two capability profiles, each referencing a different capability class structure.

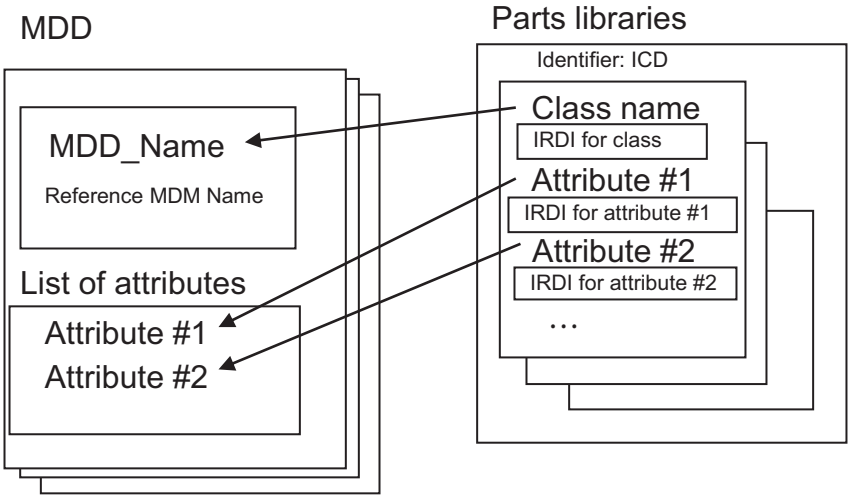
5.3 Dictionary Import Service Interface

5.3.1 Parts libraries imported to a repository

The Dictionary Import Service Interface enables the importing of a library (e.g. PLIB dictionary or OTD) to a repository.

5.3.2 Relationship between parts libraries and MDDs

A parts library imported to a repository can be used as part of MDDs in capability profiling. MDDs in an MDM include the definition of manufacturing activities. Since a PLIB dictionary and an OTD includes definitions of technical terms, it can be used in capability profiling in the same manner as MDDs are used. All classes and attributes in a PLIB dictionary or OTD can be identified by an unambiguous identifier in accordance with ISO 29002-5 and ISO 22745-13. The unambiguous identifier is a form of international registration data identifier (IRDI) as specified in ISO/IEC 11179-5. Classes and attributes within a PLIB dictionary can also be specified by a combination of a dictionary and a BSU, a code for each PLIB class and an attribute internal to the dictionary. A BSU for an attribute in a PLIB dictionary is the combination of the attribute code and the BSU of the parent class. For an OTD, it is not necessary to reference the BSU of the parent class, since ISO 22745 is classification-neutral and properties are defined independently of classes in an OTD. The relationship between a PLIB dictionary or OTD and an MDD is determined by mapping the unambiguous identifier for each MDD and attribute of an MDD, as shown in Figure 2.



NOTE 1 This figure is not in accordance with UML conventions.

NOTE 2 In ISO 13584 the term “property” is used instead of “attribute”.

Figure 2 — Relationship between parts libraries and MDDs

5.3.3 Mapping from a PLIB element to an MDD

The element “MDD_Name” in Figure 2 shall have the following extended attributes:

— “dictionary id”,

— “parent”,

— “BSU”,

— “version”, and

— “revision”.

The attribute set of “dictionary id”, “parent” and “BSU” can be used as an identifier of the MDD.

According to Figure 2, attributes in an MDD refer to attributes of items in a PLIB dictionary. An MDD corresponds to an element in a PLIB dictionary. Attributes belonging to the same element would be associated with one MDD. See Annex E.

6 Extended Service Interface

6.1 CPTI Group services

6.1.1 Scenarios handled by the CPTI Group

The CPTI Group handles the following capability profile template scenarios:

a) create a capability profile template for a specific class structure, either by partially filling the generic formal structure of the capability profile template or by modifying an existing capability profile template with a new capability profile template ID using the MDDs in the MDM, and then receive a capability profile template from a service provider;

b) request a capability profile template from a repository based on a capability profile template ID and then receive a capability profile template from a service provider;

c) modify a capability profile template, according to either user requirements or the results from a conformance test, and receive a modified capability profile template from a service provider;

d) test a capability profile template against the capability profile template conformance criteria found in ISO 16100-5:2009, Clause 8, and receive either a positive, negative, or matching level response from a service provider;

e) register a tested capability profile template into a repository and receive a “registered” status from the service provider;

f) delete a capability profile template based on a capability profile template ID and receive a “deleted” status from a service provider.

6.1.2 Capability profile template creation

6.1.2.1 Creation process

The process for creating a capability profile template for a capability class of a particular capability class structure consists of configuring a generic formal structure of a capability profile template by either

a) filling in specific values for certain attributes of certain elements in the generic formal structure of the capability profile template as defined in ISO 16100-5:2009, 6.3, or

b) modifying previously filled-in values in an existing capability profile template.

6.1.2.2 Capability profile template configuration

A generic formal structure of the capability profile template may be either partially or completely filled according to the application requirement.

For any capability profile template, the following attributes and elements shall be filled in or modified:

a) attributes “id” and “name” in the element “Capability Profile Template”;

b) attribute “domain_Name” in the element “Reference_MDM_Name”;

c) attribute “format_name” in the element “MDD_Description_Format” with one of the following four values:

   1) “Set_Of_MDD_Objects”;

   2) “List_Of_MDD_Objects”;

   3) “Time_Ordered_MDD_Objects”;

   4) “Event_Ordered_MDD_Objects”;

d) attributes “name” and “action” in the element “MDD_Name”.

Each value filled in or modified for the attributes “id” in 6.1.2.2 a), “domain_Name” in 6.1.2.2 b) and “name” in 6.1.2.2 d) shall be unique.

A capability profile template shall be used to create a capability profile associated with a capability class.

6.1.2.3 *createTemplate* service

6.1.2.3.1 Template based on a generic formal capability structure

The *createTemplate* service shall allow a template user to create a template based on a generic formal capability structure. When creating a template based on a formal capability structure, the *createTemplate* service uses at a minimum the *requestBlankTemplate*, *returnBlankTemplate*, *processFilledTemplate* and *returnProcessingResult* services. The *createTemplate* service shall consist of the following steps:

a) the template user invokes the *requestBlankTemplate* service of the *ServiceAccessPoint* object, in which there are no parameters associated with the *requestBlankTemplate*;

b) the service provider invokes the *returnBlankTemplate* service of the *ServiceAccessPoint* object, in which the parameters of the *returnBlankTemplate* service are blank template and creation error;

c) the template user fills in the blank template using MDDs of the MDM and then invokes the *processFilledTemplate* service of the *ServiceAccessPoint* object, in which the parameter of the *processFilledTemplate* service is template ID;

d) the service provider checks the uniqueness of the template ID and then invokes the *returnProcessingResult* service of the *ServiceAccessPoint* object, in which the parameters of the *returnProcessingResult* service are ID check error and storage error.

Figure 3, above the dotted line, provides a UML sequence diagram of the mandatory steps for the creation of a template from a formal template structure.

6.1.2.3.2 Template created by modifying an existing capability profile template

The *createTemplate* service shall allow a template user to create a template modified from an existing template. When creating a template modified from an existing template, the *createTemplate* service uses at a minimum the *requestExistingTemplate*, *returnExistingTemplate*, *processModifiedTemplate* and *returnProcessingResult* services. The *createTemplate* service shall consist of the following steps:

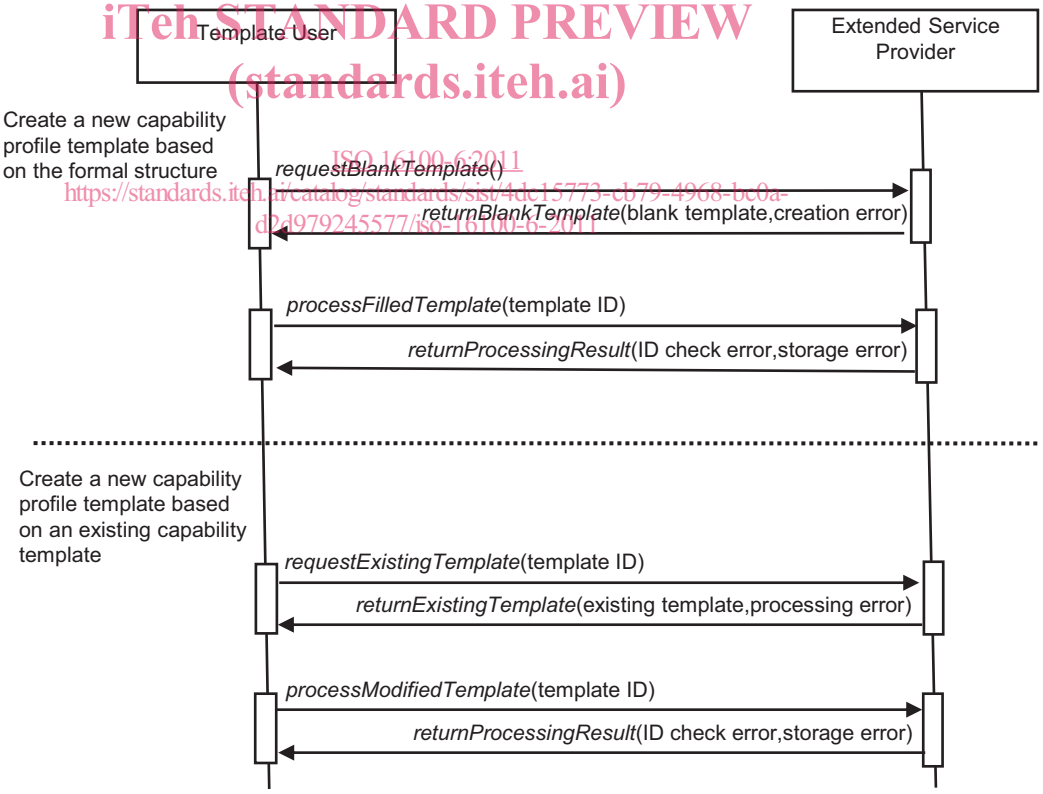
a) the template user invokes the *requestExistingTemplate* service of the *ServiceAccessPoint* object, in which the parameter of the *requestExistingTemplate* service is template ID;

b) the service provider invokes the *returnExistingTemplate* service of the *ServiceAccessPoint* object, in which the parameters of the *returnExistingTemplate* service are existing template and processing error;

c) the template user modifies the existing template and then invokes the *processModifiedTemplate* service of the *ServiceAccessPoint* object, in which the parameter of the *processModifiedTemplate* service is template ID;

d) the service provider checks the uniqueness of the template ID and then invokes the *returnProcessingResult* service of the *ServiceAccessPoint* object, in which the parameters of the *returnProcessingResult* service are ID check error and storage error.

Figure 3, below the dotted line, provides a UML sequence diagram of the mandatory steps for the creation of a template from an existing template.



NOTE 1 The dotted line separates the two creation cases.

NOTE 2 Template creation is always preceded by the preliminary step of registering and verifying unique template identifiers, which is outside the scope of ISO 16100.

Figure 3 — *createTemplate* service