TECHNICAL REPORT

ISO/TR 16161

First edition
2019-06

Automation systems and integration — Use case of capability profiles for cooperation between manufacturing software units

Reference number
ISO/TR 16161:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

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

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

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

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

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

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

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

Introduction

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) the 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 a manufacturing software unit (MSU) in 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.

This document describes an application of ISO 16100. Manufacturing software agents, which are one type of MSU, achieve interoperation using capability profiles specified in ISO 16100.

This document describes message language and protocol for software agents to collaborate with each other to emerge systems function. Presenting the MSU capability profile defined in ISO 16100-3, the agents mutually recognize the capability of manufacturing activity and recognizable messages. Software agents that need manufacturing activities are called customers, and agents that provide manufacturing activities are called performers. Customers describe the request messages for manufacturing activities by the message language. Performers describe the report messages of the result of the manufacturing activities by the message language.

Agent Communication Language (ACL), proposed by the Foundation for Intelligent Physical Agents (FIPA), is a message language exchanged with multi agents, and the protocol that defines the sequence of messages is a protocol to which the framework of interaction protocol is applied and identifies the agent in order to use the ontology prescribed by FIPA. By contrast, this document describes the protocol and message language in which the software agent acting as a customer and the software agent acting as a performer interact in a one-to-one manner and each software agent is identified using a capability profile specified in ISO 16100-3. Therefore, the protocol and message language described in ACL and in this document are different.

Automation systems and integration — Use case of capability profiles for cooperation between manufacturing software units

1 Scope

This document describes an approach for using ISO 16100 to achieve cooperation between software agents by exchanging manufacturing software unit (MSU) capability profiles. The exchanged profiles among agents describe the manufacturing capabilities requested by the requester and to be fulfilled by the performer.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

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

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

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

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

3.1
customer
requester of the manufacturing activity

3.2
C-subsystem
manufacturing software unit requesting the manufacturing activity

3.3
performer
provider of the manufacturing activity

3.4
P-subsystem
manufacturing software unit providing the manufacturing activity

3.5
capability profile service provider
software that implements the capability profile interface

[SOURCE: ISO 16100-3:2005, 3.1.2]

3.6
service provider
entity that plays the role of *capability profile service provider* (3.5) and is responsible for preparing and delivering a pair of *customer* (3.1) and *performer* (3.3)

4 Abbreviated terms

MSU Manufacturing Software Unit

UML Unified Modelling Language

URI Uniform Resource Identifier

5 Dialogue between C-subsystem and P-subsystem

The interaction that occurs between the orderer and the contractor is replaced by the dialogue between a customer and a performer. The dialogue between a customer and a performer can be represented by a sequence of actions associated with the order. The sequence of actions can be represented by the state transition diagram of dialogue. In a production system, the customer is the C-subsystem and the performer is the P-subsystem.

Figure 1 shows a dialogue that occurs between a C-subsystem and a P-subsystem. In Figure 1, "C: Request" is the operation of the request to a P-subsystem from a C-subsystem. "P: Promise" is an action promising that the P-subsystem has accepted the request to the C-subsystem. "P: Decline" is an action that the P-subsystem declines with respect to the request to the C-subsystem. Sometimes the P-subsystem offers a proposal. The C-subsystem accepts this proposal. The transition from state 2 to state 2’ shows a sequence of these actions. On the other hand, C-subsystem can offer another proposal for the proposal offered by the P-subsystem. The transition from state 2’ to state 2 shows a sequence of these actions.

It is assumed that the dialogue between the orderer and the contractor can be expressed by the state transition diagram of Figure 1, and messages are designed in accordance with this assumption. The following is a description of each state:

— State 1: initial state

— State 2: offer an order

— State 3: accept the order

— State 4: complete the order

— State 5: final state

— State 2': offer a proposal

— State 3', 2'', 3'': final state

Annex E gives an example of customization of a state transition diagram of dialogue.

Annex F proposes a solution for handling state transitions of dialogues, messages and capacity profile.

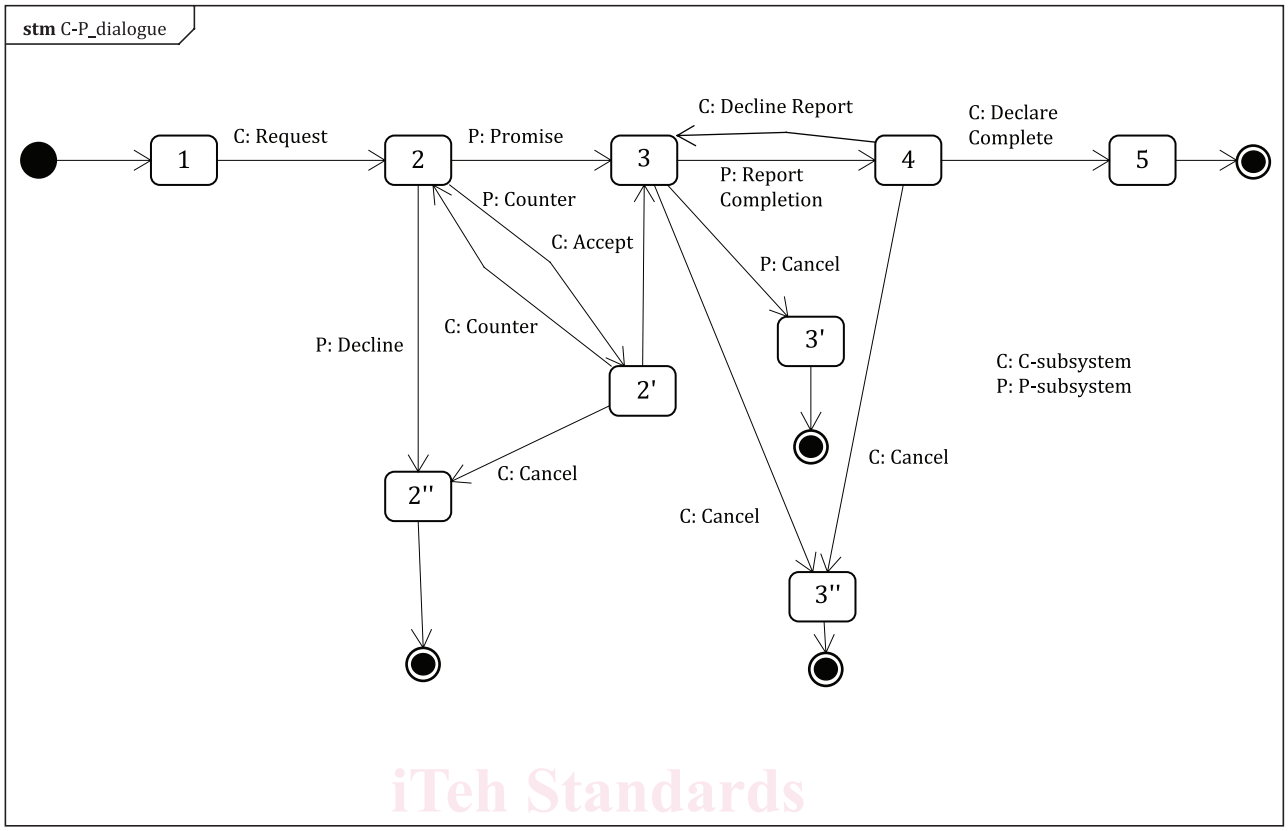

Figure 1 — State transition diagram of dialogue

6 Procedure until starting the dialogue between C-subsystem and P-subsystem

6.1 Overview

C-subsystem acquires the URIs of the P-subsystem that can perform a certain manufacturing activity from the service provider and the P-subsystem requested to perform the activity acquires the URIs of the C-subsystem which requests to perform the activity from the service provider. C-subsystem and P-subsystem start dialogue using the obtained URI respectively. When the C-subsystem sends the first Request message to the P-subsystem, the dialogue shown in Figure 1 starts and exchanges messages according to the procedure in Figure 1. This document assumes centralized control of bids by service providers which are defined in Clause 3. Other bidding processes, such as the distributed bidding process, are future works and are not described in this document.

6.2 Procedure for identifying the dialogue partner using capability profile

The procedure for identifying the dialogue partner and establishing the dialogue is as follows.

a) C-subsystem and P-subsystem register each capability profile in advance to the service provider. The details of the activities requested by C-subsystem are described in the capability profile of C-subsystem, and the details of the activities that P-subsystem can perform in the capability profile of P-subsystem are described.

b) The C-subsystem acquires the capability profile of the P-subsystem satisfying its own request from the service provider, and the P-subsystem acquires the capability profile of the requesting C-subsystem from the service provider. On matching of the capability profile, this procedure conforms to the procedure specified in the ISO 16100 series.

c) Before starting the dialogue in Figure 1, there are two cases of message exchanges between C-subsystem and P-subsystem. First case is that the C-subsystem sends a Notify message to the URI of the acquired P-subsystem, and the P-subsystem that received the Notify message immediately sends an Ask Response message to the C-subsystem. The other case is that the P-subsystem sends an Ask Response message even if there is no Notify message from the C-subsystem. The P-subsystem can repeatedly send the Ask Response message periodically or at an arbitrary timing until the response message is sent from the C-subsystem to the Ask Response message.

d) The dialogue begins when C-subsystem sends a Request message to P-subsystem. Subsequently, C-subsystem and C-subsystem exchange messages according to Figure 1.

The MSU realizes many different functions. The concern here is a communication function including how the C-subsystem and the P-subsystem establish a dialogue. Therefore, what is described here focuses on the communication function between the MSU as the role of the C-subsystem and the MSU as the role of the P-subsystem and the use of the capability profile in the communication. As Registration of capability profile in Figure 2 shows, before starting interaction between MSUs, each MSU needs to register each capability profile in advance with the service provider. The service provider determines the relationship between customer and performer between MSUs and confirms that both are connectable. This document assumes that the service provider has finished associating customer and performer between MSUs and that the service provider confirms that it can connect between the associated MSUs. The customer-performer model is illustrated in Annex C.

Figure 2 shows that the sequence diagram named 'C-P cooperation' consists of the sequence diagram named 'RegistationOfCapabilityProfiles', the sequence diagram named 'C-P dialogue', and the sequence diagram named 'C-P dialogue base'. The sequence diagram named 'RegistationOfCapabilityProfiles' shows that C-subsystem and P-subsystem register their capability profiles in the service provider beforehand. Since this document is an application example of capability profile in this document, details of 'RegistationOfCapabilityProfiles' and service provider are not explained. The sequence diagram named 'Prepare C-P dialogue' in Figure 2 consists of the sequence diagram named 'get capability profile' and the sequence diagram named 'send notify'. The first 'get capability profile' shows the interaction between MSU and the service provider. MSU acting as C-subsystem acquires capability profile of MSU acting as P-subsystem from the service provider. On the other hand, this P-subsystem obtains the capability profile of C-subsystem which is a partner of P-subsystem from the service provider. The second 'send notify' has two cases. First case is that when C-subsystem sends Notify message to P-subsystem and P-subsystem sends Ask Response to C-subsystem, the other case is when the P-subsystem sends Ask Response to the C-subsystem even if the P-subsystem does not receive the Notify message from the C-subsystem. 'Ask for Request message' in 'send notify' shows that P-subsystem can repeatedly send AskResponse until the C-subsystem responds with Request message.

The sequence diagram named C-P dialogue base in Figure 2 is a sequence diagram of the communication between the C-subsystem and the P-subsystem of Figure 1. As a result, only the C-subsystem needs to acquire the URI of P-subsystem, so parameter p is implementation-dependent and is not specified in this document, in particular. The sequence diagram of C-P dialogue base is described in Annex B.

Message exchanged between C-subsystem and P-subsystem is specified in Clause 7. Message communication between C-subsystem and P-subsystem is explained in more detail in Clause 8.

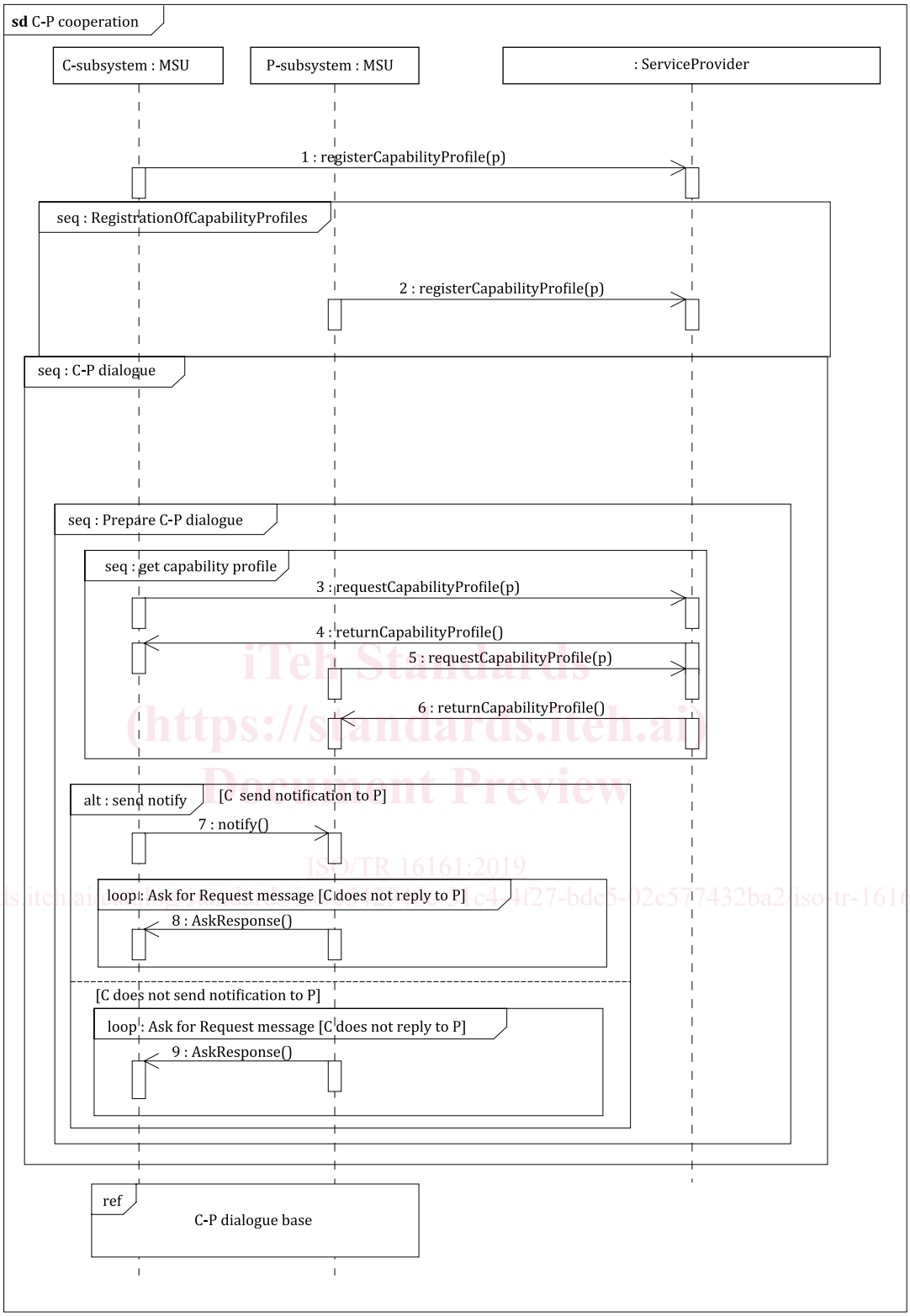

Figure 2 — Interaction overview diagram of dialogue

6.3 Capability profile

6.3.1 General

Using the capability profile, a MSU identifies the partner who dialogues with. Even if a C-subsystem sends Notify mess Annex B ages at first, every dialogue is started by Invitation message of a P-subsystem. Therefore, the communication channels of the C-subsystem need to be defined. In a case where the C-subsystem sends a Notify message, the address of P-subsystem is needed.

6.3.2 Description of specific part

Above information is described in the specific part of capability profile according to the template of the capability profile (see ISO 16100-3). The template of the specific part which added necessary elements in this document and an example of this specific part are shown in 6.4.

The XML tags added to the specific part are <Activity> and <Performatives>. These are used to describe the information needed by the adapters.

Tag <Activity> represents a manufacturing activity using a verb that the MSU provides capability. In Figure B.1 "Overall Structure of Production Management System", production activities for each domain implemented as MSU are represented by verbs such as Sell, Make, Buy, Bill, Pay, and so on. The <Channel> tag in the tag <InformationExchange> in the <Activity> section has information on communication between MSUs. The communication method is defined by the attribute name 'type' of this tag, and the URI is defined by the attribute name 'address'. The communication method will be explained in Clause 8.

Tag <Performatives> represents performative verbs (tag <Receive>) that can be receive and performative verbs (tag <Send>) of the MSU that can be sent, and tag <MessageFormat> gives information of data types in each message item.

6.4 Capability profile template and examples

In this clause, the template which added the necessary element in the specific part of the capability profile (see ISO 16100-3) is shown in 6.4.1, and an example of the capability profile applying this template is shown in 6.4.2.

6.4.1 Capability profile template

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="CapabilityProfiling">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="type">
          <xs:complexType>
            <xs:attribute name="id" type="xs:string" use="required"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="CapabilityProfile">
          <xs:complexType>
```

```
            <xs:sequence>
              <xs:element name="pkgtype">
                <xs:complexType>
                  <xs:attribute name="version" type="xs:string"
form="unqualified"/>
                </xs:complexType>
              </xs:element>
              <xs:element name="Common" type="CommonPartType"/>
              <xs:element name="Specific" type="SpecificPartType"/>
            </xs:sequence>
            <xs:attribute name="date" type="xs:string" form="unqualified"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="CommonPartType">
    <xs:sequence>
      <xs:choice>
        <xs:element name="Requirement">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ID" type="xs:string"/>
            </xs:sequence>
            <xs:attribute name="id" type="xs:string" form="unqualified"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="MSU_Capability">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ID" type="xs:string"/>
            </xs:sequence>
            <xs:attribute name="id" type="xs:string" form="unqualified"/>
```

```
      </xs:complexType>
    </xs:element>
  </xs:choice>
  <xs:sequence maxOccurs="unbounded">
    <xs:element name="ReferenceCapabilityClassStructure">
      <xs:complexType>
        <xs:attribute name="id" type="xs:string" form="unqualified"/>
        <xs:attribute name="name" type="xs:string" form="unqualified"/>
       <xs:attribute name="version" type="xs:string" form="unqualified"/>
        <xs:attribute name="url" type="xs:string" form="unqualified"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="TemplateID">
      <xs:complexType>
        <xs:attribute name="ID" type="xs:string" form="unqualified"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:element name="Version">
    <xs:complexType>
      <xs:attribute name="major" type="xs:string" form="unqualified"/>
      <xs:attribute name="minor" type="xs:string" form="unqualified"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="Owner">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="xs:string" minOccurs="0"/>
        <xs:element name="street" type="xs:string" minOccurs="0"/>
        <xs:element name="city" type="xs:string" minOccurs="0"/>
        <xs:element name="zip" type="xs:string" minOccurs="0"/>
        <xs:element name="state" type="xs:string" minOccurs="0"/>
        <xs:element name="country" type="xs:string" minOccurs="0"/>
```