

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.783

(07/2014)

SERIES X: DATA NETWORKS, OPEN SYSTEM
COMMUNICATIONS AND SECURITY

OSI management – Management functions and ODMA
functions

Guidelines for implementation conformance statement proformas associated with web services-based management systems

Recommendation ITU-T X.783

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	
Services and facilities	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalling and switching	X.50–X.89
Network aspects	X.90–X.149
Maintenance	X.150–X.179
Administrative arrangements	X.180–X.199
OPEN SYSTEMS INTERCONNECTION	
Model and notation	X.200–X.209
Service definitions	X.210–X.219
Connection-mode protocol specifications	X.220–X.229
Connectionless-mode protocol specifications	X.230–X.239
PICS proformas	X.240–X.259
Protocol Identification	X.260–X.269
Security Protocols	X.270–X.279
Layer Managed Objects	X.280–X.289
Conformance testing	X.290–X.299
INTERWORKING BETWEEN NETWORKS	
General	X.300–X.349
Satellite data transmission systems	X.350–X.369
IP-based networks	X.370–X.379
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	
Networking	X.600–X.629
Efficiency	X.630–X.639
Quality of service	X.640–X.649
Naming, Addressing and Registration	X.650–X.679
Abstract Syntax Notation One (ASN.1)	X.680–X.699
OSI MANAGEMENT	
Systems management framework and architecture	X.700–X.709
Management communication service and protocol	X.710–X.719
Structure of management information	X.720–X.729
Management functions and ODMA functions	X.730–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	
Commitment, concurrency and recovery	X.850–X.859
Transaction processing	X.860–X.879
Remote operations	X.880–X.889
Generic applications of ASN.1	X.890–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	X.1000–X.1099
SECURE APPLICATIONS AND SERVICES	X.1100–X.1199
CYBERSPACE SECURITY	X.1200–X.1299
SECURE APPLICATIONS AND SERVICES	X.1300–X.1399
CYBERSECURITY INFORMATION EXCHANGE	X.1500–X.1599
CLOUD COMPUTING SECURITY	X.1600–X.1699

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T X.783

Guidelines for implementation conformance statement proformas associated with web services-based management systems

Summary

Recommendation ITU-T X.783 provides guidelines for implementation conformance statement (ICS) proformas for web services-based interface systems. It provides an overview and constructions for the web service description language (WSDL), and provides several proformas (tables) for each WSDL component to be used in interfaces. Instructions on how to complete the columns in the conformance tables are also provided. Examples of web services interface ICSs are provided in appendices.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T X.783	2014-07-14	2	11.1002/1000/12203

* To access the Recommendation, type the URL <http://handle.itu.int/> in the address field of your web browser, followed by the Recommendation's unique ID. For example, <http://handle.itu.int/11.1002/1000/11830-en>.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2015

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

		Page
1	Scope.....	1
2	References.....	1
3	Definitions	2
	3.1 Terms defined elsewhere	2
	3.2 Terms defined in this Recommendation.....	2
4	Abbreviations and acronyms	2
5	Conventions	2
6	Basis of conformance test methodology for WSDL-based management system interface	3
	6.1 Overview	3
	6.2 Methodology of conformance test for web service-based interfaces	3
	6.3 The relationship between this Recommendation and other Recommendations	4
7	WSDL-based management interface ICS proforma	4
	7.1 WSDL and XML schema overview	5
	7.2 Guidelines for specification of WSDL-based interface ICS proformas	6
8	Instruction for completing the WIICS proformas.....	10
	8.1 Definition of <i>supported</i>	10
	8.2 The "Index" column.....	10
	8.3 The "Subindex" column.....	10
	8.4 The "Support" column.....	10
	8.5 The "Additional information" column.....	10
	8.6 The "Type" column	11
	8.7 The "Message-typing attribute" column and the "Type/element" sub-columns.....	11
	8.8 The "Type identifier" column.....	11
	8.9 The "Type/content model" column.....	11
	8.10 The "Order" column	11
	8.11 The "Constraints and values" column	11
	8.12 The "Nillable" column.....	12
	Appendix I – An example of WSDL-based interface ICS proforma specification	13
	I.1 WSDL definition example.....	13
	I.2 WIICS example	15
	Appendix II – An example showing "types support proforma" specification	18
	II.1 WSDL definition example.....	18
	II.2 WIICS example	18
	Appendix III – An example showing "appendix any type support proforma" specification...	20
	III.1 WSDL definition example.....	20

	Page
III.2 WIICS example	21
Bibliography.....	23

Recommendation ITU-T X.783

Guidelines for implementation conformance statement proformas associated with web services-based management systems

1 Scope

This Recommendation provides guidelines for the web service description language (WSDL)-based interface implementation conformance statement (WIICS) proformas, and the specification of these proformas. The WIICS is a statement made by an implementer to claim conformance to a web services-based interface definition.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- | | |
|---------------|--|
| [ITU-T Q.818] | Recommendation ITU-T Q.818 (2012), <i>Web service-based management services</i> . |
| [ITU-T X.290] | Recommendation ITU-T X.290 (1995), <i>OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – General concepts</i> . |
| [ITU-T X.291] | Recommendation ITU-T X.291 (1995), <i>OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Abstract test suite specification</i> . |
| [ITU-T X.296] | Recommendation ITU-T X.296 (1995), <i>OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Implementation conformance statements</i> . |
| [ITU-T X.724] | Recommendation ITU-T X.724 (1996), <i>Information technology – Open Systems Interconnection – Structure of management information: Requirements and guidelines for implementation conformance statement proformas associated with OSI management</i> . |
| [ITU-T X.782] | Recommendation ITU-T X.782 (2012), <i>Guidelines for defining web services for managed objects and management interfaces</i> . |
| [W3C WSDL] | W3C Recommendation (2001), <i>Web Services Description Language (WSDL) 1.1</i> . |
| [W3C XML] | W3C Recommendation (2000), <i>Extensible Markup Language (XML) 1.0 (Second Edition)</i> . |

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

- 3.1.1 (ICS) item [ITU-T X.290]
- 3.1.2 (ICS) question [ITU-T X.290]
- 3.1.3 status (value) [ITU-T X.290]
- 3.1.4 (support) answer [ITU-T X.290]
- 3.1.5 test case [ITU-T X.290]

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

CORBA	Common Object Request Broker Architecture
HTTP	Hypertext Transport Protocol
ICS	Implementation Conformance Statement
IUT	Implementation Under Test
IXIT	Implementation Extra Information for Testing
RP	Reference Point
SOAP	Simple Object Access Protocol
tML	Telecommunications Markup Language
WIICS	WSDL-based Interface Implementation Conformance Statement
W3C	World Wide Web Consortia
WSDL	Web Service Description Language
WWW	World Wide Web
XML	Extensible Markup Language

5 Conventions

A few conventions are followed in this Recommendation to make the reader aware of the purpose of the text.

Examples of web service description language (WSDL) and extensible markup language (XML) are written in courier new typeface in this Recommendation:

```
<!-- Example XML schema -->
<xsd:complexType name="AType">
  <xsd:sequence>
    <xsd:element name="a" type="xsd:string"/>
    <xsd:element name="b" type="xsd:long"/>
  </xsd:sequence>
</xsd:complexType>
```

6 Basis of conformance test methodology for WSDL-based management system interface

6.1 Overview

Conformance relates an implementation to a standard. It states in which way systems, implemented with respect to a standard, can vary without errors occurring in their cooperation. If an implementation fulfils these requirements, it is conformant to the standard. The check of the statements is the conformance test. The starting point is the definition of conformance requirements in implementation independent interface specifications on the basis of identification of reference points (RPs). A management interface specification should define conformance RPs at which an implementation must be tested to check if it fulfils a set of conformance criteria. During the test, a number of stimuli and events are observed and evaluated at these conformance RPs. Management interface specifications should include conformance statements which identify conformance RPs at every interface of the specified implementations. Because, in general, the information flow between two system components is realized through several RPs, the conformance test has to take the following into consideration:

- a) the test of information flow at each RP;
- b) the test of consistency between the combinations.

Thus, a coordinated test, at all identified RPs is necessary.

6.2 Methodology of conformance test for web service-based interfaces

There are static and dynamic conformance test requirements:

- a) Static conformance test requirements state the functionality that is minimally necessary for conformance testing. The basis of the static conformance test is an implementation conformance statement (ICS) proforma where the functional limitations and possibilities of the standard are defined. It is a document in the form of a questionnaire that has to be completed by the implementer.
- b) Dynamic conformance test requirements specify the potential behaviours of an implementation visible at the identified RPs. The dynamic conformance test contains the realization of test cases in a testing system.

Test cases include concrete test cases and abstract test cases. The concrete test cases are derived from abstract test cases that are given or defined by the standard. A basic precondition for the derivation of the test cases is the definition of test purposes by the standard.

Before starting the test in a test laboratory, the steps below must be followed:

- a) The implementer has to complete the given ICS proforma. Information which makes the ICS proforma statements more precise or concerns the test realization is included in the implementation extra information for testing (IXIT).
- b) The so-called real (concrete) test cases should be derived from the abstract test cases of the standard, or they can be derived from the test purposes directly, where generic aspects (e.g. types) are realized depending on the individual implementation.

Figure 1 illustrates this approach.

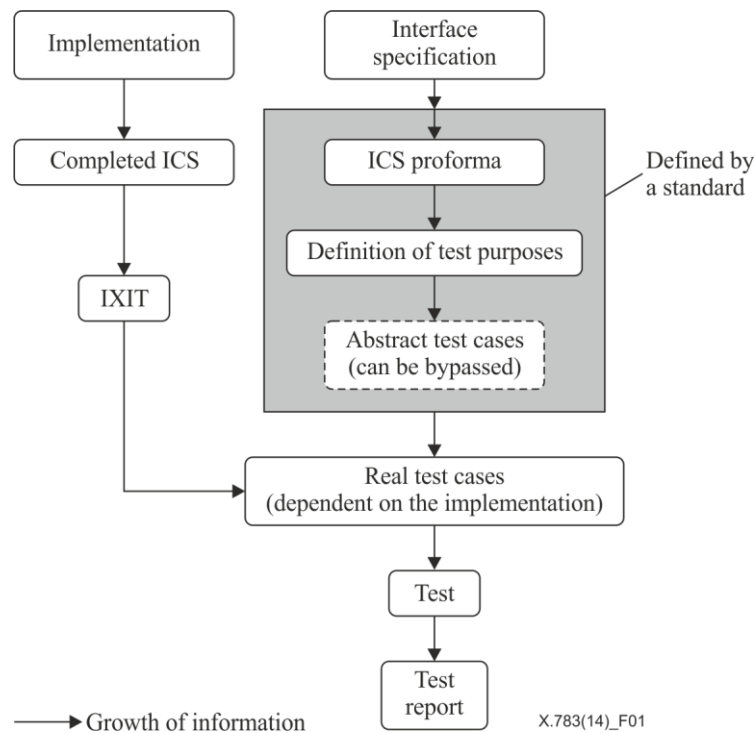


Figure 1 – Testing process of WSDL-based interfaces

6.3 Relationship between this Recommendation and other Recommendations

a) Relationship with [ITU-T X.296]

[ITU-T X.296] provides the common basic for ICS proformas. This Recommendation uses the concepts from [ITU-T X.296].

b) Relationship with [b-ITU-T X.781]

[b-ITU-T X.781] provides the guidelines for ICS of common object request broker architecture (CORBA)-based systems. The conformance testing of this Recommendation use [b-ITU-T X.781] as an example, and use some tables from [b-ITU-T X.781] and make corresponding changes based on web services technology accordingly.

c) Relationship with [b-ITU-T M.3031]

[b-ITU-T M.3031] mainly deals with the telecommunications markup language (tML)-based ICS proforma definition, but there are no ICS proformas to verify interaction, only some to verify conformance of tML-based data types. This Recommendation provides the ICS proformas for both interface interactions as well as data types.

d) Relationship with [ITU-T X.782] and [ITU-T Q.818]

[ITU-T X.782] and [ITU-T Q.818] provide the web services-based management interface framework. This Recommendation provides ICS proformas to support these two Recommendations for their conformance testing aspects.

7 WSDL-based management interface ICS proforma

WSDL is used to define interfaces for WSDL-based systems. The WIICS proforma must express the following WSDL features such as WSDL services, ports, operations, messages and other WSDL data types. This clause first introduces the features of WSDL and XML schema, and then specifies the WIICS proforma according to these language features.

7.1 WSDL and XML schema overview

A web service is a method of communication between two systems over the World Wide Web (WWW). In the World Wide Web Consortia (W3C) definition, a "web service" is "a software system designed to support interoperable machine-to-machine interaction over a network". It contains an interface described in a machine-processable format (i.e., WSDL). Other systems can interact with the web service in a manner prescribed by its description using simple object access protocol (SOAP) messages, typically conveyed using hypertext transport protocol (HTTP) with an XML serialization in conjunction with other web-related standards.

The W3C WSDL is the language used to describe web services. Service definition, described using WSDL, completely defines the service together with the service ports, and fully specifies operations, each operation's messages, each message's parts and used data types. An implementation under test (IUT) may include a service's client function or a service's server function. So, in the interface proforma there should be a space available to specify which side of a function (client or server) the IUT implements.

The W3C WSDL has constructions such as service, binding, operation, message, part, type and so on. According to the WSDL syntax specification (see [W3C WSDL]), the containment relationship between these constructions can be illustrated as in Figure 2.

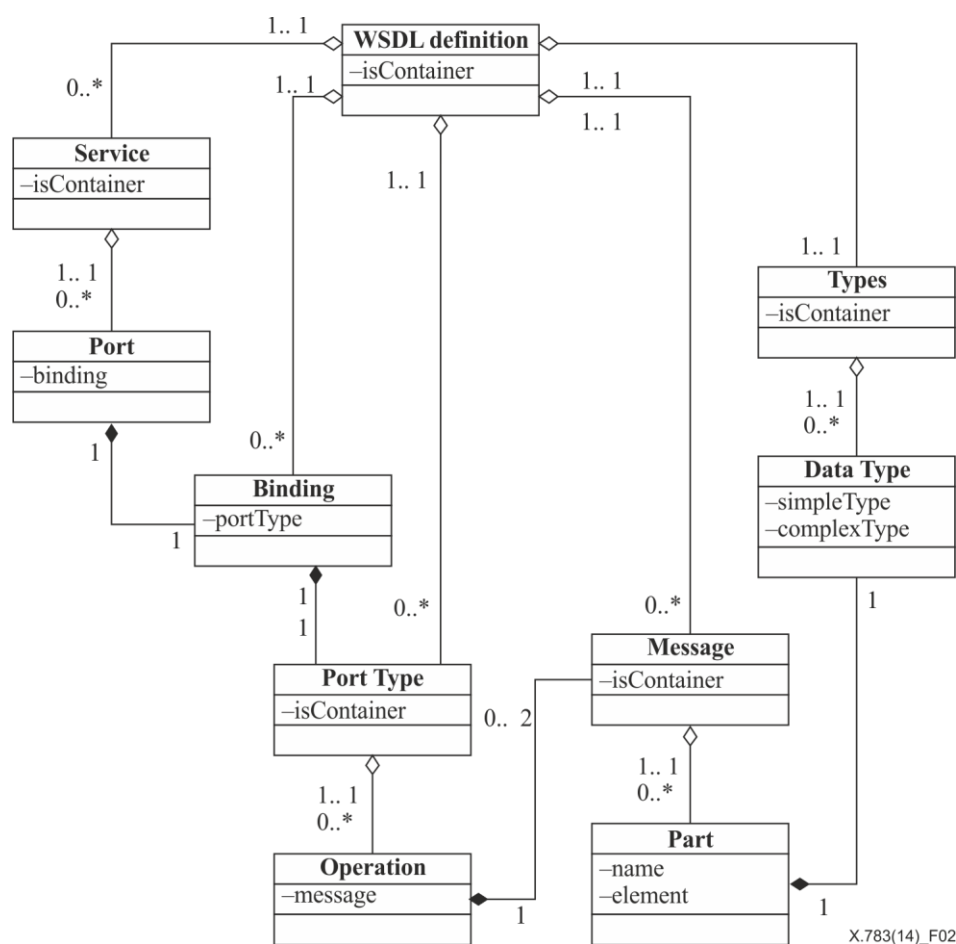


Figure 2 – Containment relationship of WSDL components

The WSDL is based on the W3C XML schema (see [W3C XML]), in other words, a WSDL file itself is, or contains, an XML schema. An XML schema is a description of a type of XML document, typically expressed in terms of constraints on the structure and content of documents of that type, above and beyond the basic syntactical constraints imposed by XML itself. These constraints are generally expressed using some combination of grammatical rules governing the order of elements.

7.2 Guidelines for specification of WSDL-based interface ICS proformas

Proforma specifications shall follow the style as documented in the following clauses. Proforma specifications shall provide the information required by this Recommendation. Additional tables may be included for other information, if needed.

There are three levels of documentation pertaining to WIICS, namely:

- 1) guidelines or recommendation tools for the production of WIICS proformas;
- 2) the WIICS proforma, which is associated with a standard related to WSDL-based network management and is to be completed by a supplier of the implementation;
- 3) the completed WIICS prepared by a supplier of the implementation as part of a conformance claim to a standard related to CORBA-based network management.

7.2.1 General instructions for WSDL ICS proforma specification

This Recommendation provides instructions to construct WSDL-based management system ICS proforma specifications. WIICS proformas include six sub proformas: service support proforma, port support proforma, operation support proforma, message support proforma, port support proforma and types support proforma. All of these proformas, described in the following clauses, are in a tabular form and are similar to the proformas specified in [ITU-T X.724]. Appendices I, II and III provide examples of WIICS proformas specification, which are to be filled in by a supplier of an implementation.

The following common notations, defined in [ITU-T X.291] and [ITU-T X.296], are used for the "Status" value column in this Recommendation:

- m Mandatory
- o Optional
- Not applicable or out of scope

NOTE – The notations "m" or "o" are prefixed by a "c:" when nested under an optional item of the same table.

The following common notations, defined in [ITU-T X.291] and [ITU-T X.296] are used for the "Support" answer column:

- Y Implemented
- N Not implemented
- No answer required
- Ig The item is ignored (i.e., processed syntactically but not semantically)

The WIICS proforma specification is formed by copying clauses 7.2.2, 7.2.3, 7.2.4, 7.2.5, 7.2.6 and 7.2.7, completing the tables except for the "Support" and "Additional information" columns, and extending the remaining tables to meet the requirements of the specification. WIICS proforma shall provide tables for all the operations, messages, parts and types defined in the WSDL information model, regardless of whether they are derived from super-classes or added by redefinition.

To form a WIICS from a WIICS proforma, the supplier of the implementation shall fill in the "Support" and, if appropriate, "Additional information" columns of all the tables in the WIICS proforma.

7.2.2 Service support proforma

The purpose of the proforma for service support is to provide a mechanism for a supplier of an implementation who claims conformance to a service specification to provide conformance information in a standard form.

The service support proforma is given in Table 7-1:

Table 7-1 – Service support table

Index	Service identifier	Status	Support	Additional information

where:

- "Index" field is made up of a consecutive number for readers of the WIICS to refer to each item. Each service stated in a WSDL information model is given a unique number as its index.
- "Service identifier" field is the absolute name of the service.
- For each service, if it is mandatory, the "Status" field should be filled with "m"; if it is optional, the "Status" should be filled with "o".

7.2.3 Service port support proforma

The purpose of the proforma for port support is to provide a mechanism for a supplier of an implementation who claims conformance to a port specification to provide conformance information in a standard form.

The service port support proforma is given in Table 7-2:

Table 7-2 – Service port support table

Index	Sub index	Port identifier	Binding	Style	Transport	Status	Support	Additional information

where:

- "Index" field specifies the index of the parent service to which this port information belongs.
- "Subindex" field is the unique reference of the port within a service.
- "Port identifier" field is to be filled in with the port name.
- "Binding" field indicates the binding attribute of the port.
- "Style" field is the style attribute of the binding.
- "Transport" field is the transport attribute of the binding.
- For each port, if it is mandatory, the "Status" field should be filled with "m"; if it is optional, the "Status" should be filled with "o".
- The "Additional information" field is used to provide a space for the implementer to add more specific information about this port.

7.2.4 Port operation support proforma

The purpose of the proforma for operation support is to provide a mechanism for a supplier of an implementation who claims conformance to an operation specification to provide conformance information in a standard form.

The port operation support proforma is given in Table 7-3:

Table 7-3 – Port operation support table

Index	Subindex	Operation identifier	Status	Support	Additional information

where:

- "Index" field specifies the index of the parent port to which this operation belongs.
- "Subindex" field is made up of a consecutive number for readers of the WIICS to refer to each operation. Each operation stated in a WSDL information model is given a unique number as its index.
- "Operation identifier" field is the absolute name of the operation.
- For each operation, if it is mandatory, the "Status" field should be filled with "m"; if it is optional, the "Status" should be filled with "o".
- The "Additional information" field is used to provide a space for the implementer to add more specific information about this port operation.

7.2.5 Operation message support proforma

The purpose of the proforma for operation message support is to provide a mechanism for a supplier of an implementation who claims conformance to a message specification to provide conformance information in a standard form.

The operation message support proforma is given in Table 7-4:

Table 7-4 – Operation message support table

Index	Subindex	Message identifier	Type	Status	Support	Additional information

where:

- "Index" field specifies the index of the parent operation of this message.
- "Subindex" field is made up of a consecutive number for readers of the WIICS to refer to each message.
- "Message identifier" field is the absolute name of the message.
- "Type" field shows the type of each message. See clause 8.6 for details.
- For each message, if it is mandatory, the "Status" field should be filled with "m"; if it is optional, the "status" should be filled with "o".
- The "Additional information" field is used to provide a space for the implementer to add more specific information about this operation message.

7.2.6 Message part support proforma

The purpose of the proforma for message part support is to provide a mechanism for a supplier of an implementation who claims conformance to a part specification to provide conformance information in a standard form.

The message part support proforma is given in Table 7-5:

Table 7-5 – Message part support table

Index	Subindex	Part identifier	Message-typing attribute		Status	Support	Additional information
			Element	Type			

where:

- "Index" field specifies the index of the parent message of this part.
- "Subindex" field is made up of a consecutive number for readers of the WIICS to refer to each part.
- "Part identifier" field is the absolute name of the part.
- "Message-typing attribute" field shows the type information of the part. The part can be a basic type, a user-defined type or an element reference. See clause 8.7 for details.
- For each part, if it is mandatory, the "Status" field should be filled with "m"; if it is optional, the "Status" should be filled with "o".
- The "Additional information" field is used to provide a space for the implementer to add more specific information about this message part.

7.2.7 Types support proforma

The purpose of the proforma for types support is to provide a mechanism for a supplier of an implementation who claims conformance to types specification to provide conformance information in a standard form.

The types support proforma is given in Table 7-6:

Table 7-6 – Types support table

Index	Subindex	Type identifier	Type/content model	Order	Constraints and values	Nillable	Status	Support	Additional information

where:

- "Index" field specifies the index of the parent part of this type.
- "Subindex" field is made up of a consecutive number for readers of the WIICS to refer to each type.
- "Type identifier" field is the name of the type. See clause 8.8 for details.
- "Type/content model" is used to identify the type or content model of the defined type. If it denotes a simple type, the field is filled in with "simpleType"; if it denotes a complex type, the field is filled in with "complexType" or "group" when it is a reference to a group. See clause 8.9 for details.
- If the row denotes a complex or a group, this "order" field should be filled in with more specific order information of the type, e.g., complexContent, all, sequence or choice. See clause 8.10 for details.
- "Constraints and values" field is used to record constraints and values of the type such as restriction of simple or complex types, occurrence times and default or fixed number of each type. See clause 8.11 for details.
- The "Nillable" field shows whether the type can be nill or not.
- For each type, if it is mandatory, the "Status" field should be filled with "m"; if it is optional, the "Status" field should be filled with "o".
- The "Additional information" field is used to add complementary information.

If the type is a complex type, a simple type or a group, additional type rows are needed to provide support for its nested type fields.

8 Instruction for completing the WIICS proformas

This clause gives the instructions for completing each column defined in clause 7.2.

8.1 Definition of supported

A capability is said to be supported if the IUT is able to realize the specified functionality.

8.2 The "Index" column

The index is located in the leftmost column of each row in the WIICS proforma. The numbering is used to uniquely identify all possible implementation details within the WIICS proforma.

The unique number is constructed with a sequence as follows:

- a) a reference to the super-clause of the item;
- b) the separating character ".";
- c) a unique number.

The "Index" column exhibits the containment relationship between WSDL syntax structures. This containment relationship can be referred to in clause 7.1

In the "types support proforma", the "Index" column of the current row is the corresponding "Subindex" in the "message part support proforma" if it is a direct data type. Otherwise, the "Index" column of the current row is empty if it is a subitem of a data type.

8.3 The "Subindex" column

This column has a similar meaning and the same format as the "Index" column. It is also made up of consecutive numbers.

For each row, only the "Index" column in the "service support proforma" stands for the item of this row. In other proformas, the "Index" column specifies the super-clause item of the item in this row while the "Subindex" column identifies the item in this row.

8.4 The "Support" column

The "Support" column shall be completed, by the supplier or implementer, to indicate the level of support, provided by the implementation, for each item. The available selections for this field are listed in clause 7.2.1.

The following are the guidelines for completing this field:

- a) If an item is claimed as "supported", all the mandatory items it contains must also be supported. Otherwise, the "Support" column can just be filled with 'N'.
- b) If the "Status" column of an item is filled with '-', the only selection for the corresponding "Support" column is '-'.
- c) In the WIICS proforma tables, every item marked with 'm' should be supported by the IUT.

8.5 The "Additional information" column

The "Additional information" column contains additional information provided by suppliers not contained anywhere else. To achieve WIICS for each proforma there may be some important and necessary information that is not or cannot be contained in any column of the row. In such cases, suppliers can complete this field with this information.

8.6 The "Type" column

The "Type" column in the "operation message support proforma" specifies the type of the message. The field should be filled in with one of "input" if it is an input message, "output" if it is an output message or "fault" if it is a fault message.

8.7 The "Message-typing attribute" column and the "Type/element" sub-columns

The "Message-typing attribute" column indicates the type information of the part. If it is an element reference, the sub-field "element" should be completed with the absolute name of the referred element. If it specifies its type, the sub-field "type" should be completed with the type name, whether the type is user-defined or not. In these cases mentioned above, the specific information about both the referred element and the data types can be achieved through the "types support proforma".

8.8 The "Type identifier" column

There are two cases when completing this field:

- a) If the type itself is a nested complex type, simple type or a group in another data type, complete this field with its name and its specific type separated using a ":". Otherwise;
- b) Complete this field with its name.

8.9 The "Type/content model" column

This "Type/content model" column shows the type or content model of a user defined data type. There are two cases when filling this field.

- a) If it is a basic type such as string, int etc., complete this field with "basicType" and the type, separated by a ":", e.g., "basicType:string". Otherwise;
- b) If the data type is a simple type, complete this field with "simpleType"; if it denotes a complex type, complete this field with "complexType" and then complete the "Order" column with specific information; if it denotes a group, complete this field with "group".

8.10 The "Order" column

For each complex type or group, this field should be filled in with additional detailed information. According to the substructure information of the complex type or group, this field should be filled in with different content: complexContent, all, sequence or choice for complex type, annotation, all, choice or sequence for group and "element" for an element.

8.11 The "Constraints and values" column

The "Constraints and values" column in the "types support proforma" contain the constraints and values of the specific type. If applicable, this field may include:

- a) any constraints regarding support of the specific type;
- b) specific values for message parts which are supported;
- c) the allowed types according to the standard specification;
- d) some other value information for this type, such as default and fixed values.

For each user defined data types, the "type kind" is also suggested to be filled in this column. For example, this field should be filled in with "list" or "union" or "enumeration" information.

8.12 The "Nillable" column

The "Nillable" column shows whether the type can be null or not. This field can have the value "True" if the type can be null or "False" otherwise.

Appendix I

An example of WSDL-based interface ICS proforma specification

(This appendix does not form an integral part of this Recommendation.)

I.1 WSDL definition example

In this appendix, the WIICS proforma for "service support", "service port support", "port operation support", "operation message support" and "message part support" will be illustrated. The following is an example of a WSDL definition:

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:y="http://new.webservice.namespace"
targetNamespace="http://new.webservice.namespace">
  <types>
    <schema targetNamespace="http://new.webservice.namespace"
xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <element name="globalResponseType" type="xs:boolean" />
      <simpleType name="WeatherType">
        <restriction base="string">
          <enumeration value="Sunny" />
          <enumeration value="Cloudy" />
          <enumeration value="Rainy" />
          <enumeration value="Snowy" />
        </restriction>
      </simpleType>
      <complexType name="DateType">
        <sequence>
          <element name="d_year" type="int" default="2012"
maxOccurs="1" minOccurs="0" />
          <element name="d_month" type="int" default="1"
maxOccurs="1" minOccurs="1" />
          <element name="d_day" type="int" default="1"
maxOccurs="1" minOccurs="1" />
        </sequence>
      </complexType>
      <complexType name="CityArrayType">
        <complexContent>
          <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType"
xmlns:arrayType="string[]" />
          </restriction>
        </complexContent>
      </complexType>
      <complexType name="WeatherArrayType">
        <complexContent>
          <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType"
xmlns:arrayType="y:WeatherType[]" />
          </restriction>
        </complexContent>
      </complexType>
      <complexType name="CityWeatherType">
        <sequence>
          <element name="cityName" type="string"
nillable="false" />
          <element name="weather" type="y:WeatherType"
nillable="false" />
        </sequence>
      </complexType>
      <group name="CityPairType">
        <sequence>
          <element name="cityOne" type="string" />
          <element name="cityTwo" type="string" />
        </sequence>
      </group>
      <complexType name="BooleanType">
        <sequence>
          <element ref="y:globalResponseType" />
        </sequence>
      </complexType>
    </schema>
  </types>

```

```

        </sequence>
    </complexType>
</schema>
</types>
<message name="getCurrentWeatherRequest">
    <part name="inputCity" element="" type="xs:string"/>
</message>
<message name="getCurrentWeatherResponse">
    <part name="getCurrentWeatherReturn" element="" type="y:WeatherType"/>
</message>
<message name="getPastWeatherRequest">
    <part name="inputDate" element="" type="y:DateType"/>
    <part name="inputCity" element="" type="xs:string"/>
</message>
<message name="getPastWeatherResponse">
    <part name="getPastWeatherReturn" element="" type="y:WeatherType"/>
</message>
<message name="getWeatherOfCitiesRequest">
    <part name="inputCities" element="" type="y:CityArrayType"/>
</message>
<message name="getWeatherOfCitiesResponse">
    <part name="getWeatherReturn" element="" type="y:WeatherArrayType"/>
</message>
<message name="isWeatherOfCityRequest">
    <part name="inputWeatherOfCity" element="" type="y:CityWeatherType"/>
</message>
<message name="isWeatherOfCityResponse">
    <part name="weatherJudgeReturn" element="y:globalResponseType"/>
</message>
<message name="isSameWeatherRequest">
    <part name="inputCityPair" element="" group="y:CityPairType"/>
</message>
<message name="isSameWeatherResponse">
    <part name="judgeReturn" element="" type="y:BooleanType"/>
</message>
<portType name="dailyInformationPort">
    <operation name="getCurrentWeather">
        <input message="y:getCurrentWeatherRequest"/>
        <output message="y:getCurrentWeatherResponse"/>
    </operation>
    <operation name="getPastWeather" parameterOrder="inputDate inputCity
getPastWeatherReturn">
        <input message="y:getPastWeatherRequest"/>
        <output message="y:getPastWeatherResponse"/>
    </operation>
    <operation name="getWeatherOfCities">
        <input message="y:getWeatherOfCitiesRequest"/>
        <output message="y:getWeatherOfCitiesResponse"/>
    </operation>
    <operation name="isWeatherOfCity">
        <input message="y:isWeatherOfCityRequest"/>
        <output message="y:isWeatherOfCityResponse"/>
    </operation>
    <operation name="isSameWeather">
        <input message="y:isSameWeatherRequest"/>
        <output message="y:isSameWeatherResponse"/>
    </operation>
</portType>
<binding name="dailyInformation" type="y:dailyInformationPort">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="getCurrentWeather">
        <soap:operation soapAction="urn:#getCurrentWeather"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
    <operation name="getPastWeather">
        <soap:operation soapAction="urn:#getPastWeather"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
    <operation name="getWeatherOfCities">
        <soap:operation soapAction="urn:#getWeatherOfCities"/>

```

```

        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
    <operation name="isWeatherOfCity">
        <soap:operation soapAction="urn:#isWeatherOfCity"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
    <operation name="isSameWeather">
        <soap:operation soapAction="urn:#isSameWeather"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
</binding>
<service name="establishConnection"/>
<service name="disConnection"/>
<service name="dailyInformation">
    <port name="dailyInformation" binding="y:dailyInformation">
        <soap:address location="No Target Address"/>
    </port>
</service>
</definitions>

```

I.2 WIICS example

According to the WSDL definition above, the WIICS for this WSDL definition can be illustrated as in the following tables.

Table I.1 – Service establishConnection support

Index	Service identifier	Status	Support	Additional information
1	establishConnection	m	Y	

There are no service port, operation, message and part definitions in the establishConnection interface. Support tables for service port, port operation, operation message and message part are not required.

Table I.2 – Service disconnection support

Index	Service identifier	Status	Support	Additional information
2	disConnection	m	Y	

There are no service port, operation, message and part definitions in the disconnection interface. Support tables for service port, port operation, operation message and message part are not required.

Table I.3 – Service dailyInformation support

Index	Service identifier	Status	Support	Additional information
3	dailyInformation	m	Y	

Table I.4 – Service dailyInformation port support

Index	Subindex	Port identifier	Binding	Style	Transport	Status	Support	Additional information
3	3.1	dailyInformation	dailyInformation	null		m	Y	

Table I.5 – Service dailyInformation port operation support

Index	Subindex	Operation identifier	Status	Support	Additional information
3.1	3.1.1	getCurrentWeather	m	Y	
	3.1.2	getPastWeather	m	Y	
	3.1.3	getWeatherOfCities	m	Y	
	3.1.4	isWeatherOfCity	m	Y	
	3.1.5	isSameWeather	m	Y	

Table I.6 – Service dailyInformation operation message support

Index	Subindex	Message identifier	Type	Status	Support	Additional information
3.1.1	3.1.1.1	getCurrentWeatherRequest	input	m	Y	
	3.1.1.2	getCurrentWeatherResponse	output	m	Y	
3.1.2	3.1.2.1	getPastWeatherRequest	input	m	Y	
	3.1.2.2	getPastWeatherResponse	output	m	Y	
3.1.3	3.1.3.1	getWeatherOfCitiesRequest	input	m	Y	
	3.1.3.2	getWeatherOfCitiesResponse	output	m	Y	
3.1.4	3.1.4.1	isWeatherOfCityRequest	input	m	Y	
	3.1.4.2	isWeatherOfCityResponse	output	m	Y	
3.1.5	3.1.5.1	isSameWeatherRequest	input	m	Y	
	3.1.5.2	isSameWeatherResponse	output	m	Y	

Table I.7 – Service dailyInformation message part support

Index	Subindex	Part identifier	Message-typing attribute		Status	Support	Additional information
			Element	Type			
3.1.1.1	3.1.1.1.1	inputCity		string	m	Y	
3.1.1.2	3.1.1.2.1	getCurrentWeatherReturn		WeatherType	m	Y	
3.1.2.1	3.1.2.1.1	inputDate		DateType	m	Y	
	3.1.2.1.2	inputCity		string	m	Y	
3.1.2.2	3.1.2.2.1	getPastWeatherReturn		WeatherType	m	Y	
3.1.3.1	3.1.3.1.1	inputCities		CityArrayType	m	Y	
3.1.3.2	3.1.3.2.1	getWeatherReturn		WeatherArrayType	m	Y	
3.1.4.1	3.1.4.1.1	inputWeatherOfCity		CityWeatherType	m	Y	
3.1.4.2	3.1.4.2.1	weatherJudgeReturn	globalResponseType		m	Y	
3.1.5.1	3.1.5.1.1	inputCityPair		CityPairType	m	Y	
3.1.5.2	3.1.5.2.1	judgeReturn		BooleanType	m	Y	

NOTE – The part "weatherJudgeReturn" is an element reference, so the "Element" column is completed with its referred element name, "globalResponseType", instead of completing the "Type" column.

All type support information of this example will be processed in Appendix II.

Appendix II

An example showing "types support proforma" specification

(This appendix does not form an integral part of this Recommendation.)

II.1 WSDL definition example

This appendix uses the WSDL definition example from Appendix I, which contains several user-defined, complex data types, which could be used to show the format of the "data type support proforma". Please refer to the WSDL in Appendix I.

II.2 WIICS example

According to the referred WSDL definition, the WIICS for this example information model can be illustrated in the following tables:

Table II.1 – Service establishConnection types support

Index	Subindex	Type identifier	Type/content model	Order	Constraints and values	Nilla ble	Status	Support	Additional information
3.1.1.1.1	3.1.1.1.1.1	string	basicType:string				m	Y	
3.1.1.2.1	3.1.1.2.1.1	Weather Type	simpleType		restriction: base=string		m	Y	
	3.1.1.2.1.1.1	string	basicType:string		enumeration: Sunny, Cloudy, Rainy, Snowy		m	Y	
3.1.2.1.1	3.1.2.1.1.1	DateType	complexType	sequence			m	Y	
	3.1.2.1.1.1.1	d_year	baiscType:int		MinOccurs=0 MaxOccurs=1 default=2012		m	Y	
	3.1.2.1.1.1.2	d_month	baiscType:int		MinOccurs=0 MaxOccurs=1 default=1		m	Y	
	3.1.2.1.1.1.3	d_day	baiscType:int		MinOccurs=0 MaxOccurs=1 default=1		m	Y	
3.1.2.1.2	3.1.2.1.2.1	string	basicType:string				m	Y	
3.1.2.2.1	3.1.2.2.1.1	Weather Type	simpleType		restriction: base=string		m	Y	

Table II.1 – Service establishConnection types support

Index	Subindex	Type identifier	Type/content model	Order	Constraints and values	Nilable	Status	Support	Additional information
	3.1.2.2.1.1.1	string	basicType:string		enumeration: Sunny, Cloudy, Rainy, Snowy		m	Y	
3.1.3.1.1	3.1.3.1.1.1	CityArrayType	complexType	complexContent	restriction: base=Array		m	Y	
	3.1.3.1.1.1.1	arrayType:string[]	basicType:string				m	Y	
3.1.3.2.1	3.1.3.2.1.1	WeatherArrayType	complexType	complexContent	restriction: base=Array		m	Y	
	3.1.3.2.1.1.1	arrayType:WeatherType[]	simpleType		restriction: base=string		m	Y	
	3.1.3.2.1.1.1.1	string	basicType:string		enumeration: Sunny, Cloudy, Rainy, Snowy		m	Y	
3.1.4.1.1	3.1.4.1.1.1	CityWeatherType	complexType	sequence			m	Y	
	3.1.4.1.1.1.1	cityName	basicType:string			false	m	Y	
	3.1.4.1.1.1.2	weather:WeatherType	simpleType		restriction: base=string		m	Y	
	3.1.4.1.1.1.2.1	string	basicType:string		enumeration: Sunny, Cloudy, Rainy, Snowy		m	Y	
3.1.5.1.1	3.1.5.1.1.1	globalResponseType	basicType:boolean	element			m	Y	
3.1.5.2.1	3.1.5.2.1.1	BooleanType	complexType	sequence			m	Y	
	3.1.5.2.1.1.1	globalResponseType	basicType:boolean	element			m	Y	

NOTE – Fixed value property should be entered in the "Constraints and values" column as per the default value.

Appendix III

An example showing "appendix any type support proforma" specification

(This appendix does not form an integral part of this Recommendation.)

III.1 WSDL definition example

This appendix illustrates an WSDL example containing a user-defined data type which has an element of type "any" and an operation part using this user-defined data type, which could be used to show the format of "appendix any type support proforma". A service is defined to find a person's location through their name (together with country) or ID number. Below is the WSDL definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:y="http://new.webservice.namespace"
targetNamespace="http://new.webservice.namespace">
  <types>
    <schema targetNamespace="http://new.webservice.namespace"
xmlns="http://www.w3.org/2001/XMLSchema">
      <complexType name="SpecifiedPersonType">
        <sequence>
          <element name="specifiedPerson" type="anyType"/>
          <element name="countryName" type="string"/>
        </sequence>
      </complexType>
      <complexType name="PersonNameType">
        <sequence>
          <element name="surname" type="string"/>
          <element name="firstname" type="string"/>
        </sequence>
      </complexType>
    </schema>
  </types>
  <message name="findPersonRequest">
    <part name="inputPerson" element="" type="y:SpecifiedPersonType"/>
  </message>
  <message name="findPersonResponse">
    <part name="personLocation" element="" type="xs:string"/>
  </message>
  <portType name="findPerson">
    <operation name="findSpecifiedPerson">
      <input message="y:findPersonRequest"/>
      <output message="y:findPersonResponse"/>
    </operation>
  </portType>
  <binding name="findPerson" type="y:findPerson">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="findSpecifiedPerson">
      <soap:operation soapAction="urn:#findSpecifiedObject"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="findPersonService">
    <port name="findPersonPort" binding="y:findPerson">
      <soap:address location="No Target Adress"/>
    </port>
  </service>
```

</definitions>

III.2 WIICS example

According to the WSDL definition above, the WIICS for this example information model can be illustrated in the following tables:

Table III.1 – Service findObjectService support

Index	Service identifier	Status	Support	Additional information
1	findPersonService	m	Y	

Table III.2 – Service findObjectService port support

Index	sub Index	Port identifier	Binding	Style	Transport	Status	Support	Additional information
1	1.1	findPersonPort	findPerson	document		m	Y	

Table III.3 – Service findObjectService port operation support

Index	Subindex	Operation identifier	Status	Support	Additional information
1.1	1.1.1	findSpecifiedPerson	m	Y	

Table III.4 – Service findObjectService operation message support

Index	Subindex	Message identifier	Type	Status	Support	Additional information
1.1.1	1.1.1.1	findPersonRequest	input	m	Y	
	1.1.1.2	findPersonResponse	output	m	Y	

Table III.5 – Service findObjectService message part support

Index	Subindex	Part identifier	Message-typing attribute		Status	Support	Additional information
			Element	Type			
1.1.1.1	1.1.1.1.1	inputPerson		SpecifiedPersonType	m	Y	
1.1.1.2	1.1.1.2.1	personLocation		string	m	Y	

Table III.6 – Service findObjectService types support

Index	Subindex	Type identifier	Type/content model	Order	Constraints and values	Nillable	Status	Support	Additional information
1.1.1.1.1	1.1.1.1.1.1	specifiedPersonType	complexType	sequence			m	Y	
	1.1.1.1.1.1.1	specifiedPerson	anyType				m	Y	Get "any" support from Table III.7
	1.1.1.1.1.1.2	countryName	basicType:string						
1.1.1.2.1	1.1.1.2.1.1	locationReturn	basicType:string				m	Y	

NOTE – The nested type "specifiedPerson" is an any type, thus, the reference information is provided in the “Additional information” column to specify the nested field, which refers to Table III.7.

The following content is the main part of this Appendix III. The table below shows how to deal with any type in WIICS. In this example WSDL, it is assumed that a case where the "inputPerson" part of the "findPersonRequest" message in the "findSpecifiedPerson" operation specifies a person which is to be found. The type "SpecifiedPersonType", which in fact is an any type, can be an int type to identify a person id or a user-defined type, "PersonNameType", to indicate a specific person. Hence, the two types above can be used to replace the any type semantically.

Table III.7 – findPersonService::findPerson::findSpecifiedPerson::findPersonRequest-> inputPerson-> SpecifiedPersonType::specifiedPerson any support

Index	Subindex	Type identifier	Type/content model	Order	Constraints and values	Nillable	Status	Support	Additional information
1.1.1.1.1.1.1	1.1.1.1.1.1.1.1	int	basicType:int				m	Y	
	1.1.1.1.1.1.1.2	PersonNameType	complexType	sequence			m	Y	
	1.1.1.1.1.1.1.2.1	surname	basicType:string				m	Y	
	1.1.1.1.1.1.1.2.2	firstname	basicType:string				m	Y	

NOTE – The table shows the WIICS of any type, only replace it with semantically replaceable types. Complex type, simple type, group should be dealt with in the same way as types support table.

Bibliography

- [b-ITU-T M.3031] Recommendation ITU-T M.3031 (2004), *Guidelines for Implementation Conformance Statement proformas for tML schemas.*
- [b-ITU-T X.781] Recommendation ITU-T X.781 (2001), *Requirements and guidelines for Implementation Conformance Statements proformas associated with CORBA-based systems.*

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems