INTERNATIONAL STANDARD

ISO 16100-4

First edition
2006-12-15

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

Part 4:
Conformance test methods, criteria and reports

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

Partie 4: Méthodes d'essai, critères et rapports de conformité

Reference number
ISO 16100-4:2006(E)

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 electro-technical standardization.

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

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 part of ISO 16100 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 16100 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 5, *Architecture, communications and integration frameworks.*

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

The following part is under preparation

Part 5: Methodology for profile matching using multiple capability classes

Introduction

The motivation for ISO 16100 stems from the industrial and economic environment noted in the ISO/TC 184/SC5 strategic plan, in particular:

a) a growing base of vendor-specific solutions;

b) user difficulties in applying standards;

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

d) a 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 software capability profile. Its goal is to provide a method to represent the capability of manufacturing software relative to its role throughout the life cycle of a manufacturing application, independent of a particular system architecture or implementation platform.

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 4: Conformance test methods, criteria and reports

1 Scope

This part of ISO 16100 specifies the test method, the associated test criteria and the statement format used to evaluate and declare the degree of conformance of an implementation, i.e. a unit under test (UUT), to the requirements specified in other parts of ISO 16100.

This part of ISO 16100 includes definitions intended to assist a manufacturer or supplier (first party), a user or purchaser (second party), or an independent body (third party) to perform the assessment for type evaluation.

This part of ISO 16100 contains the following:

— an enumeration of those conformance aspects that can be used to determine whether an implementation conforms to ISO 16100;

— a definition of the conformance tests and statements used in declaring which aspects are met by an implementation;

— a description of the aspects to be included in a conformance statement;

— a set of rules to select valid or invalid combinations of aspects when they are combined.

The following topics are not addressed in this part of ISO 16100:

— matters relating to marks or labels of conformance, certificates of conformance or manufacturers' or suppliers' declarations of conformance;

— dates of implementation or allocation of responsibilities to various parties making use of ISO 16100;

— requirements for production, execution or delivery procedures, unless it is impossible to specify adequately the conforming product, process or service, respectively, without doing so;

— requirements for quality control during production, execution or delivery of the product, process or service, respectively.

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 16100-1:2002 *Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 1: Framework*

ISO 16100-2:2003 *Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 2: Profiling methodology*

ISO 16100-3:2005	*Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 3: Interface services, protocols and capability templates*
REC-xml-20000814	*Extensible Markup Language (XML) 1.0 Ed. 2 W3C Recommendation*
REC-xmlschema-1-20010502	*XML Schema Part 1: Structures*
REC-xmlschema-2-20010502	*XML Schema Part 2: Datatypes*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply. Some of these terms and definitions have been taken verbatim or were adapted from other standards. In such cases this is indicated in brackets with the specific part and subclause of the standard given.

3.1
abstract test case
specification, encapsulating at least one test purpose, that is independent of implementation platform, parameter values, and method

[adapted from ISO 10303-31:1994, 3.2.1]

3.2
abstract test suite
set of abstract test cases

3.3
capability
<software> set of functions and services with a set of criteria for evaluating the performance of a capability provider

[ISO 16100-1:2002, 3.3]

NOTE This definition differs from that given in ISO 15531-1 and ISO/DIS 19439, where capability is defined as the quality of being able to perform a given activity. See IEC 62264-1 for a general definition of capability.

3.4
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

[ISO 16100-2:2003, 3.3]

3.5
capability profiling
selection of a set of offered services defined by a particular interface within a software interoperability framework

[ISO 16100-1:2002, 3.4]

3.6
conformance
conformity
relation between a specification and a real implementation that is realized when any proposition that is true in the specification is also true in the implementation

EXAMPLE A profile implementation is in conformance with the template specification that is created according to the rules in ISO 16100.

3.7
conformance point
specific requirement contained in a set of subclauses in ISO 16100 that are used as a basis to generate and perform a test to determine if an implementation is conformant

3.8
conformance statement
statement that identifies conformance points of a specification and the behaviour that must be satisfied at these points

[adapted from ISO/IEC 10746-2:1996, 15.1]

3.9
conformance testing
conformity assessment
testing of a candidate product for the existence of specific characteristics required by a standard in order to determine the extent to which that product is a conforming implementation

[ISO 10303-31:1994, 3.2.22]

3.10
conformance test report
document written at the end of the conformance assessment process, that provides the overall summary of the conformance of the UUT to the standard for which conformance testing was carried out, and that gives the details of the testing

[ISO 10303-31:1994, 3.2.23]

3.11
conforming implementation
implementation which satisfies the conformance requirements, consistent with the capabilities stated in the CSI

[adapted from ISO 10303-31:1994, 3.2 24]

3.12
executable test case
implementation of an abstract test case that is platform-dependent and is associated with parameter values and a specific test method

3.13
executable test suite
set of executable test cases

3.14
falsification testing
test method developed to find errors in the implementation

[adapted from ISO 10303-31:1994, 3.2.32]

3.15
interface
abstraction of the behaviour of an object that consists of a subset of the interactions of that object together with a set of constraints on when they may occur

[ISO 16100-3:2005, 3.3.3]

3.16
manufacturing software unit
class of software resource, consisting of one or more manufacturing software components, performing a definite function or role within a manufacturing activity while supporting a common information exchange mechanism with other units

[ISO 16100-1:2002, 3.12]

NOTE A manufacturing software unit can be modeled as a software object using a UML convention.

3.17
matcher
mechanism to compare an offered capability profile with a required capability profile.

[ISO 16100-3:2005, 3.1.6]

3.18
matching level
<profile> qualitative measure of how closely a capability profile of an MSU meets the software functional requirements of a manufacturing activity

[ISO 16100-3:2005, 3.1.7]

3.19
MSU interoperability
capability of a MSU to support a particular usage of an interface specification in exchanging a set of application information with another MSU

[ISO 16100-3:2005, 3.1.8]

3.20
profile
set of one or more base specifications or sub-profiles or both, 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

[ISO 16100-2:2003, 3.10]

3.21
reference capability class structure
schema representing a hierarchy of capability classes to be used for capability profiling

[ISO 16100-3:2005, 3.1.11]

3.22
template
schema for a manufacturing software capability profile

[ISO 16100-3:2005, 3.1.14]

3.23
unit under test
capability profile, capability template, capability class structure or profile matcher being evaluated to determine if it meets or provides specific characteristics described in ISO 16100

4 Abbreviated terms

ATC	Abstract test case
ATG	Abstract test group
ATS	Abstract test suite
CITI	Conformance information for testing the implementation
CSI	Conformance statement for the implementation
ETC	Executable test case
ETG	Executable test group
ETS	Executable test suite
MSU	Manufacturing software unit
UML	Unified Modeling Language
UUT	Unit under test
XIPI	eXtra information for platform implementation
XITI	eXtra information for testing the implementation
XML	eXtensible Markup Language

5 Conformance framework

5.1 Conformance testing

A UUT, such as capability profile, template, reference capability class structure, or profile matcher shall be called conforming if its externally visible behaviour fulfils specific conformance requirements in this part of ISO 16100.

Conformance testing shall be used to verify if an implementation meets the requirements of a standard or specification. Conformance testing is a necessary step toward achieving interoperability, but is not a guarantee for interoperability. Conformance testing provides developers and users the assurance and confidence that the conforming UUT behaves as expected, performs functions in a known manner, or possesses a prescribed interface or format.

The basic conformance testing strategy for ISO 16100 shall be falsification testing. Falsification testing subjects an implementation to various combinations of valid and invalid inputs, and compares the test outputs to the corresponding expected outputs as defined in the test criteria in order to determine the degree of conformance. When a test output does not match the expected output, the deduction that the implementation does not conform to the specification can be made. When the conformance testing output is true, it does not mean absolute conformance. Falsification testing shall only demonstrate non-conformance. The use of a greater variety of test inputs can increase the likelihood of conformance.

5.2 Types of UUTs

The interoperability for manufacturing software can be realized through the capability profiling method described in ISO16100-2. The key phases of this capability profiling method both for MSU capability profiling and required activity capability profiling are as follows:

a) create a capability class structure and register it in the database;

b) search for a capability class structure in the database according to the manufacturing application requirements;

c) select capability class from the reference capability class structure in the database;

d) create a capability template and register it in the database;

e) search for a capability template in the database corresponding to a capability class;

f) create a capability profile by filling in each field of the template and register it in the database;

g) match a MSU capability profile with a requirement profile using a profile matcher.

Before registering the UUTs in steps (a), (d) and (f), a conformance test associated with the UUT type shall be performed on the UUT.

The likelihood of interoperability of MSUs will be ensured when their respective capability profiles have been validated using a capability class structure, a capability class, and a capability profile template that have also been validated.

The four types of UUTs that shall undergo conformance testing to ensure the likelihood of interoperability are:

— reference capability class structure;

— capability template;

— capability profile;

— capability profile matcher.

5.3 Conformance test methodology

As shown in Figure 1, the following set of activities shall form a conformance testing process:

a) create CSI;

b) create ATC;

c) create ETC;

d) test UUT.

The process shall begin with the creation of a CSI based on analyzing the conformance points and conformance test criteria contained in ISO 16100.

Adding XITI and a CSI shall result in the creation of an ATC. XITI shall be UUT type-specific and shall include those items listed in Table 2 for each UUT type.

Each ATC shall be traceable back to a CSI and shall be implemented as a set of ETCs. For a particular test platform, extra information as listed in Table 3 shall be combined with the set of ETCs corresponding to an ATS to form an ETS.

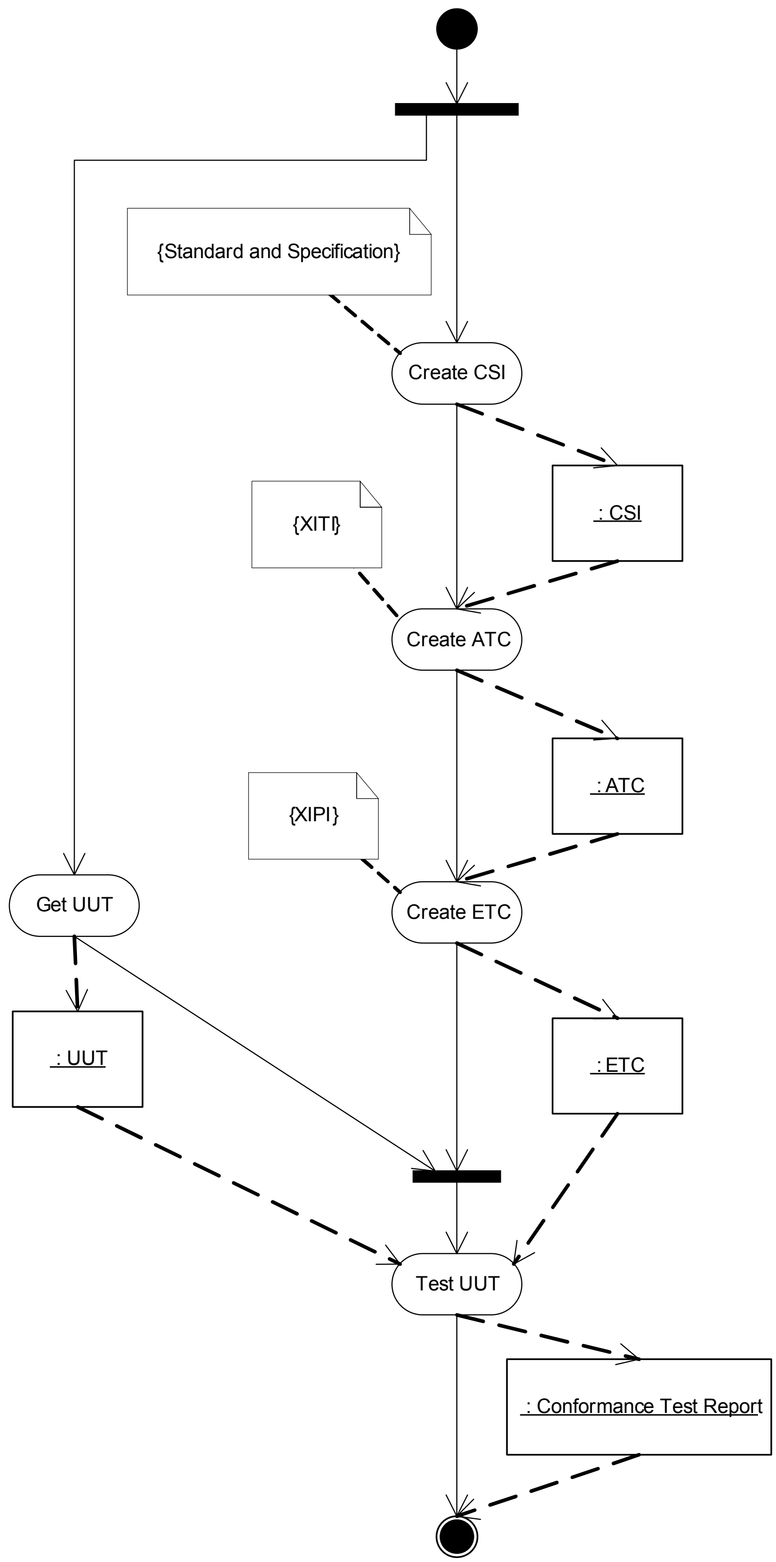

Figure 1 — Methodology for developing the conformance testing process

When an ETS is performed on a particular UUT the test suite result shall be the combined results from each ETC belonging to this particular ETS. The test result for each ETC shall either be a PASS or a FAIL condition. A UUT that fails a specific ETC shall imply that the UUT does not conform to the corresponding ATC that maps to a set of conformance points and associated test criteria. When the execution of at least one ETC belonging to an ETS results in a FAIL outcome then the execution of the ETS shall be considered as not in full conformance.

For each type of UUT, there are inputs and outputs for all activities of the conformance test. These inputs and outputs are detailed as follows:

a) "Create CSI" activity — see Table 1;

b) "Create ATC" activity — see Table 2;

c) "Create ETC" activity — see Table 3;

d) "Test UUT" activity — see Table 4.

Table 1 — "Create CSI" inputs and outputs

Input/output \ Type of UUT		Capability class structure	Capability template	Capability profile	Capability profile matcher
Input	Standard or specification	• Application domain ontology • Dictionary and taxonomy	• Capability class • Dictionary and taxonomy • Path continuity	• Template	• Set of elements in both well-defined MSU and requirement profiles
Output		• CSI for Capability class structure in Table 6	• CSI for template in Table 7	• CSI for profile in Table 8	• CSI for profile matcher in Table 9

Table 2 — "Create ATC" inputs and outputs

Input/output \ Type of UUT		Capability class structure	Capability template	Capability profile	Capability profile matcher
Input	CITI	• CITI	• CITI	• CITI	• CITI
	XITI	• Relative range and depth inside tree • UML diagrams for activity classes and relationships	• Capability class structure • Dictionary and taxonomy	• Template • Dictionary and taxonomy	• Template • Dictionary and taxonomy
Output	ATC	• ATC • ATS	• ATC • ATS (refer to Annex B)	• ATC • ATS (refer to Annex A)	• ATC • ATS (refer to Annex C)

Table 3 — "Create ETC" inputs and outputs

Input/output \ Type of UUT		Capability class structure	Capability template	Capability profile	Capability profile matcher
Input	ATC	• ATC • ATS	• ATC • ATS	• ATC • ATS	• ATC • ATS
	XIPI	• Platform • Application domain ontology • Dictionary and taxonomy	• Platform • Template schema	• Platform • Template	• Platform • Capability class structure
Output		• ETC • ETS	• ETC • ETS	• ETC • ETS	• ETC • ETS

Table 4 — "Test UUT" inputs and outputs

Input/output \ Type of UUT		Capability class structure	Capability template	Capability profile	Capability profile matcher
Input	ETC	• ETC • ETS	• ETC • ETS	• ETC • ETS	• ETC • ETS
	UUT	• Reference class structure file • Varied inputs for parameters	• Template file • Varied inputs for parameters	• Profile file • Varied inputs for parameters	• Matcher • Varied inputs for parameters
Output		• Conformance test report	• Conformance test report	• Conformance test report	• Conformance test report

6 Conformance testing process

6.1 "Create CSI" activity

6.1.1 Conformance statements for implementation

For each type of UUT a specific CSI set shall be used to identify a set of conformance points at which conformance tests can be performed using a corresponding set of conformance test criteria. A CSI shall have a structure as shown in Figure 2.

The CSI is used to have a better understanding for the conformance assessment and to identify the boundaries of the testing domain. The conformance statement shall be positive or negative. The statement indicating what should be done is positive, and the statement indicating what should not be done (prohibited) is negative.

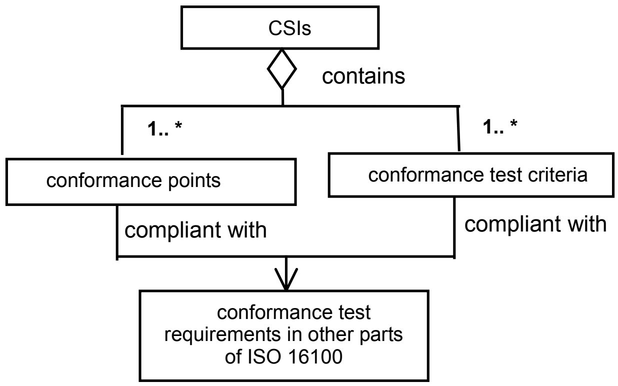

Figure 2 — Content of CSIs

6.1.2 Types of conformance points

A conformance point is a conformance requirement in a standard or a specification. At this point, the test should be made to see if the implementation meets a set of conformance criteria. There are four types of conformance points as shown in Table 5.

Table 5 — Types of conformance points

Type	Implemented in UUT	Shall be tested	ATC result
A	yes	yes	PASS
B	yes	yes	PASS or FAIL
C	optional	yes, if implemented	PASS
D	optional	yes, if implemented	PASS or FAIL

6.1.3 The CSI schema

Each CSI is a record in the conformance statement table that has the following columns:

a) Conformance set number — a unique identifier for each conformance set which is composed of a set of logically related conformance points;

b) Conformance point number — a unique identifier for each conformance point;

c) Conformance point description — a short description for the conformance point to be checked;

d) Specification reference — the clause/subclause and ISO 16100 part number from which the conformance point is derived;

e) Conformance point type — the type of conformance point according to Table 5;

f) Abstract test criteria — an expression of the expected behaviour.

6.2 "Create ATC" activity

For each type of UUT a set of ATCs based on a set of UUT-specific CSIs shall be used to test the UUT. Each ATC shall be related to the other conformance test elements shown in Figure 3 and have the following elements:

a) UUT type;

b) a conformance point identifier;

c) conformance point set identifier;

d) the test outcome for a conformance point.

Each ATC shall have an individual test purpose to verify and validate a certain UUT behaviour. ATCs shall be logically grouped into ATGs according to CSI sets. A complete set of ATGs shall form an ATS for a given type of UUT.

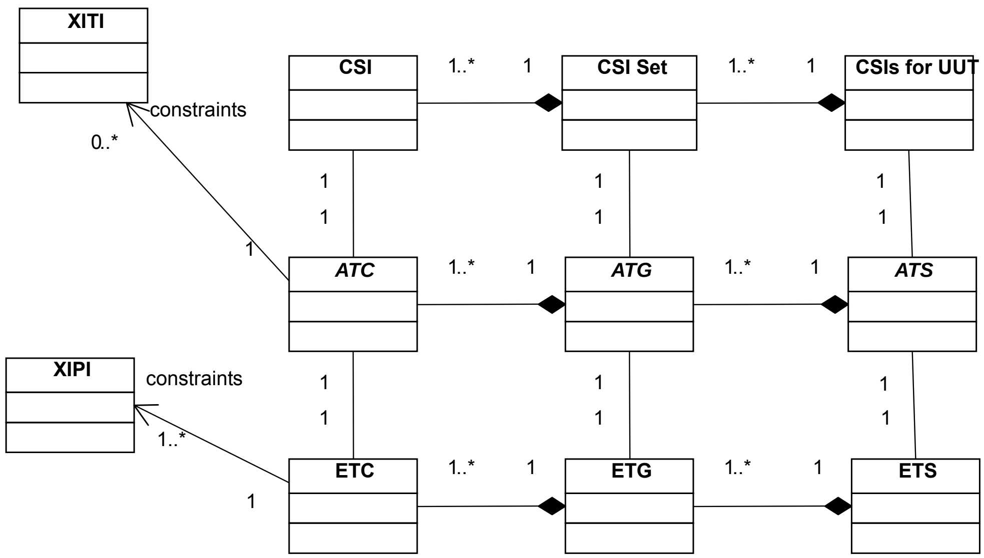

Figure 3 — Class diagram for conformance testing elements

6.3 "Create ETC" activity

For each type of UUT a set of ETCs based on UUT-specific ATCs shall be used to generate the platform-specific implementations of the ATCs. Platform-specific factors can include computer hardware, software, communication networks, and programming languages. ETCs shall form ETGs in the same manner as the corresponding set of ATCs form ATGs. A complete set of ETGs shall form an ETS for a given type of UUT, as shown in Figure 3.

6.4 "Test UUT" activity

For each type of UUT an ETS shall be used to conduct the conformance test. Test inputs shall include a UUT and other UUT-specific information required to perform each ETC belonging to the ETS. Test outputs shall include a conformance test report having the following elements:

a) a statement of the conformance level that shall contain one of the following values:

1) FULL CONFORMANCE — when all types of conformance points passed the ETS;

2) MINIMAL CONFORMANCE — when all type A and C conformance points passed the ETS;

3) NO CONFORMANCE — when any type A or C conformance point did not pass the ETS.

NOTE The statement of the conformance level may not be reliable if the ETS has not been properly derived from the ATS.

b) corresponding information that shall contain the list of conformance points, the outcome of each ETC test, and detailed informative messages in cases of MINIMAL CONFORMANCE and NO CONFORMANCE.

7 Conformance of UUTs

7.1 Conformance of a capability class structure

7.1.1 Reference capability class structure

As shown in Figure B.1 of ISO 16100-3:2005, a manufacturing application domain can be modeled as an activity tree structure, where each activity can be associated with a capability class that is provided by a MSU. The capability classes also form a reference capability class tree structure that maps one-to-one to the corresponding activity tree structure.

Figure 4 shows an example of a mapping of a capability class structure to an application activity tree.

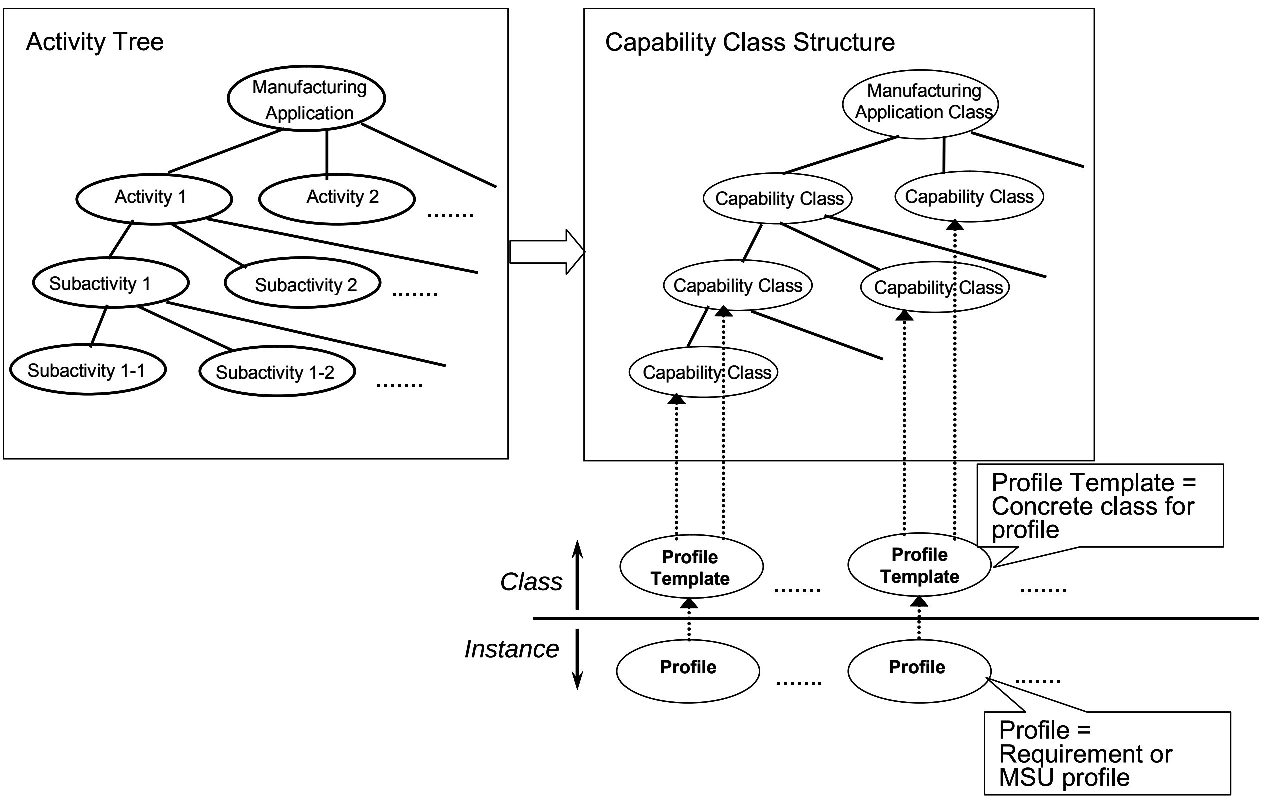

Figure 4 — Example mapping of a capability class structure to an application activity tree

7.1.2 CSIs of a reference capability class structure

Table 6 shall be used to generate the ATCs for the UUT type "reference capability class structure".

Table 6 — CSIs of a reference capability class structure

Conformance point or set number	Conformance point description	Specification reference	Conformance point type (see Table 5)	Abstract test criteria
Index_1	XML format	ISO 16100-2: 2003, 6.2	A	ReferenceClassStructure in XML schema format.
Index_2	Capability class tree structure	ISO 16100-2: 2003, 6.2		
index_2.1	Single parent per node	ISO 16100-2: 2003, 6.2	A	Exactly one parent per node, and no parent for the root node
index_2.2	Tree depth	ISO 16100-2: 2003, 6.2	D	Presence of a value for tree depth
index _2.3	Tree width	ISO 16100-2: 2003, 6.2	D	Presence of a value for tree width
index _2.4	Node ID	ISO 16100-2: 2003, 6.2.2	A	Presence and uniqueness of node identifier within a tree to distinguish the location of an activity function associated with a capability class within the tree structure, especially if the function (or capability class) is instantiated multiple times
Index_2.5	Tree structure ID	ISO 16100-3: 2005, 7.1.2[a]	A	Presence and uniqueness of the structure component
Index_2.6	Tree structure type	ISO 16100-3: 2005, 7.1.2[a]	A	Presence and uniqueness of the structure component
index_2.7	CapabilityClassID	ISO 16100-3: 2005, 7.2.2[b]	A	Presence of the ID

[a] The tree structure ID and tree structure type correspond to the element "CapabilityProfiling" in the schema in 7.1.2 of ISO 16100-3:2005.
[b] CapabilityClassID corresponds to element "TemplateID" in the schema in ISO 16100-3:2005, 7.2.2.

7.2 Conformance of a capability template

7.2.1 Capability template

The structure of capability class is described in ISO 16100-2:2003, 6.2.1. The capability template is described in clause 7 of ISO 16100-3:2005. The mapping of capability class to capability template is described in ISO 16100-2:2003, 6.3.

Conformance testing of a capability template shall involve verifying that the template is:

a) a well formed XML schema;

b) valid according to the specifications in clause 7 of ISO 16100-3:2005.

7.2.2 CSIs of a capability template

Table 7 shall be used to generate the ATCs for the UUT type "capability template".

Table 7 — CSIs of a capability template

Conformance point or set number	Conformance point description	Specification reference	Conformance point type (see Table 5)	Abstract test criteria
Index_1	XML format	ISO 16100-3: 2005, clause 7	A	Capability template in XML schema format
Index_2	Schema component header elements	ISO 16100-3: 2005, 7.1		
Index_2.1	XML version and target namespaces	ISO 16100-3: 2005, 7.1	A	Presence and location of XML version and target namespaces and values in conformance with ISO 16100-3:2005, 7.1
Index_2.2	CapabilityProfiling	ISO 16100-3: 2005, 7.1	A	Presence and location of schema component
index_2.2.1	Type	ISO 16100-3: 2005, 7.1	A	Presence and location of schema component
Index_2.2.2	CapabilityProfile	ISO 16100-3: 2005, 7.1	A	Presence and location of schema component
Index_2.2.2.1	Pkgtype and version	ISO 16100-3: 2005, 7.1	A	Presence and location of schema components
Index_2.2.2.2	CommonPartType and SpecificPartType	ISO 16100-3: 2005, 7.1	A	Presence and location of schema components

Table 7 — CSIs of a capability template (*continued*)

Conformance point or set number	Conformance point description	Specification reference	Conformance point type (see Table 5)	Abstract test criteria
Index _3	CommonPartType	ISO 16100-3: 2005, 7.2		
Index_3.1	Choice of capability profile type	ISO 16100-3: 2005, 7.2	A	Presence and location of schema component and value of either "Requirement" or "MSU_Capability"
Index_3.2	ReferenceCapability ClassStructure	ISO 16100-3: 2005, 7.2.1	A	Presence and location of attributes "id", "name", "version", and "url"
Index_3.3	TemplateID	ISO 16100-3: 2005, 7.2.1	A	Presence and location of schema component
Index_3.4	Addtional ReferenceCapability ClassStructure and TemplateID pair(s)	ISO 16100-3: 2005, 7.2.1	C	Presence and location of both components of pair satisfying criteria in index_3.2 and index_3.3
Index_3.5	Version	ISO 16100-3: 2005, 7.2	C	Presence and location of schema component
Index_3.6	Owner	ISO 16100-3: 2005, 7.2	C	Presence and location of schema component
Index_3.7	ComputingFacilities	ISO 16100-3: 2005, 7.2	C	Presence of schema components "Processor0", "OperatingSystem0", "Language", "Memory", "DiskSpace"
Index_3.8	Additional ComputingFacilities elements	ISO 16100-3: 2005, 7.2	C	Presence and location of schema components satisfying criteria in index_3.7
Index_3.9	Performance	ISO 16100-3: 2005, 7.2	D	Presence and location of attributes "ElapsedTime" and "TransactionsPerUnit Time"

Table 7 — CSIs of a capability template (*continued*)

Conformance point or set number	Conformance point description	Specification reference	Conformance point type (see Table 5)	Abstract test criteria
Index_3.10	Additional Performance elements	ISO 16100-3: 2005, 7.2	D	Presence and location of schema components satisfying criteria in index_3.9
Index_3.11	ReliabilityData	ISO 16100-3: 2005, 7.2	D	Presence and location of schema components "UsageHistory", "Shipments", "IntendedSafetyInteg rity", and "Certification"
Index_3.12	Additional ReliabilityData elements	ISO 16100-3: 2005, 7.2	D	Presence and location of schema components satisfying criteria in index_3.11
Index_3.13	SupportPolicy	ISO 16100-3: 2005, 7.2	D	Presence and location of attribute "index"
Index_3.14	Additional SupportPolicy elements	ISO 16100-3: 2005, 7.2	D	Presence and location of schema components satisfying criteria in index_3.13
Index_3.15	PriceData	ISO 16100-3: 2005, 7.2	D	Presence and location of attributes "invest", "annualSupport", and "unit"
Index_3.16	Additional PriceData elements	ISO 16100-3: 2005, 7.2	D	Presence and location of schema components satisfying criteria in index_3.15

7.3 Conformance of a capability profile

7.3.1 Capability profile

A capability profile is the capability template with, at a minimum, a profile name instantiated. Other items are fulfilled according to the specification level.

Conformance testing of a capability profile shall involve verifying that the profile is:

a) a well formed XML document;

b) is valid according to the template specification in clause 7 of ISO 16100-3:2005.

7.3.2 CSIs of a capability profile

Table 8 shall be used to generate the ATCs for the UUT type "capability profile".

Table 8 — CSIs of a capability profile

Conformance point and set number	Conformance point description	Specification reference	Conformance point type (see Table 5)	Abstract test criteria
index_1	XML Format	ISO 16100-3: 2005, clause 7	A	Capability Profile in XML format.
index _2	Schema component header elements	ISO 16100-3: 2005, 7.1.2	A	
Index_2.1	XML version and target namespaces	ISO 16100-3: 2005, 7.1.2	A	Presence and location of XML version and target namespaces and values in conformance with ISO 16100-3:2005 7.1. 2
Index_2.2	CapabilityProfiling	ISO 16100-3: 2005, 7.1.2	A	Presence and location of schema component
Index_2.2.1	Type	ISO 16100-3: 2005, 7.1.2	A	Presence of attribute “id” The value of id in “string” type , ”unqualified” form and distinguish a requirement profile or a MSU profile.
Index_2.2.2	CapabilityProfile	ISO 16100-3: 2005, 7.1.2	A	Presence of components “PKgtype”, “Common”, “Specific”. Presence of attribute “date”. The value of date in “string” type and ” unqualified” form.
Index_2.2.2.1	Pkgtype and version	ISO 16100-3: 2005, 7.1.2	A	The value of Pkgtype version in “string” type and ”unqualified” form.

Table 8 — CSIs of a capability profile (*continued*)

Conformance point and set number	Conformance point description	Specification reference	Conformance point type (see Table 5)	Abstract test criteria
Index_2.2.2.2	CommonPartType and SpecificPartType	ISO 16100-3: 2005, 7.1.2	A	The value of Common in CommonPartType The value of Specific in SpecificnPartType
Index _3	CommonPartType	ISO 16100-3: 2005, 7.2.2	A	
Index _3.1	Profile ID	ISO 16100-3: 2005, 7.2.2	A	The value in either Requirement ID or MSU_Capability ID with “string” type and ”unqualified” form
Index_3.2	ReferenceCapability ClassStructure	ISO 16100-3: 2005, 7.2.2	A	Presence and location of attributes "id", "name", "version", and "url"
Index_3.2.1	id	ISO 16100-3: 2005, 7.2.2	A	The value in “string” type and ”unqualified” form
Index_3.2.2	name	ISO 16100-3: 2005, 7.2.2	A	The value in “string” type and ”unqualified” form.
Index_3.2.3	version	ISO 16100-3: 2005, 7.2.2	A	The value in “string” type and ”unqualified” form.
Index_3.2.4	uri	ISO 16100-3: 2005, 7.2.2	A	The value in “string” type and ”unqualified” form.
Index_3.3	TemplateID	ISO 16100-3: 2005, 7.2.2	A	The value of attribute ID in “string” type and “unqualified”form, Typically, the value is NULL if capability profile matching is required for the full capability class structure
Index_3.4	Addtional ReferenceCapability ClassStructure and TemplateID pair(s)	ISO 16100-3: 2005, 7.2.1	C	Presence and location of both components of pair satisfying criteria in index_3.2 and index_3.3

Table 8 — CSIs of a capability profile (*continued*)

Conformance point and set number	Conformance point description	Specification reference	Conformance point type (see Table 5)	Abstract test criteria
Index_3.5	Version	ISO 16100-3: 2005, 7.2.2	C	Presence of attributes “Major”, “minor”
Index_3.5.1	major	ISO 16100-3: 2005, 7.2.2	C	The value in ”string” type and ”unqualified” form.
Index_3.5.2	minor	ISO 16100-3: 2005, 7.2.2	C	The value in ”string” type and ”unqualified” form.
Index_3.6	Owner	ISO 16100-3: 2005, 7.2.2	C	Presence of schema components Name, Street, City, Zip, State, Country, Comment It is necessary for MSU capability profiling, but, not necessary for requirement capability profiling.
Index_3.6.1	name	ISO 16100-3: 2005, 7.2.2	C	The value in ”string” type.
Index_3.6.2	street	ISO 16100-3: 2005, 7.2.2	D	The value in ”string” type.
Index_3.6.3	city	ISO 16100-3: 2005, 7.2.2	D	The value in ”string” type.
Index_3.6.4	zip	ISO 16100-3: 2005, 7.2.2	D	The value in ”string” type.
Index_3.6.5	state	ISO 16100-3: 2005, 7.2.2	D	The value in ”string” type.
Index_3.6.6	country	ISO 16100-3: 2005, 7.2.2	D	The value in ”string” type.
Index_3.6.7	comment	ISO 16100-3: 2005, 7.2.2	D	The value in ”string” type.
Index_3.7	ComputingFacilities	ISO 16100-3: 2005, 7.2.2	C	Presence of schema components "Processor0", "OperatingSystem0", "Language", "Memory", "DiskSpace"

Table 8 — CSIs of a capability profile (*continued*)

Conformance point and set number	Conformance point description	Specification reference	Conformance point type (see Table 5)	Abstract test criteria
Index_3.7.1	type	ISO 16100-3: 2005, 7.2.2	C	The value of type in “string” type and “unqualified” form.
Index_3.7.2	Processor type	ISO 16100-3: 2005, 7.2.2	C	The value of Processor type in “string” type and “unqualified” form.
Index_3.7.3	OperatingSystem type	ISO 16100-3: 2005, 7.2.2	C	The value of OperatingSystem type in “string” type and “unqualified” form.
Index_3.7.4	Language name	ISO 16100-3: 2005, 7.2.2	C	The value of Language name in “string” type and “unqualified” form.
Index_3.7.5.	Memory	ISO 16100-3: 2005, 7.2.2	C	Presence of attributes “Memory size”, “Memory unit”
Index_3.7.5.1	Memory size	ISO 16100-3: 2005, 7.2.2	C	The value of Memory size in “string” type and “unqualified” form.
Index_3.7.5.2	Memory unit	ISO 16100-3: 2005, 7.2.2	C	The value of Memory unit in “string” type and “unqualified” form.
Index_3.7.6.	DiskSpace	ISO 16100-3: 2005, 7.2.2	C	Presence of attributes “DiskSpace size”, “DiskSpace unit”
Index_3.7.6.1	DiskSpace size	ISO 16100-3: 2005, 7.2.2	C	The value of DiskSpace size is in “string” type and “unqualified” form.
Index_3.7.6.2	DiskSpace unit	ISO 16100-3: 2005, 7.2.2	C	The value of DiskSpace unit in “string” type and “unqualified” form.
Index_3.8	Additional ComputingFacilities elements	ISO 16100-3: 2005, 7.2	C	Presence and location of schema components satisfying criteria in index_3.7

Table 8 — CSIs of a capability profile (*continued*)

Conformance point and set number	Conformance point description	Specification reference	Conformance point type (see Table 5)	Abstract test criteria
Index_3.9	Performance	ISO 16100-3: 2005, 7.2.2	D	Presence and location of components "ElapsedTime" and "TransactionsPerUnit Time"
Index_3.9.1	ElapsedTime	ISO 16100-3: 2005, 7.2.2	D	The value of ElapsedTime in ”string” type and ”unqualified” form.
Index_3.9.2	TransactionPerUnit Time	ISO 16100-3: 2005, 7.2.2	D	The value of TransactionPerUnitTi me in “string” type and ”unqualified” form.
Index_3.10	Additional Performance elements	ISO 16100-3: 2005, 7.2	D	Presence and location of schema components satisfying criteria in index_3.9
Index _3.11	ReliabilityData	ISO 16100-3: 2005, 7.2.2	D	Presence and location of schema components "UsageHistory", "Shipments", "IntendedSafetyInteg rity", and "Certification"
Index_3.11.1	UsageHistory	ISO 16100-3: 2005, 7.2.2	D	The value of in “string” type and ”unqualified” form.
Index_3.11.2	Shipments number	ISO 16100-3: 2005, 7.2.2	D	The value of in “string” type and ”unqualified” form.
Index_3.11.3	IntendedSafetyInteg rity level	ISO 16100-3: 2005, 7.2.2	D	The value of in “string” type and ”unqualified” form.
Index_3.11.4	Certification no1	ISO 16100-3: 2005, 7.2.2	D	The value in “string” type and ”unqualified” form.

Table 8 — CSIs of a capability profile (*continued*)

Conformance point and set number	Conformance point description	Specification reference	Conformance point type (see Table 5)	Abstract test criteria
Index_3.12	Additional ReliabilityData elements	ISO 16100-3: 2005, 7.2	D	Presence and location of schema components satisfying criteria in index_3.11
Index_3.13	SupportPolicy	ISO 16100-3: 2005, 7.2.2	D	Presence and location of attribute "index"
Index_3.13.1	index	ISO 16100-3: 2005, 7.2.2	D	The value in “string” type and ”unqualified” form.
Index_3.14	Additional SupportPolicy elements	ISO 16100-3: 2005, 7.2	D	Presence and location of schema components satisfying criteria in index_3.13
index _3.15	PriceData	ISO 16100-3: 2005, 7.2.2	D	Presence and location of attributes "invest", "annualSupport", and "unit"
index_3.15.1	invest	ISO 16100-3: 2005, 7.2.2	D	The value in “string” type and ”unqualified” form.
index_3.15.2	annualSupport	ISO 16100-3: 2005, 7.2.2	D	The value in “string” type and ”unqualified” form.
index_3.15.3	unit	ISO 16100-3: 2005, 7.2.2	D	The value in “string” type and ”unqualified” form.
Index_3.16	Additional PriceData elements	ISO 16100-3: 2005, 7.2	D	Presence and location of schema components satisfying criteria in index_3.15

7.4 Conformance of a capability profile matcher

7.4.1 Type 1 matcher

In this part of ISO 16100 the UUT type "capability profile matcher" shall be a Type 1 matcher only. The conformance test of a Type 1 matcher shall involve verifying that the matcher:

a) will accept two vaild profiles and valid templates;

b) can acquire valid profiles that are needed;

c) will provide a report on the matching level described in 6.1.2 of ISO 16100-3:2005.

This conformance test shall not include verifying the behaviour of a Type 1 matcher

7.4.2 CSIs of a capability profile matcher

Table 9 shall be used to generate the ATCs for the UUT type "capability profile matcher".

Table 9 —CSIs for a capability profile matcher

Conformance point and set number	Conformance point description	Specification reference	Conformance point type	Abstract test criteria
Index_1	XML format	ISO 16100-3: 2005, clause 7	A	Capability profile in XML schema format
Index_2	CapabilityProfileConformity	ISO 16100-3: 2005, 5.3	A	Both Required Capability Profile and MSU Capability Profile in conformance with the Capability Template.
Index _3	MatchingResultReport	ISO 16100-3: 2005, 6.1.3 and 7.4	A	Present and location of the schema components “Matching level” and “DetailListReport”
Index_3.1	MatchingLevel	ISO 16100-3: 2005, 6.1.2 and 7.4	A	Matching Level in one of following: — Complete match — All Mandatory match — Some Mandatory match — No Mandatory match
Index_3.2	DetailListReport	ISO 16100-3: 2005, 6.1.2 and 7.4	A	Present of matched functions and the unmatched functions.

Annex A
(informative)

Conformance test for a capability profile

NOTE The following convention is used for the abstract test cases below:

```
if (condition is true)
       perform action
```

A.1 Abstract test cases

The ATCs corresponding to the CSIs of a capability profile can be as follows:

```
if (! Index_1)
    output message “Capability Profile should be in XML format”
if (! Index_2)
    output message “The schema component of the header elements should be in conformance with ISO
                          16100-3:2005, 7.1.2.”
else {
       if (!Index _2.1)
             output message “XML version and target namespaces should be in conformance with ISO 16100-3:2005,
                          7.1.2
       if (! Index_2.2)
             output message “schema component should be in conformance with ISO 16100-3:2005, 7.1.2.”
       else {
              if (!Index_2.2.1)
                     output message “Type should have attribute id”
              If (!Index_2.2.2)
                      Output message “CapabilityProfile should have components ‘PKgtype’, ‘Common’, ‘Specific’
                                 and attribute date.”
              else {
                     if (!Index_2.2.2.1)
                           output message “’PKgtype’ should have version”
                     if (! Index_2.2.2.2)
                           output message “’Common’ should be in ‘CommonPartType’ and ‘Specific’ should be in
                                 ‘SpecificPartType’.”
                   }
             }
        }

 if ( ! Index_3)
       output message “Common part should exist and be in ‘CommonPartType’.”
else {
       if (!Index_3.1)
            output message “The profile should have profile ID but only one ID should be appeared, either
                          Requirement ID or MSU_Capability ID and each ID should have the value in “string”
                          type and “unqualified” form”
       if (! Index_3.2)
             output message “Each profile should have at least one element ‘ReferenceCapabilityClassStructure’.
                          The value of ReferenceCapabilityClassStructure should be Complex Type”
       else {
              If (!Index _3.2.1)
                    output message “The value of id should be in “string” type and “unqualified” form”
              If (!Index _3.2.2)
                    output message “The value of name should be in “string” type and “unqualified” form”
              If (!Index _3.2.3)
                    output message “The value of version should be in “string” type and “unqualified” form”
              If (!Index _3.2.4)
                    output message “The value of uri should be in “string” type and “unqualified” form”
```

```
        }
if (!Index_3.3)
        output message "A Profile should have a TemplateID, the value should be in "string" type and
                       "unqualified"form. It should distinguish the start class within a Capability Class
                       Structure. Typically, the value is NULL if capability profile matching is required for the
                       full capability class structure."
            }

if (Index _3.5) {     //Index_3.5 is type C conformance point
     // "It is necessary to fill the value for MSU capability profiling, but it is not necessary for requirement
        capability profiling. The value of Version is Complex Type"
        If (!Index _3.5.1)
            output message "The value of major should be in "string" type and "unqualified" form"
        If (!Index _3.5.2)
            output message "The value of minor should be in "string" type and "unqualified" form"
        }

if (Index_3.6) {     // Index _3.6 is type C conformance point
      // "it is not necessary for requirement capability profiling, but. It is necessary to fill the value for MSU
      capability profiling. If 'Owner' is empty, it means that the profile is for requirement capability profiling"
        If (!Index _3.6.1)
            output message "The value of name should be in "string" type and "unqualified" form"
        If (!Index _3.6.2)
            output message "The value of street should be in "string" type and "unqualified" form"
        If (!Index _3.6.3)
            output message "The value of city should be in "string" type and "unqualified" form"
        If (!Index _3.6.4)
            output message "The value of zip should be in "string" type and "unqualified" form"
        If (!Index _3.6.5)
            output message "The value of state should be in "string" type and "unqualified" form"
        If (!Index _3.6.6)
            output message "The value of country should be in "string" type and "unqualified" form"
        If (!Index _3.6.7)
            output message "The value of comments should be in "string" type and "unqualified" form"
        }

if (Index _3.7) {     // Index _3.7 is type C conformance point
        If (!Index _3.7.1)
            output message "The value of ComputingFacilities type should be in "string" type and
                           "unqualified" form"
        If (!Index _3.7.2)
            output message "The value of Processor type should be in "string" type and "unqualified" form"
        If (!Index _3.7.3)
            output message "The value of OperatingSystem type should be in "string" type and "unqualified"
                           form"
        If (!Index _3.7.4)
            output message "The value of Language type should be in "string" type and "unqualified" form"
        if (Index_3.7.5) {
            If (!Index _3.7.5.1)
                output message "The value of Memory size should be in "string" type and "unqualified" form"
            If (!Index _3.7.5.2)
                output message "The value of Memory unit should be in "string" type and "unqualified" form"
                    }
        If (Index _3.7.6) {
            If (!Index _3.7.6.1)
                output message "The value of DiskSpace size should be in "string" type and "unqualified"
                               form"
            If (!Index _3.7.6.2)
                output message "The value of DiskSpace unit should be in "string" type and "unqualified"
                               form"
                    }
         }

if (Index _3.9) {     // Index _3.9 is type D conformance point
        If (!Index _3.9.1)
            output message "The value of ElapsedTime should be in "string" type and "unqualified" form"
        If (!Index _3.9.2)
```

```
                output message “The value of TransactionPerUnitTime should be in “string” type and
                                  “unqualified” form”
              }

      if (Index _3.11 ) {    // Index _3.11 is type D conformance point
            If (!Index _3.11.1)
                output message “The value of UsageHistory should be in “string” type and “unqualified” form”
            If (!Index _3.11.2)
                output message “The value of Shipments number should be in “string” type and “unqualified”
            form”
            If (!Index _3.11.3)
                output message “The value of IntendedSafetyIntegrity level should be in “string” type and
                                  “unqualified” form”
            If (!Index _3.11.4)
                output message “The value of Certification no1 should be in “string” type and “unqualified” form”
              }

       if (Index _3.13) {    // Index _3.13 is type D conformance point
            If (!Index _3.13.1)
                output message “The value of SupportPolicy index should be in “string” type and “unqualified”
                                  form”
              }

       if (Index _3.15 exists ) {    // Index _3.15 is type D conformance point
            If (!Index _3.15.1)
                output message “The value of PriceData invest should be in “string” type and “unqualified” form”
            If (!Index _3.15.2)
                output message “The value of PriceData annualSupport should be in “string” type and
                                  “unqualified” form”
            If (!Index _3.15.3)
                output message “The value of PriceData unit should be in “string” type and “unqualified” form”
                      }
         }
```

A template is the schema (refer to 7.1.2 of ISO 16100-3:2005, or as shown below) of the corresponding profile. It is additional information used for the conformance testing of a capability profile.

```
<CapabilityProfiling xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\...\ISO16100-General.xsd">
```

A.2 The test framework

The testing steps and sequence can be as shown in Figure A.1.

A.3 The format of the conformance test report

A recommended format for the conformance test report (see 6.4) is shown below.

a) Validation of XML format

Name of UUT: {name of profile}

XML format validation result: {PASS or FAIL}

The format error occurs as: {NO ERROR or description and exact location of error}

Suggested error correction: {NO CORRECTION NEEDED or description of corrected format}

b) Validation of the Conformance Point:

Name of UUT: {name of profile}

Conformance point validation result: {PASS or FAIL}

The error occurs as: {NO ERROR or description and exact location of error}

Suggested error correction: {NO CORRECTION NEEDED or description of corrected error}

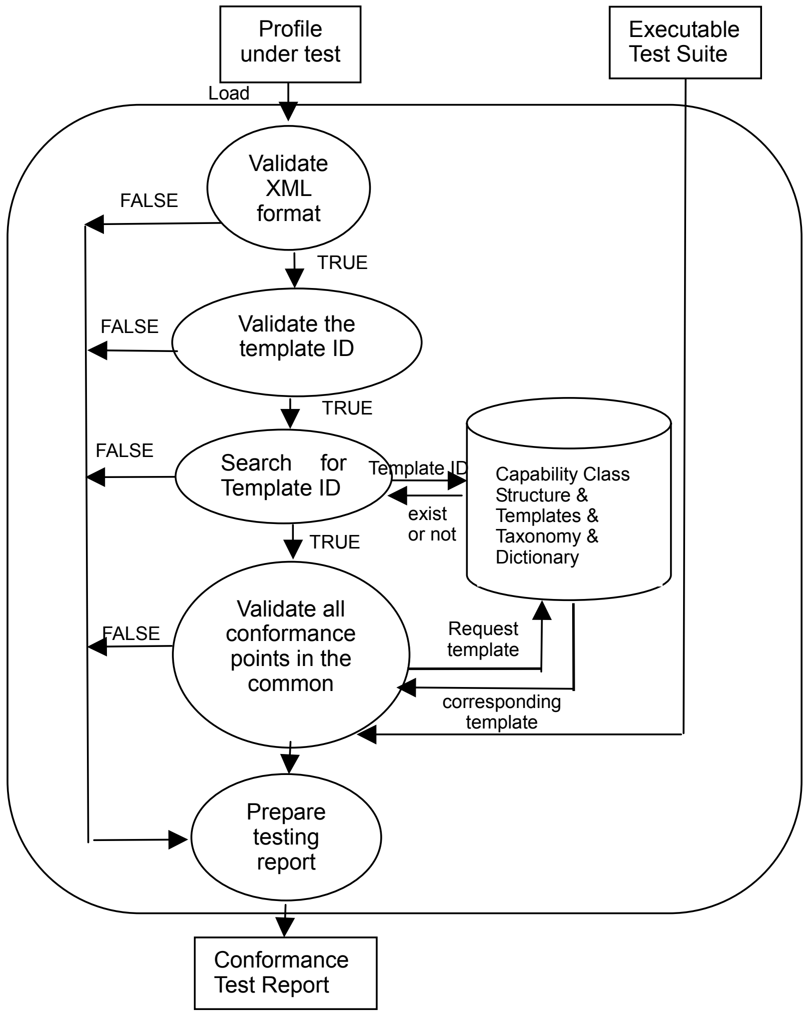

Figure A.1 — Test UUT activity for capability profile

Annex B
(informative)

Conformance test for a type 1 matcher

NOTE The following convention is used for the abstract test cases below:

```
if (condition is true)
      perform action
```

B.1 Abstract test cases

The ATCs corresponding to the CSIs of a type 1 matcher for accepting and acquiring profiles can be as follows:

```
// accept the required profile
  if (! Index_1) // Validate whether the profile is in XML format
        output message "Any Capability profile should be in XML format"
   if (!Index_2)
        output message "Both Required Capability Profile and MSU's Capability Profile should be in conformance
                  with the Capability Profile."

// search for MSU profile according to Reference capability class ID and the Template ID
  if (! Find the profile according to Reference capability class ID and the Template ID)
        output message "Cannot find the msu profile to match"
   if (! msu profile is accessible)
        output message "the msu profile is not accessible"
```

The ATCs corresponding to the CSIs of a type 1 matcher for verifying the syntax of the matching report can be as follows:

```
If (! Index_3)
// validate if the report has 4 items: the MSU profile ID; required profile ID; the matching level and the detail list
      output message " The MatchingLevel and a detail list report should be in the result report."
else {
      If (MatchingLevel == "Complete Match") && (DetailListReport does not include any mandatory match or
optional match)
              output message "the matching level is invalid."
      if (MatchingLevel =="All Mandatory") && (DetailListReport does not include all mandatory match)
              output message "the matching level is invalid."
      if (MatchingLevel ==" Some_Mandatory") && (DetailListReport does not include any mandatory match)
              output message "the matching level is invalid."
      if (MatchingLevel ==" No_Mandatory") && (DetailListReport includes at least one mandatory match )
              output message "the matching level is invalid."
    }
```

The ATCs corresponding to the CSIs of a Type 1 matcher for verify whether the MatchingLevel is in accordance with the detail list can be as follows:

```
counterAllFail = 0,   counterAllPass = 0,
while (Review the detail list of the matching report) // for mandatory point counting
 {
      if (mandatory point XX is no match) {
            output message "Mandatory point XX does not match"
            counterAllFail = countertAllFail + 1
          }
      else
          countertAllPass = counterAllPass + 1
 }
```


if ((counterAllFail == 0) && (MatchingLevel != All_Mandatory match) && (MatchingLevel != Complete match))
output message “The matching level is not in accordance with the detail list”
else if((counterAllPass == 0) && (MatchingLevel == Some_Mandatory match))

output message “The matching level is not in accordance with the detail list”
else if ((counterAllPass = = 0) && (MatchingLevel != No_Mandatory match)
output message “The matching level is not in accordance with the detail list”

B.2 The reference schema for a capability profile matching report

The schema for the matching report can be as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<!--Inline XSD schema-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="MatchingReport">
    <xs:complexType>
      <xs:sequence>
        <xs:element name=" reqirement_ capability_profileID">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ID" type="xs:string" />
            </xs:sequence>
            <xs:attribute name="Validated" type="xs:string" />
//this reqirement_ capability_profile has been validated
            <xs:attribute name="HasMSU" type="xs:string" />
                          //MSU with the required ID has been found
          </xs:complexType>
        </xs:element>
        <xs:element name="capability_msu_profileID" type="xs:string" />
        <xs:element name="the_matching_level">
          <xs:complexType>
            <xs:choice>
              <xs:element name="Complete_Match" />
              <xs:element name="All_Mandatory_Match" />
              <xs:element name="Some_Mandatory_Match" />
              <xs:element name="No_Mandatory_Match" />
            </xs:choice>
          </xs:complexType>
        </xs:element>
        <xs:element name="the_matching_comment" type="xs:string" />
      </xs:sequence>
      <xs:attribute name="date" type="xs:string" />
    </xs:complexType>
  </xs:element>
</xs:schema>
```

B.3 One example of conformance testing flowchart for the matching report

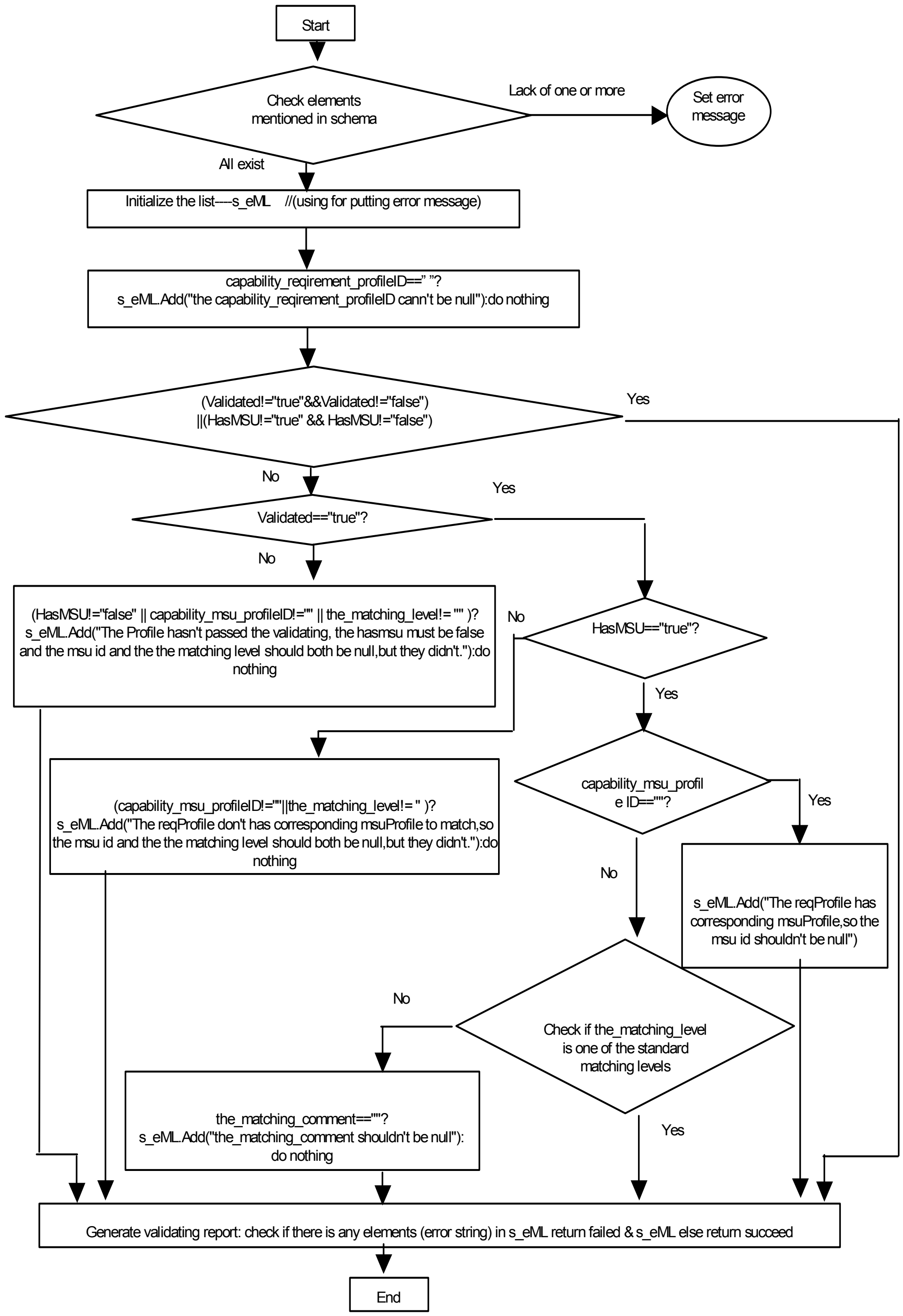

Figure B.1 — Example of conformance testing flowchart for the matching report

B.4 Conformance testing for matching report

The testing steps and sequence can be as shown in Figure B.2.

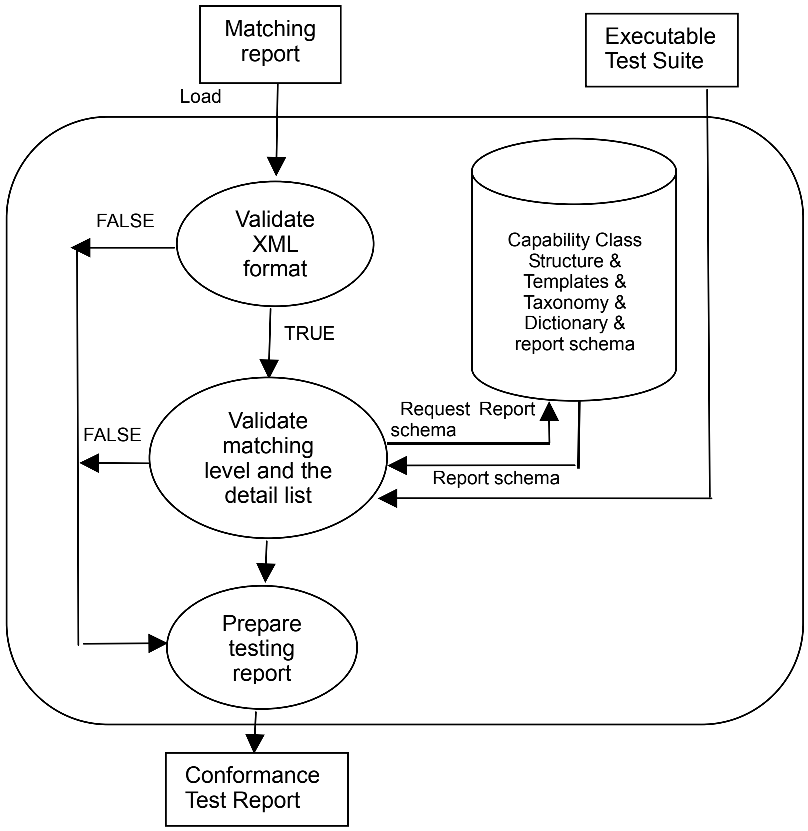

Figure B.2 — "Test UUT" activity for matching report

Bibliography

[1] ISO/IEC Guide 7:1994, *Guidelines for drafting of standards suitable for use for conformity assessment*

[2] ISO/IEC 9646 (all parts), *Information technology — Open Systems Interconnection — Conformance testing methodology and framework*

[3] ISO 10303-31:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 31: Conformance testing methodology and framework: General concepts*

[4] ISO 10303-34, *Industrial automation systems and integration — Product data representation and exchange — Part 34: Conformance testing methodology and framework: Abstract test methods for application protocol implementations*

[5] ISO/IEC 10746-1, *Information technology — Open Distributed Processing — Reference Model: Overview — Part 1*

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

[7] LYNNE ROSENTHAL, MARK SKALL, LISA CARNAHAN, *Conformance Testing and Certification Framework*, NIST White Paper, April 2001

[8] BERND BAUMGARTEN and ALFRED GIESSLER, *OSI Conformance Testing Methodology and TTCN*, Elsevier Science, 1994

ICS 25.040.01

Price based on 32 pages