ETSI ES 201 873-6 V4.9.2 (2018-01)

Methods for Testing and Specification (MTS);

The Testing and Test Control Notation version 3;

Part 6: TTCN‑3 Control Interface (TCI)

**ETSI Standard**

Reference

RES/MTS-201873-6 T3TCI ed491

Keywords

control, interface, methodology, TCI, testing, TTCN-3

***ETSI***

650 Route des Lucioles

F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C

Association à but non lucratif enregistrée à la

Sous-Préfecture de Grasse (06) N° 7803/88

***Important notice***

The present document can be downloaded from:  
<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at <https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:  
<https://portal.etsi.org/People/CommiteeSupportStaff.aspx>

***Copyright Notification***

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.  
The content of the PDF version shall not be modified without the written authorization of ETSI.  
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2017.

All rights reserved.

**DECT**TM, **PLUGTESTS**TM, **UMTS**TM and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.  
**3GPP**TM and **LTE**™ are Trade Marks of ETSI registered for the benefit of its Members and  
of the 3GPP Organizational Partners.  
**oneM2M** logo is protected for the benefit of its Members  
**GSM**® and the GSM logo are Trade Marks registered and owned by the GSM Association.

# 11 W3C XML mapping

## 11.1 Introduction

This clause introduces the TCI XML mapping [10], [11] and [12] for the logging interface of TCI. The XML mapping for the logging interface defines how the IDL definitions for TCI-TL described in clause 7 are mapped to XML. The complete schema definitions for this mapping are given in annex B.

## 11.2 Scopes

The IDL module **tciInterface** is mapped to an XML schema with the name space http://uri.etsi.org/ttcn‑3/tci/TLI\_v4\_10\_1.xsd.

This schema uses further schemas:

* http://uri.etsi.org/ttcn‑3/tci/SimpleTypes\_v4\_10\_1.xsd  
   for the mapping of simple types to XML.
* http://uri.etsi.org/ttcn‑3/tci/Types\_v4\_10\_1.xsd   
   for the mapping of structured types to XML.
* http://uri.etsi.org/ttcn‑3/tci/Values\_v4\_10\_1.xsd  
   for the mapping of values to XML.
* http://uri.etsi.org/ttcn‑3/tci/Templates\_v4\_10\_1.xsd  
   for the mapping of templates to XML.
* http://uri.etsi.org/ttcn‑3/tci/Events\_v4\_10\_1.xsd  
   for the mapping of logging events to XML.

## 11.3 Type mapping

### 11.3.1 Mapping of simple types

#### 11.3.1.1 TBoolean

The IDL **TBoolean** type is mapped to the xsd basic type boolean.

#### 11.3.1.2 TString

The IDL **TString** type is mapped to the xsd basic type string.

#### 11.3.1.3 TInteger

The IDL **TInteger** type is mapped to the xsd basic type integer.

#### 11.3.1.4 TriTimerDurationType

The IDL **TriTimerDurationType** type is mapped to the xsd basic type float.

#### 11.3.1.5 TciParameterPassingModeType

The IDL **TciParameterPassingModeType** type is mapped to the xsd basic type string with enumeration values "in", "out" and "inout".

#### 11.3.1.6 TriStatusType

The IDL **TriStatusType** type is mapped to the xsd basic type string with enumeration values "TRI\_Ok" and "TRI\_Error".

#### 11.3.1.7 TciStatusType

The IDL **TriStatusType** type is mapped to the xsd basic type string with enumeration values "TCI\_Ok" and "TCI\_Error".

#### 11.3.1.8 ComponentStatusType

The IDL **ComponentStatusType** type is mapped to the xsd basic type string with enumeration values "inactiveC", "runningC", "stoppedC", "killedC" and "nullC".

#### 11.3.1.9 TimerStatusType

The IDL **TimerStatusType** type is mapped to the xsd basic type string with enumeration values "runningT", "inactiveT", "expiredT", and "nullT".

#### 11.3.1.10 PortStatusType

The IDL **PortStatusType** type is mapped to the xsd basic type string with enumeration values "startedP", "haltedP" and "stoppedP".

### 11.3.2 Complex type mapping

#### 11.3.2.1 TriPortIdType

**TriPortIdType** is mapped to the following complex type:

<xsd:complexType name="TriPortIdType">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType" />

<xsd:element name="port" type="Types:Port"/>

</xsd:sequence>

</xsd:complexType>

**Elements:**

* comp The TRI component identifier.
* port The identification of the port.

**Attributes:**

* none.

#### 11.3.2.2 TriComponentIdType

**TriComponentIdType** is mapped to the following complex type:

<xsd:complexType name="TriComponentIdType">

<xsd:sequence>

<xsd:choice>

<xsd:element name="null" type="Templates:null"/>

<xsd:element name="id" type="Types:Id"/>

</xsd:choice>

</xsd:sequence>

</xsd:complexType>

**Elements:**

* id The identifier of the TRI component.
* null The null identifier. To be used if there is no TRI component identifier.

**Attributes:**

* none.

#### 11.3.2.3 TriComponentIdListType

**TriComponentIdListType** is mapped to the following complex type:

<xsd:complexType name="TriComponentIdListType">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType" minOccurs="0"

maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

**Elements:**

* comp The identifiers of TRI components in that list.

**Attributes:**

* none.

#### 11.3.2.4 Port

**Port** is mapped to the following complex type:

<xsd:complexType name="Port">

<xsd:sequence>

<xsd:element name="id" type="Types:Id"/>

<xsd:element name="index" type="xsd:int"/>

</xsd:sequence>

</xsd:complexType>

**Elements:**

* id The port identifier.
* port The port index.

**Attributes:**

* none.

#### 11.3.2.5 Id

**Id** is used as identification for components, ports and timers and is mapped to the following complex type:

<xsd:complexType name="Id">

<xsd:sequence>

<xsd:element name="name" type="SimpleTypes:TString"/>

<xsd:element name="id" type="SimpleTypes:TString" minOccurs="0"/>

<xsd:element name="type" type="SimpleTypes:TString" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

**Elements:**

* name The name of the component, port or timer.
* id The internal representation of the component, port or timer.
* type The type of the component, port or timer.

**Attributes:**

* none.

#### 11.3.2.6 TriMessageType

**TriMessageType** is mapped to the following complex type:

<xsd:complexType name="TriMessageType">

<xsd:attribute name="val" type="xsd:hexBinary"/>

<xsd:attribute name="paddingBits" type="xsd:integer" use="optional" default="0"/>

</xsd:complexType>

NOTE: paddingBits is optional with a default value of 0 and should only take values between 0 and 7.   
The relation between paddingBits and numberOfBits is:   
numberOfBits == (length(val-attribute)/2)\*8-paddingBits  
In the byte-aligned case which is the typical one, the paddingBits attribute can be left out.

**Elements:**

* none.

**Attributes:**

* val The encoded message.
* paddingBits The padding bits of the encoded message.

#### 11.3.2.7 TriParameterType

**TriParameterType** is mapped to the following complex type:

<xsd:complexType name="TriParameterType">

<xsd:attribute name="val" type="xsd:hexBinary"/>

<xsd:attribute name="paddingBits" type="xsd:integer" use="optional" default="0"/>

<xsd:attribute name="name" type="SimpleTypes:TString"/>

<xsd:attribute name="mode" type="SimpleTypes:TciParameterPassingModeType"/>

</xsd:complexType>

NOTE: paddingBits is optional with a default value of 0 and should only take values between 0 and 7.   
The relation between paddingBits and numberOfBits is:   
numberOfBits == (length(val-attribute)/2)\*8-paddingBits  
In the byte-aligned case which is the typical one, the paddingBits attribute can be left out.

**Elements:**

* none.

**Attributes:**

* val The encoded parameter.
* paddingBits The padding bits of the encoded parameter.
* name The parameter name.
* mode The parameter passing mode.

#### 11.3.2.8 TriParameterListType

**TriParameterListType** is mapped to the following complex type:

<xsd:complexType name="TriParameterListType">

<xsd:sequence>

<xsd:element name="par" type="Types:TriParameterType" minOccurs="0"

maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

**Sequence of Elements:**

* par The parameters in that list.

**Attributes:**

* none.

#### 11.3.2.9 TriAddressType

**TriAddressType** is mapped to the following complex type:

<xsd:complexType name="TriAddressType">

<xsd:attribute name="val" type="xsd:hexBinary"/>

<xsd:attribute name="paddingBits" type="xsd:integer" use="optional" default="0"/>

</xsd:complexType>

NOTE: paddingBits is optional with a default value of 0 and should only take values between 0 and 7.   
The relation between paddingBits and numberOfBits is:   
numberOfBits == (length(val-attribute)/2)\*8-paddingBits  
In the byte-aligned case which is the typical one, the paddingBits attribute can be left out.

**Elements:**

* none.

**Attributes:**

* val The address value.
* paddingBits The padding bits of the encoded address.

#### 11.3.2.10 TriAddressListType

**TriAddressListType** is mapped to the following complex type:

<xsd:complexType name="TriAddressListType">

<xsd:sequence>

<xsd:element name="addr" type="Types:TriAddressType" minOccurs="0"

maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

**Elements:**

* addr The addresses in that list.

**Attributes:**

* none.

#### 11.3.2.11 TriExceptionType

**TriExceptionType** is mapped to the following complex type:

<xsd:complexType name="TriExceptionType">

<xsd:attribute name="val" type="xsd:hexBinary"/>

<xsd:attribute name="paddingBits" type="xsd:integer" use="optional" default="0"/>

</xsd:complexType>

NOTE: paddingBits is optional with a default value of 0 and should only take values between 0 and 7.   
The relation between paddingBits and numberOfBits is:   
numberOfBits == (length(val-attribute)/2)\*8-paddingBits  
In the byte-aligned case which is the typical one, the paddingBits attribute can be left out.

**Elements:**

* val The exception.

**Attributes:**

* none.

#### 11.3.2.12 TriSignatureIdType

**TriSignatureIdType** is mapped to the following complex type:

<xsd:complexType name="TriSignatureIdType">

<xsd:attribute name="val" type="SimpleTypes:TString" use="required"/>

</xsd:complexType>

**Elements:**

* val The signature.

**Attributes:**

* none.

#### 11.3.2.13 TriTimerIdType

**TriTimerIdType** is mapped to the following complex type:

<xsd:complexType name="TriTimerIdType">

<xsd:sequence>

<xsd:element name="id" type="Types:Id"/>

</xsd:sequence>

</xsd:complexType>

**Elements:**

* id The identification of the timer.

**Attributes:**

* none.

#### 11.3.2.14 TriTimerDurationType

**TriTimerDurationType** is mapped to the following simple type:

<xsd:simpleType name="TriTimerDurationType">

<xsd:restriction base="xsd:float"/>

</xsd:simpleType>

#### 11.3.2.15 QualifiedName

**QualifiedName** is used to fully qualify module parameters, variables, etc and is mapped to the following complex type:

<xsd:complexType name="QualifiedName">

<xsd:attribute name="moduleName" type="SimpleTypes:TString" use="required"/>

<xsd:attribute name="baseName" type="SimpleTypes:TString" use="required"/>

</xsd:complexType>

**Elements:**

* moduleName The module name of the TTCN‑3 module.
* baseName The name of the object that is fully qualified.

**Attributes:**

* none.

#### 11.3.2.16 TciBehaviourIdType

**TciBehaviourIdType** is mapped to the following complex type:

<xsd:complexType name="TciBehaviourIdType">

<xsd:sequence>

<xsd:element name="name" type="Types:QualifiedName"/>

</xsd:sequence>

</xsd:complexType>

**Elements:**

* name The qualified name of the behaviour.

**Attributes:**

* none.

#### 11.3.2.17 TciTestCaseIdType

**TciTestCaseIdType** is mapped to the following complex type:

<xsd:complexType name="TciTestCaseIdType">

<xsd:sequence>

<xsd:element name="name" type="Types:QualifiedName"/>

</xsd:sequence>

</xsd:complexType>

**Elements:**

* name The qualified name of the test case.

**Attributes:**

* none.

#### 11.3.2.18 TciParameterType

**TciParameterType** is mapped to the following complex type:

<xsd:complexType name="TciParameterType">

<xsd:sequence>

<xsd:element name="val" type="Values:Value"/>

</xsd:sequence>

<xsd:attribute name="name" type="SimpleTypes:TString"/>

<xsd:attribute name="mode" type="SimpleTypes:TciParameterPassingModeType"/>

</xsd:complexType>

**Elements:**

* val The encoded parameter.

**Attributes:**

* name The parameter name.
* mode The parameter passing mode.

#### 11.3.2.19 TciParameterListType

**TciParameterListType** is mapped to the following complex type:

<xsd:complexType name="TciParameterListType">

<xsd:sequence>

<xsd:element name="par" type="Types:TciParameterType"

minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

**Sequence of Elements:**

* par The parameters in that list.

**Attributes:**

* none.

#### 11.3.2.20 TriPortIdListType

**TriPortIdListType** is mapped to the following complex type:

<xsd:complexType name="TriPortIdListType">

<xsd:sequence>

<xsd:element name="port" type="Types:TriPortIdType" minOccurs="0"

maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

**Elements:**

* port The identifiers of TRI ports in that list.

**Attributes:**

* none.

### 11.3.3 Abstract value mapping

#### 11.3.3.1 Value

**Value** is mapped to the following complex type:

<xsd:complexType name="Value" mixed="true">

<xsd:group ref="Values:Value"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:group name="Value">

<xsd:choice>

<xsd:element name="integer" type="Values:IntegerValue"/>

<xsd:element name="float" type="Values:FloatValue"/>

<xsd:element name="boolean" type="Values:BooleanValue"/>

<xsd:element name="verdicttype" type="Values:VerdictValue"/>

<xsd:element name="bitstring" type="Values:BitstringValue"/>

<xsd:element name="hexstring" type="Values:HexstringValue"/>

<xsd:element name="octetstring" type="Values:OctetstringValue"/>

<xsd:element name="charstring" type="Values:CharstringValue"/>

<xsd:element name="universal\_charstring" type="Values:UniversalCharstringValue"/>

<xsd:element name="record" type="Values:RecordValue"/>

<xsd:element name="record\_of" type="Values:RecordOfValue"/>

<xsd:element name="set" type="Values:SetValue"/>

<xsd:element name="set\_of" type="Values:SetOfValue"/>

<xsd:element name="enumerated" type="Values:EnumeratedValue"/>

<xsd:element name="union" type="Values:UnionValue"/>

<xsd:element name="anytype" type="Values:AnytypeValue"/>

<xsd:element name="address" type="Values:AddressValue"/>

<xsd:element name="component" type="Values:ComponentValue"/>

<xsd:element name="port" type="Values:PortValue"/>

<xsd:element name="default" type="Values:DefaultValue"/>

<xsd:element name="timer" type="Values:TimerValue"/>

</xsd:choice>

</xsd:group>

<xsd:simpleType name="ValueModifier">

<xs:restriction base="SimpleTypes:TString">  
 <xs:enumeration value="lazy"/>  
 <xs:enumeration value="fuzzy"/>

</xs:restriction>

</xsd:simpleType>

<xsd:attributeGroup name="ValueAtts">

<xsd:attribute name="name" type="SimpleTypes:TString" use="optional"/>

<xsd:attribute name="type" type="SimpleTypes:TString" use="optional"/>

<xsd:attribute name="module" type="SimpleTypes:TString" use="optional"/>

<xsd:attribute name="modifier" type="Values:ValueModifier" use="optional"/>

<xsd:attribute name="annotation" type="SimpleTypes:TString" use="optional"/> </xsd:attributeGroup>

<xsd:group name="BaseValue">

<xsd:choice>

<xsd:sequence>

<xsd:choice>

<xsd:element name="value" type="SimpleTypes:TString"/>

<xsd:element name="matching\_symbol" type="Templates:MatchingSymbol"/>

</xsd:choice>

<xsd:element name="ifpresent" type="SimpleTypes:TEmpty" minOccurs="0"/>

<xsd:element name="length" type="Values:LengthRestriction" minOccurs="0"/>

</xsd:sequence>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type=" SimpleTypes:TEmpty"/>

<xsd:element name="not\_evaluated" type=" SimpleTypes:TEmpty"/>

</xsd:choice>

</xsd:group>

<xsd:complexType name="LengthRestriction">

<xsd:sequence>

<xsd:element name="lower" type="SimpleTypes:TInteger" />

<xsd:element name="upper" type="SimpleTypes:TInteger" minOccurs="0" />

</xsd:sequence>

</xsd:complexType>

**Value Group:**

* integer An integer value.
* float A float value.
* boolean A boolean value.
* verdicttype A verdicttype value.
* bitstring A bitstring value.
* hexstring A hexstring value.
* octetstring An octetstring value.
* charstring A charstring value.
* universal\_charstring A universal charstring value.
* record A record value.
* record\_of A record of value.
* array An array value.
* set A set value.
* set\_of A set of value.
* enumerated An enumerated value.
* union A union value.
* anytype An anytype value.
* address An address value.
* component A component value.
* port A port value.
* default A default value.
* timer A timer value.

**Attributes:**

* name The name of the value, if known.
* type The type of the value, if known.
* module The module of the value, if known.
* modifier The value modifier, if used: either lazy or fuzzy.
* annotation A helper attribute to provide additional matching/mismatching information, etc.

**BaseValue Group:**

* value A value in the string format.
* matching\_symbol A matching symbol when used instead of a value.
* ifpresent The ifpresent matching attribute.
* length The length matching attribute.
* null If no value is given.
* omit If the value is omitted.
* not\_evaluated Used if a @lazy or @fuzzy value contains not evaluated content.

**LengthRestriction Element:**

* lower The lower bound of the length matching attribute.
* upper The upper bound of the length matching attribute. Omitted when equal to infinity.

#### 11.3.3.2 IntegerValue

**IntegerValue** is mapped to the following complex type:

<xsd:complexType name="IntegerValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Items:**

* BaseValue Integer value content described in 11.3.3.1.
* ValueAtts Value attributes described in 11.3.3.1.

#### 11.3.3.3 FloatValue

**FloatValue** is mapped to the following complex type:

<xsd:complexType name="FloatValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Items:**

* BaseValue Float value content described in 11.3.3.1.
* ValueAtts Value attributes described in 11.3.3.1.

#### 11.3.3.4 BooleanValue

**BooleanValue** is mapped to the following complex type:

<xsd:complexType name="BooleanValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Items:**

* BaseValue Boolean value content described in 11.3.3.1.
* ValueAtts Value attributes described in 11.3.3.1.

#### 11.3.3.5 Void

#### 11.3.3.6 VerdictValue

**VerdictValue** is mapped to the following complex type:

<xsd:complexType name="VerdictValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Items:**

* BaseValue Verdict value content described in 11.3.3.1.
* ValueAtts Value attributes described in 11.3.3.1.

#### 11.3.3.7 BitstringValue

**BitstringValue** is mapped to the following complex type:

<xsd:complexType name="BitstringValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Items:**

* BaseValue Bitstring value content described in 11.3.3.1.
* ValueAtts Value attributes described in 11.3.3.1.

#### 11.3.3.8 HexstringValue

**HexstringValue** is mapped to the following complex type:

<xsd:complexType name="HexstringValue">

<xsd:attributeGroup ref="Values:ValueAtts"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Items:**

* BaseValue Hexstring value content described in 11.3.3.1.
* ValueAtts Value attributes described in 11.3.3.1.

#### 11.3.3.9 OctetstringValue

**OctetstringValue** is mapped to the following complex type:

<xsd:complexType name="OctetstringValue">

<xsd:attributeGroup ref="Values:ValueAtts"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Items:**

* BaseValue Octetstring value content described in 11.3.3.1.
* ValueAtts Value attributes described in 11.3.3.1.

#### 11.3.3.10 CharstringValue

**CharstringValue** is mapped to the following complex type:

<xsd:complexType name="CharstringValue">

<xsd:attributeGroup ref="Values:ValueAtts"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Items:**

* BaseValue Charstring value content described in 11.3.3.1.
* ValueAtts Value attributes described in 11.3.3.1.

#### 11.3.3.11 UniversalCharstringValue

**UniversalCharstringValue** is mapped to the following complex type:

<xsd:complexType name="UniversalCharstringValue">

<xsd:attributeGroup ref="Values:ValueAtts"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Items:**

* BaseValue Universal charstring value content described in 11.3.3.1.
* ValueAtts Value attributes described in 11.3.3.1.

#### 11.3.3.12 RecordValue

**RecordValue** is mapped to the following complex type:

<xsd:complexType name="RecordValue">

<xsd:choice>

<xsd:sequence>  
 <xsd:choice>  
 <xsd:group ref="Values:Value" minOccurs="0" maxOccurs="unbounded"/>

    <xsd:element name="matching\_symbol" type="Templates:MatchingSymbol"/>  
 </xsd:choice>  
 <xsd:element name="ifpresent" type="SimpleTypes:TEmpty" minOccurs="0"/>

</xsd:sequence>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type="SimpleTypes:TEmpty"/>

<xsd:element name="not\_evaluated" type="SimpleTypes:TEmpty"/>

</xsd: choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Sequence of Elements:**

* Value The value group is specified in 11.3.3.1. It is used for describing individual elements of the record.
* matching\_symbol A matching symbol if used instead of a value.
* ifpresent The ifpresent matching attribute.
* null If no field is given.
* omit If the field is omitted.
* not\_evaluated Used if a @lazy or @fuzzy value contains not evaluated content.

**Attributes:**

* The same attributes as those of Value.

#### 11.3.3.13 RecordOfValue

**RecordOfValue** is mapped to the following complex type:

<xsd:complexType name="RecordOfValue">

<xsd:group ref="Values:Values"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:group name="Values">

<xsd:choice>

<xsd:sequence>  
 <xsd:choice>

<xsd:element name="integer" type="Values:IntegerValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="float" type="Values:FloatValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="boolean" type="Values:BooleanValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="bitstring" type="Values:BitstringValue"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="hexstring" type="Values:HexstringValue"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="octetstring" type="Values:OctetstringValue"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="charstring" type="Values:CharstringValue"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="universal\_charstring"

type="Values:UniversalCharstringValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="record" type="Values:RecordValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="record\_of" type="Values:RecordOfValue"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="set" type="Values:SetValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="set\_of" type="Values:SetOfValue"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="enumerated" type="Values:EnumeratedValue"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="union" type="Values:UnionValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="anytype" type="Values:AnytypeValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="address" type="Values:AddressValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="component" type="Values:ComponentValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="port" type="Values:PortValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="default" type="Values:DefaultValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="timer" type="Values:TimerValue" minOccurs="0"

maxOccurs="unbounded"/>

    <xsd:element name="matching\_symbol" type="Templates:MatchingSymbol"/>  
 </xsd:choice>  
 <xsd:element name="ifpresent" type="SimpleTypes:TEmpty" minOccurs="0"/>  
 <xsd:element name="length" type="Values:LengthRestriction" minOccurs="0"/>  
 </xsd:sequence>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type="SimpleTypes:TEmpty"/>

<xsd:element name="not\_evaluated" type="SimpleTypes:TEmpty"/>

</xsd:choice>

</xsd:group>

**Values Group:**

* integer An integer value.
* float A float value.
* boolean A boolean value.
* verdicttype A verdicttype value.
* bitstring A bitstring value.
* hexstring A hexstring value.
* octetstring An octetstring value.
* charstring A charstring value.
* universal\_charstring A universal charstring value.
* record A record value.
* record\_of A record of value.
* array An array value.
* set A set value.
* set\_of A set of value.
* enumerated An enumerated value.
* union A union value.
* anytype An anytype value.
* address An address value.
* component A component value.
* default A default value.
* matching\_symbol A matching symbol if used instead of a value.
* ifpresent The ifpresent matching attribute.
* length The length matching attribute.
* null If no field is given.
* omit If the field is omitted.
* not\_evaluated Used if a @lazy or @fuzzy value contains not evaluated content.

**Attributes:**

* The same attributes as those of Value.

#### 11.3.3.14 ArrayValue

**ArrayValue** is mapped to the following complex type:

<xsd:complexType name="ArrayValue">

<xsd:group ref="Values:Values"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Items:**

* Values Array value content described in 11.3.3.13.
* ValueAtts Value attributes described in 11.3.3.1.

#### 11.3.3.15 SetValue

**SetValue** is mapped to the following complex type:

<xsd:complexType name="SetValue">

<xsd:choice>

<xsd:sequence>  
 <xsd:choice>  
 <xsd:group ref="Values:Value" minOccurs="0" maxOccurs="unbounded"/>

    <xsd:element name="matching\_symbol" type="Templates:MatchingSymbol"/>  
 </xsd:choice>  
 <xsd:element name="ifpresent" type="SimpleTypes:TEmpty" minOccurs="0"/>

</xsd:sequence>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type="SimpleTypes:TEmpty"/>

<xsd:element name="not\_evaluated" type="SimpleTypes:TEmpty""/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Sequence of Elements:**

* Value The value group is specified in 11.3.3.1. It is used for describing individual elements of the record.
* matching\_symbol A matching symbol if used instead of a value.
* ifpresent The ifpresent matching attribute.
* null If no field is given.
* omit If the field is omitted.
* not\_evaluated Used if a @lazy or @fuzzy value contains not evaluated content.

**Attributes:**

* The same attributes as those of Value.

#### 11.3.3.16 SetOfValue

**SetOfValue** is mapped to the following complex type:

<xsd:complexType name="SetOfValue">

<xsd:group ref="Values:Values"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Items:**

* Values Set of value content described in 11.3.3.13.
* ValueAtts Value attributes described in 11.3.3.1.

#### 11.3.3.17 EnumeratedValue

**EnumeratedValue** is mapped to the following complex type:

<xsd:complexType name="EnumeratedValue">

<xsd:attributeGroup ref="Values:ValueAtts"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Items:**

* BaseValue Enumerated value content described in 11.3.3.1.
* ValueAtts Value attributes described in 11.3.3.1.

#### 11.3.3.18 UnionValue

**UnionValue** is mapped to the following complex type:

<xsd:complexType name="UnionValue">

<xsd:choice>

<xsd:sequence>  
 <xsd:choice>  
 <xsd:group ref="Values:Value"/>

    <xsd:element name="matching\_symbol" type="Templates:MatchingSymbol"/>  
 </xsd:choice>  
 <xsd:element name="ifpresent" type="SimpleTypes:TEmpty" minOccurs="0"/>

</xsd:sequence>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type="SimpleTypes:TEmpty"/>

<xsd:element name="not\_evaluated" type="SimpleTypes:TEmpty"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Choice of Elements:**

* Value The chosen value. The value group is specified in 11.3.3.1.
* matching\_symbol A matching symbol if used instead of a value.
* ifpresent The ifpresent matching attribute.
* null If no field is given.
* omit If the field is omitted.
* not\_evaluated Used if a @lazy or @fuzzy value contains not evaluated content.

**Attributes:**

* The same attributes as those of Value.

#### 11.3.3.19 AnytypeValue

**AnytypeValue** is mapped to the following complex type:

<xsd:complexType name="AnytypeValue">

<xsd:choice>

<xsd:sequence>  
 <xsd:choice>  
 <xsd:group ref="Values:Value"/>

    <xsd:element name="matching\_symbol" type="Templates:MatchingSymbol"/>  
 </xsd:choice>  
 <xsd:element name="ifpresent" type="SimpleTypes:TEmpty" minOccurs="0"/>

</xsd:sequence>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type="SimpleTypes:TEmpty"/>

<xsd:element name="matching\_symbol" type="Templates:MatchingSymbol"/>

<xsd:element name="not\_evaluated" type="SimpleTypes:TEmpty"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Choice of Elements:**

* Value The chosen value. The value group is specified in 11.3.3.1.
* matching\_symbol A matching symbol if used instead of a value.
* ifpresent The ifpresent matching attribute.
* null If no field is given.
* omit If the field is omitted.
* not\_evaluated Used if a @lazy or @fuzzy value contains not evaluated content.

**Attributes:**

* The same attributes as those of Value.

#### 11.3.3.20 AddressValue

**AddressValue** is mapped to the following complex type:

<xsd:complexType name="AddressValue">

<xsd:group ref="Values:Value"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Items:**

* Value The value group is specified in 11.3.3.1. It is used for describing the content of the address value.ValueAtts Value attributes described in 11.3.3.1.

#### 11.3.3.21 ComponentValue

Value type used for component instances is mapped to the complex type specified below. The content of the XML elements based on the ComponentValue type shall be equal to the string produced by the valueToString operation (described in clause 7.2.2.2.1):

<xsd:complexType name="ComponentValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Items:**

* BaseValue Enumerated value content described in 11.3.3.1.
* ValueAtts Value attributes described in 11.3.3.1.

#### 11.3.3.22 PortValue

Value type used for port instances is mapped to the complex type specified below. The content of the XML elements based on the PortValue type shall be equal to the string produced by the valueToString operation (described in clause 7.2.2.2.1):

<xsd:complexType name="PortValue">

<xsd:choice>

<xsd:element name="value" type="SimpleTypes:TString"/>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type="SimpleTypes:TEmpty"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Choice of Elements:**

* value The universal charstring value as string.
* null If no value is given.
* omit If the value is omitted.

**Attributes:**

* The same attributes as those of Value.

#### 11.3.3.23 DefaultValue

Value type used for default instances is mapped to the complex type specified below. The content of the XML elements based on the DefaultValue type shall be equal to the string produced by the valueToString operation (described in clause 7.2.2.2.1):

<xsd:complexType name="DefaultValue">

<xsd:choice>

<xsd:element name="value" type="SimpleTypes:TString"/>

<xsd:element name="null" type="SimpleTypes:TEmpty"

<xsd:element name="omit" type="SimpleTypes:TEmpty"/>

<xsd:element name="not\_evaluated" type="Values:NotEvaluated"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Choice of Elements:**

* value The universal charstring value as string.
* null If no value is given.
* omit If the value is omitted.
* not\_evaluated Used if a @lazy or @fuzzy value contains not evaluated content.

**Attributes:**

* The same attributes as those of Value.

#### 11.3.3.24 TimerValue

Value type used for timer instances is mapped to the complex type specified below. The content of the XML elements based on the TimerValue type shall be equal to the string produced by the valueToString operation (described in clause 7.2.2.2.1):

<xsd:complexType name="TimerValue">

<xsd:choice>

<xsd:element name="value" type="SimpleTypes:TString"/>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type="SimpleTypes:TEmpty"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

**Choice of Elements:**

* value The universal charstring value as string.
* null If no value is given.
* omit If the value is omitted.

**Attributes:**

* The same attributes as those of Value.

#### 11.3.3.25 MatchingMechanism

**MatchingMechanism** is mapped into a sub-element of a typed value element. The sub-element is based on the complex type specified below:

<xsd:complexType name="MatchingSymbol">

<xsd:choice>

<xsd:element name="any\_value" type="SimpleTypes:TEmpty"/>

<xsd:element name="any\_value\_or\_none" type="SimpleTypes:TEmpty"/>

<xsd:element name="any\_element " type="SimpleTypes:TEmpty"/>

<xsd:element name="any\_element\_or\_none" type="SimpleTypes:TEmpty"/>

<xsd:element name="range" type="Templates:Range"/>

<xsd:element name="list" type="Templates:MatchingList"/>

<xsd:element name="complement" type="Templates:MatchingList"/>

<xsd:element name="subset" type="Templates:MatchingList"/>

<xsd:element name="superset" type="Templates:MatchingList"/>

<xsd:element name="permutation" type="Templates:MatchingList"/>

<xsd:element name="decmatch" type="Templates:DecMatch"/>

</xsd:choice>

</xsd:complexType>

**Choice of Elements:**

* any\_value The *AnyValue* matching symbol.
* any\_value\_or\_none The *AnyValueOrNone* matching symbol.
* any\_element The *AnyElement* matching symbol.
* any\_element\_or\_none The *AnyElementOrNone* matching symbol.
* range A range template.
* list A template list.
* complement A complemented template list.
* subset A subset template.
* superset A superset template.
* permutation A permutation.
* pattern A pattern.
* decmatch A *MatchDecodedContent* matching mechanism.

**Attributes:**

* The same attributes as those of Value.

#### 11.3.3.26 MatchingList

**MatchingList** is mapped to the following complex type. This complex type is also used for mapping of permutations.

<xsd:complexType name="MatchingList">

<xsd:sequence>

<xsd:group ref="Values:Value" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

**Items:**

* Value Individual values present in the list. The Value group is specified in 11.3.3.1.

#### 11.3.3.27 ValueRange

**ValueRange** is mapped to the following complex type:

<xsd:complexType name="Range">

<xsd:sequence>

<xsd:element name="excludeLower" minOccurs="0"/>

<xsd:element name="lower" type="Values:Value" minOccurs="0"/>

<xsd:element name="excludeUpper" minOccurs="0"/>

<xsd:element name="upper" type="Values:Value" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

**Items:**

* excludeLower Present if the lower bound is excluded.
* lower The lower bound of the range. The element is omitted in case of integer range whose lower bound is equal to -infinity. The associated Value type is described in 11.3.3.1.
* excludeUpper Present if the upper bound is excluded.
* upper The upper bound of the range. The element is omitted in case of integer range whose upper bound is equal to infinity. The associated Value type is described in 11.3.3.1.

#### 11.3.3.28 CharacterPattern

**CharacterPattern** is mapped to the following complex type:

<xsd:complexType name="Pattern">

<xsd:sequence>

<xsd:choice>

<xsd:element name="charstring" type="Values:CharstringValue"/>

<xsd:element name="universal\_charstring" type="Values:UniversalCharstringValue"/>

</xsd:choice>

</xsd:sequence>

</xsd:complexType>

**Items:**

* charstring A pattern string in the charstring format.
* universal\_charstring A pattern string in the universal charstring format.

#### 11.3.3.29 MatchDecodedContent

**MatchDecodedContent** is mapped to the following complex type:

<xsd:complexType name="DecMatch">

<xsd:sequence>

<xsd:group ref="Values:Value"/>

</xsd:sequence>

</xsd:complexType>

**Items:**

* Value The content of the *MatchDecodedContent* matching mechanism. The Value group is described in 11.3.3.1.

### 11.3.4 Abstract logging types mapping

#### 11.3.4.1 TciValueTemplate

**TciValueTemplate** is mapped to the following complex type:

<xsd:group name="TypedTemplate">

<xsd:choice>

<xsd:element name="integer" type="Templates:SimpleTemplate"/>

<xsd:element name="float" type="Templates:SimpleTemplate"/>

<xsd:element name="boolean" type="Templates:SimpleTemplate"/>

<xsd:element name="verdicttype" type="Templates:SimpleTemplate"/>

<xsd:element name="bitstring" type="Templates:SimpleTemplate"/>

<xsd:element name="hexstring" type="Templates:SimpleTemplate"/>

<xsd:element name="octetstring" type="Templates:SimpleTemplate"/>

<xsd:element name="charstring" type="Templates:SimpleTemplate"/>

<xsd:element name="universal\_charstring" type="Templates:SimpleTemplate"/>

<xsd:element name="record" type="Templates:RecordTemplate"/>

<xsd:element name="record\_of" type="Templates:RecordOfTemplate"/>

<xsd:element name="array" type="Templates: RecordOfTemplate"/>

<xsd:element name="set" type="Templates:RecordTemplate"/>

<xsd:element name="set\_of" type="Templates:RecordOfTemplate"/>

<xsd:element name="enumerated" type="Templates:SimpleTemplate"/>

<xsd:element name="union" type="Templates:UnionTemplate"/>

<xsd:element name="anytype" type="Templates:AnytypeTemplate"/>

<xsd:element name="address" type="Templates:AddressTemplate"/>

</xsd:choice>

</xsd:group>

<xsd:group name="SpecialTemplate">

<xsd:choice>

<xsd:element name="omit" type="Templates:omit"/>

<xsd:element name="any" type="Templates:any"/>

<xsd:element name="anyoromit" type="Templates:anyoromit"/>

<xsd:element name="templateDef" type="SimpleTypes:TString"/>

</xsd:choice>

</xsd:group>

<xsd:complexType name="TciValueTemplate">

<xsd:choice>

<xsd:group ref="Values:Value"/>

<xsd:group ref="Templates:TypedTemplate"/>

<xsd:group ref="Templates:SpecialTemplate"/>

</xsd:choice>

</xsd:complexType>

**Choice of Elements:**

* Value A structured and typed template definitition.
* TypedTemplate A typed template definitition. It is an obsolete feature kept for backwards compatibility reasons. The referenced group contains a choice of the following elements:
  + integer An integer template.
  + float A float template.
  + boolean A boolean template.
  + verdicttype A verdicttype template.
  + bitstring A bitstring template.
  + hexstring A hexstring template.
  + octetstring An octetstring template.
  + charstring A charstring template.
  + universal\_charstring A universal charstring template.
  + record A record template.
  + record\_of A record of template.
  + array An array template.
  + set A set template.
  + set\_of A set of template.
  + enumerated An enumerated template.
  + union A union template.
  + anytype An anytype template.
  + address An address template.
* SpecialTemplate This group contains special matching elements. It is an obsolete feature kept for backwards compatibility reasons. The referenced group contains a choice of the following elements:
  + omit An omit template.
  + any An any template.
  + anyoromit An anyoromit template.
  + templateDef A complex template definition in text format.

**Attributes:**

* none.

#### 11.3.4.2 TciNonValueTemplate

**TciNonValueTemplate** is mapped to the following complex type:

<xsd:complexType name="TciNonValueTemplate">

<xsd:sequence>

<xsd:choice>

<xsd:element name="any" type="Templates:any"/>

<xsd:element name="all" type="Templates:all"/>

<xsd:element name="templateDef" type="SimpleTypes:TString"/>

<xsd:element name="null" type="Templates:null"/>

<xsd:group ref="Values:Value"/>

</xsd:choice>

</xsd:sequence>

</xsd:complexType>

**Choice of Elements:**

* any An any template.
* all An all template.
* templateDef A complex template definition in text format.
* null No template is given.
* Value A structured and typed template definitition.

**Attributes:**

* none.

#### 11.3.4.3 TciValueList

**TciValueList** is mapped to the following complex type:

<xsd:complexType name="TciValueListType">

<xsd:sequence>

<xsd:element name="val" type="Values:Value"

maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

**Sequence of Elements:**

* val The values in the value list.

**Attributes:**

* none.

#### 11.3.4.4 TciValueDifference

**TciValueDifference** is mapped to the following complex type:

<xsd:complexType name="TciValueDifference">

<xsd:sequence>

<xsd:element name="val" type="SimpleTypes:xpath"/>

<xsd:element name="tmpl"

type="SimpleTypes:xpath"/>

</xsd:sequence>

<xsd:attributeGroup ref="Values:ValueAtts"/>

<xsd:attribute name="desc" type="SimpleTypes:TString"

use="optional"/>

</xsd:complexType>

**Sequence of Elements:**

* val A reference to the mismatching value.
* tmpl A reference to the template.

**Attributes:**

* The same attributes as those of Value.
* desc The reason of the mismatch.

#### 11.3.4.5 TciValueDifferenceList

**TciValueDifferenceList** is mapped to the following complex type:

<xsd:complexType name="TciValueDifferenceList">

<xsd:sequence>

<xsd:element name="diff" type="Templates:TciValueDifference"

maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

**Sequence of Elements:**

* diff The value/template differences in the value difference list.

**Attributes:**

* none.

## 11.4 Mapping of the operations on the logging interface

### 11.4.0 Mapping rules

Every operation provided at the logging interface has a corresponding complex type definition in XML. These complex type definitions are extensions of Event.

### 11.4.1 Event

**Event** is mapped to the following complex type:

<!‑‑ common definition for all events ‑‑>

<xsd:complexType name="Event" mixed="true">

<xsd:sequence>

<xsd:element name="am" type="SimpleTypes:TString"/>

</xsd:sequence>

<xsd:attribute name="ts" type="xsd:long" use="required"/>

<xsd:attribute name="src" type="SimpleTypes:TString" use="optional"/>

<xsd:attribute name="line" type="SimpleTypes:TInteger" use="optional"/>

<!‑‑ general identifier structure for test components, ports and timer ‑‑>

<xsd:attribute name="name" type="SimpleTypes:TString" use="required"/>

<xsd:attribute name="id" type="SimpleTypes:TString " use="required"/>

<xsd:attribute name="type" type="SimpleTypes:TString" use="required"/>

</xsd:complexType>

**Elements:**

* am A message, to be used for further information in the log.

**Attributes:**

* ts The time when the event is produced.
* src The source file of the test specification.
* line The line number where the request is performed.
* name The name of the component which produces this event.
* id The id of the component which produces this event.
* type The type of the component which produces this event.

### 11.4.2 The TCI‑TL interface

#### 11.4.2.1 TCI‑TL provided

The TCI‑TL Provided interface is mapped to the following interface:

<!‑‑ testcases ‑‑>

<xsd:complexType name="tliTcExecute">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="tcId" type="Types:TciTestCaseIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="dur" type="SimpleTypes:TriTimerDurationType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTcStart">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="tcId" type="Types:TciTestCaseIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="dur" type="SimpleTypes:TriTimerDurationType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTcStop">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event"/>

<xsd:sequence>

<xsd:element name="reason" type="SimpleTypes:TString" minOccurs="0"/>

</xsd:sequence>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTcStarted">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="tcId" type="Types:TciTestCaseIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="dur" type="SimpleTypes:TriTimerDurationType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTcTerminated">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="tcId" type="Types:TciTestCaseIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="verdict" type="Values:VerdictValue"/>

<xsd:element name="reason" type="SimpleTypes:TString" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ control ‑‑>

<xsd:complexType name="tliCtrlStart">

<xsd:complexContent>

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCtrlStop">

<xsd:complexContent>

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCtrlTerminated">

<xsd:complexContent>

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ asynchronous communication ‑‑>

<xsd:complexType name="tliMSend\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="msgValue" type="Values:Value"/>

<xsd:element name="addrValue " type="Values:Value" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="msg" type="Types:TriMessageType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMSend\_m\_BC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="msgValue" type="Values:Value"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="msg" type="Types:TriMessageType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMSend\_m\_MC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="msgValue" type="Values:Value"/>

<xsd:element name="addrValues" type="Types:TciValueListType" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="msg" type="Types:TriMessageType" minOccurs="0"/>

<xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMSend\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="msgValue" type="Values:Value"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMSend\_c\_BC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdListType" minOccurs="0"/>

<xsd:element name="msgValue" type="Values:Value"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMSend\_c\_MC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdListType" minOccurs="0"/>

<xsd:element name="msgValue" type="Values:Value"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMDetected\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="msgValue" type="Types:TriMessageType"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMDetected\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="msgValue" type="Values:Value"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMMismatch\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="msgValue" type="Values:Value"/>

<xsd:element name="msgTmpl" type="Templates:TciValueTemplate"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMMismatch\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="msgValue" type="Values:Value"/>

<xsd:element name="msgTmpl" type="Templates:TciValueTemplate"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMReceive\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="msgValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="msgTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMReceive\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="msgValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="msgTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ synchronous communication ‑‑>

<xsd:complexType name="tliPrCall\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCall\_m\_BC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCall\_m\_MC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="addrValues" type="Types:TciValueListType" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCall\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCall\_c\_BC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdListType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCall\_c\_MC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdListType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetCallDetected\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetCallDetected\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetCallMismatch\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetCallMismatch\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetCall\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetCall\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrReply\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType"

minOccurs="0"/>

<xsd:sequence>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

<xsd:element name="transmission‑failure"

type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrReply\_m\_BC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType"

minOccurs="0"/>

<xsd:sequence>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/>

<xsd:element name="transmission‑failure"

type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrReply\_m\_MC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addrValues" type="Types:TciValueListType" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType"

minOccurs="0"/>

<xsd:sequence>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/>

<xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/>

<xsd:element name="transmission‑failure"

type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrReply\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrReply\_c\_BC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdListType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrReply\_c\_MC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdListType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetReplyDetected\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetReplyDetected\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetReplyMismatch\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="replTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetReplyMismatch\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="replTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetReply\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="replTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetReply\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="replTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrRaise\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrRaise\_m\_BC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrRaise\_m\_MC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addrValues" type="Types:TciValueListType" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/>

<xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrRaise\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrRaise\_c\_BC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdListType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrRaise\_c\_MC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdListType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatchDetected\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatchDetected\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatchMismatch\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="excTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatchMismatch\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="excTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatch\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="excValue" type="Values:Value"/>

<xsd:element name="excTmpl" type="Templates:TciValueTemplate"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatch\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="excTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatchTimeoutDetected">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatchTimeout">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ components ‑‑>

<xsd:complexType name="tliCCreate">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType"/>

<xsd:element name="name" type="SimpleTypes:TString"/>

<xsd:element name="hostId" type="Values:Value" minOccurs="0"/>

<xsd:element name="alive" type="SimpleTypes:TBoolean"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCStart">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType"/>

<xsd:element name="name" type="Types:TciBehaviourIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCRunning">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType"/>

<xsd:element name="status" type="SimpleTypes:ComponentStatusType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCAlive">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType"/>

<xsd:element name="status" type="SimpleTypes:ComponentStatusType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCStop">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCKill">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCDoneMismatch">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType"/>

<xsd:element name="compTmpl" type="Templates:TciNonValueTemplate"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCKilledMismatch">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType"/>

<xsd:element name="compTmpl" type="Templates:TciNonValueTemplate"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCDone">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="compTmpl" type="Templates:TciNonValueTemplate"/>

<xsd:element name="verdict" type="Values:VerdictValue" minOccurs="0" />

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCKilled">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="compTmpl" type="Templates:TciNonValueTemplate"/>

<xsd:element name="verdict" type="Values:VerdictValue" minOccurs="0" />

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCTerminated">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="verdict" type="Values:VerdictValue" />

<xsd:element name="reason" type="SimpleTypes:TString" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ ports ‑‑>

<xsd:complexType name="tliPConnect">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:PortConfiguration"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPDisconnect">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:PortConfiguration"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPMap">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:PortConfiguration"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPMapParam">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:tliPMap">

<xsd:sequence>

<xsd:element name="tciPars" type="Types:TciParameterListType" />

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:element name="triPars" type="Types:TriParameterListType" />

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPUnmap">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:PortConfiguration"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPUnmapParam">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:tliPUnmap">

<xsd:sequence>

<xsd:element name="tciPars" type="Types:TciParameterListType" />

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:element name="triPars" type="Types:TriParameterListType"/>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPClear">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:PortStatus"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPStart">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:PortStatus"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPStop">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:PortStatus"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPHalt">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:PortStatus"/>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ codec ‑‑>

<xsd:complexType name="tliEncode">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="val" type="Values:Value"/>

<xsd:choice>

<xsd:element name="msg" type="Types:TriMessageType"/>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

</xsd:choice>

<xsd:element name="codec" type="SimpleTypes:TString"

minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliDecode" mixed="true">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="msg" type="Types:TriMessageType"/>

<xsd:choice>

<xsd:element name="decoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:element name="val" type="Values:Value"/>

</xsd:choice>

<xsd:element name="codec" type="SimpleTypes:TString"

minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ timers ‑‑>

<xsd:complexType name="tliTTimeoutDetected">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="timer" type="Types:TriTimerIdType" />

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTTimeoutMismatch">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="timer" type="Types:TriTimerIdType" />

<xsd:element name="timerTmpl" type="Templates:TciNonValueTemplate" />

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTTimeout">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="timer" type="Types:TriTimerIdType" />

<xsd:element name="timerTmpl" type="Templates:TciNonValueTemplate" />

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTStart">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="timer" type="Types:TriTimerIdType"/>

<xsd:element name="dur" type="SimpleTypes:TriTimerDurationType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTStop">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="timer" type="Types:TriTimerIdType"/>

<xsd:element name="dur" type="SimpleTypes:TriTimerDurationType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTRead">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="timer" type="Types:TriTimerIdType"/>

<xsd:element name="elapsed" type="SimpleTypes:TriTimerDurationType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTRunning">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="timer" type="Types:TriTimerIdType"/>

<xsd:element name="status" type="SimpleTypes:TimerStatusType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ scope ‑‑>

<xsd:complexType name="tliSEnter">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="name" type="Types:QualifiedName" />

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="kind" type="SimpleTypes:TString"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliSLeave">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="name" type="Types:QualifiedName" />

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="returnValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="kind" type="SimpleTypes:TString"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ variables and module parameter ‑‑>

<xsd:complexType name="tliVar">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="name" type="Types:QualifiedName" />

<xsd:element name="val" type="Values:Value" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliModulePar">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="name" type="Types:QualifiedName" />

<xsd:element name="val" type="Values:Value" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ verdicts ‑‑>

<xsd:complexType name="tliGetVerdict">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="verdict" type="Values:VerdictValue"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliSetVerdict">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="verdict" type="Values:VerdictValue"/>

<xsd:element name="reason" type="SimpleTypes:TString" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ log ‑‑>

<xsd:complexType name="tliLog">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="log" type="SimpleTypes:TString"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ alt ‑‑>

<xsd:complexType name="tliAEnter">

<xsd:complexContent>

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliALeave">

<xsd:complexContent>

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliADefaults">

<xsd:complexContent>

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliAActivate">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="name" type="Types:QualifiedName" />

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="ref" type="Values:Value"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliADeactivate">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="ref" type="Values:Value"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliANomatch">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliARepeat">

<xsd:complexContent>

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliAWait">

<xsd:complexContent>

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliAction">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="action" type="SimpleTypes:TString"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMatch">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="expr" type="Values:Value"/>

<xsd:element name="tmpl" type="Templates:TciValueTemplate"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMatchMismatch">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="expr" type="Values:Value"/>

<xsd:element name="tmpl" type="Templates:TciValueTemplate"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliInfo">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="level" type="SimpleTypes:TInteger"/>

<xsd:element name="info" type="SimpleTypes:TString"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMChecked\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="msgValue" type="Types:TriMessageType"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMChecked\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="msgValue" type="Values:Value"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetCallChecked\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetCallChecked\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetReplyChecked\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetReplyChecked\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatchChecked\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatchChecked\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCheckedAny\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCheckedAny\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCheckMismatch\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate"

minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCheckMismatch\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriComponentIdType"

minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate"

minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliRnd">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="val" type="Values:FloatValue"/>

<xsd:element name="seed" type="Values:FloatValue"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliEvaluate">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="name" type="Types:QualifiedName" />

<xsd:element name="evalResult" type="Values:Value" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

Annex B (normative):  
XML Mapping for TCI TL Provided

# B.0 Introduction

This annex defines a mapping for the logging interface of TCI using eXtended Markup Language (XML) schema definitions.

# B.1 TCI‑TL XML Schema for Simple Types

<?xml version="1.0" encoding="UTF‑8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

targetNamespace="http://uri.etsi.org/ttcn‑3/tci/SimpleTypes\_v4\_10\_1.xsd"

xmlns:SimpleTypes="http://uri.etsi.org/ttcn‑3/tci/SimpleTypes\_v4\_10\_1.xsd"

elementFormDefault="qualified">

<!‑‑ Basic definitions ‑‑>

<xsd:simpleType name="xpath">

<!‑‑ this string should be XPATH complient ‑‑>

<xsd:restriction base="xsd:string"/>

</xsd:simpleType>

<xsd:simpleType name="TBoolean">

<xsd:restriction base="xsd:boolean"/>

</xsd:simpleType>

<xsd:simpleType name="TString">

<xsd:restriction base="xsd:string"/>

</xsd:simpleType>

<xsd:simpleType name="TInteger">

<xsd:restriction base="xsd:integer"/>

</xsd:simpleType>

<!‑‑ Miscellaneous ‑‑>

<xsd:simpleType name="TriTimerDurationType">

<xsd:restriction base="xsd:float"/>

</xsd:simpleType>

<xsd:simpleType name="TciParameterPassingModeType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="in"/>

<xsd:enumeration value="inout"/>

<xsd:enumeration value="out"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="TriStatusType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="TRI\_Ok"/>

<xsd:enumeration value="TRI\_Error"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="TciStatusType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="TCI\_Ok"/>

<xsd:enumeration value="TCI\_Error"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="ComponentStatusType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="inactiveC"/>

<xsd:enumeration value="runningC"/>

<xsd:enumeration value="stoppedC"/>

<xsd:enumeration value="killedC"/>

<xsd:enumeration value="nullC"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="TimerStatusType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="runningT"/>

<xsd:enumeration value="inactiveT"/>

<xsd:enumeration value="expiredT"/>

<xsd:enumeration value="nullT"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="PortStatusType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="startedP"/>

<xsd:enumeration value="haltedP"/>

<xsd:enumeration value="stoppedP"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:complexType name="TEmpty" />

</xsd:schema>

# B.2 TCI‑TL XML Schema for Types

<?xml version="1.0" encoding="UTF‑8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

targetNamespace="http://uri.etsi.org/ttcn‑3/tci/Types\_v4\_10\_1.xsd"

xmlns:Types="http://uri.etsi.org/ttcn‑3/tci/Types\_v4\_10\_1.xsd"

xmlns:SimpleTypes="http://uri.etsi.org/ttcn‑3/tci/SimpleTypes\_v4\_10\_1.xsd"

xmlns:Values="http://uri.etsi.org/ttcn‑3/tci/Values\_v4\_10\_1.xsd"

xmlns:Templates="http://uri.etsi.org/ttcn‑3/tci/Templates\_v4\_10\_1.xsd"

elementFormDefault="qualified">

<xsd:import namespace="http://uri.etsi.org/ttcn‑3/tci/Values\_v4\_10\_1.xsd"

schemaLocation="Values\_v4\_10\_1.xsd"/>

<xsd:import namespace="http://uri.etsi.org/ttcn‑3/tci/SimpleTypes\_v4\_10\_1.xsd"

schemaLocation="SimpleTypes\_v4\_10\_1.xsd"/>

<xsd:import namespace="http://uri.etsi.org/ttcn‑3/tci/Templates\_v4\_10\_1.xsd"

schemaLocation="Templates\_v4\_10\_1.xsd"/>

<!‑‑ Connection ‑‑>

<xsd:complexType name="TriPortIdType">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType" />

<xsd:element name="port" type="Types:Port" />

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="TriPortIdListType">

<xsd:sequence>

<xsd:element name="port" type="Types:TriPortIdType" minOccurs="0"

maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="Port">

<xsd:sequence>

<xsd:element name="id" type="Types:Id" />

<xsd:element name="index" type="xsd:int" minOccurs="0" />

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="TriComponentIdType">

<xsd:sequence>

<xsd:choice>

<xsd:element name="null" type="Templates:null"/>

<xsd:element name="id" type="Types:Id" />

</xsd:choice>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="TriComponentIdListType">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType" minOccurs="0"

maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<!‑‑ Communication ‑‑>

<xsd:complexType name="TriMessageType">

<xsd:attribute name="val" type="xsd:hexBinary"/>

<xsd:attribute name="paddingBits" type="xsd:integer" use="optional" default="0"/>

</xsd:complexType>

<xsd:complexType name="TriParameterType">

<xsd:attribute name="val" type="xsd:hexBinary" />

<xsd:attribute name="paddingBits" type="xsd:integer" use="optional" default="0"/>

<xsd:attribute name="name" type="SimpleTypes:TString"/>

<xsd:attribute name="mode" type="SimpleTypes:TciParameterPassingModeType"/>

</xsd:complexType>

<xsd:complexType name="TriParameterListType">

<xsd:sequence>

<xsd:element name="par" type="Types:TriParameterType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="TriExceptionType">

<xsd:attribute name="val" type="xsd:hexBinary"/>

</xsd:complexType>

<xsd:complexType name="TciValueListType">

<xsd:complexContent>

<xsd:extension base="Values:RecordValue"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="TriSignatureIdType">

<xsd:attribute name="val" type="SimpleTypes:TString" use="required"/>

</xsd:complexType>

<xsd:complexType name="TriAddressType">

<xsd:attribute name="val" type="xsd:hexBinary"/>

<xsd:attribute name="paddingBits" type="xsd:integer" use="optional" default="0"/>

</xsd:complexType>

<xsd:complexType name="TriAddressListType">

<xsd:sequence>

<xsd:element name="addr" type="Types:TriAddressType" minOccurs="0"

maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<!‑‑ Miscellaneous ‑‑>

<xsd:complexType name="TriTimerIdType">

<xsd:sequence>

<xsd:element name="id" type="Types:Id" />

</xsd:sequence>

</xsd:complexType>

<!‑‑ Basic definitions ‑‑>

<xsd:complexType name="QualifiedName">

<xsd:attribute name="moduleName" type="SimpleTypes:TString" use="required"/>

<xsd:attribute name="baseName" type="SimpleTypes:TString" use="required"/>

</xsd:complexType>

<!‑‑ general TCI abstract data types ‑‑>

<xsd:complexType name="TciBehaviourIdType">

<xsd:sequence>

<xsd:element name="name" type="Types:QualifiedName" />

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="TciTestCaseIdType">

<xsd:sequence>

<xsd:element name="name" type="Types:QualifiedName" />

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="TciParameterType">

<xsd:sequence>

<xsd:element name="val" type="Values:Value" />

</xsd:sequence>

<xsd:attribute name="name" type="SimpleTypes:TString"/>

<xsd:attribute name="mode" type="SimpleTypes:TciParameterPassingModeType"/>

</xsd:complexType>

<xsd:complexType name="TciParameterListType">

<xsd:sequence>

<xsd:element name="par" type="Types:TciParameterType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<!‑‑ general identifier structure for test components, ports and timer ‑‑>

<xsd:complexType name="Id">

<xsd:sequence>

<xsd:element name="name" type="SimpleTypes:TString" />

<xsd:element name="id" type="SimpleTypes:TString" minOccurs="0"/>

<xsd:element name="type" type="SimpleTypes:TString" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

</xsd:schema>

# B.3 TCI‑TL XML Schema for Values

<?xml version="1.0" encoding="UTF‑8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

targetNamespace="http://uri.etsi.org/ttcn‑3/tci/Values\_v4\_10\_1.xsd"

xmlns:Values="http://uri.etsi.org/ttcn‑3/tci/Values\_v4\_10\_1.xsd"

xmlns:Templates="http://uri.etsi.org/ttcn‑3/tci/Templates\_v4\_10\_1.xsd"

xmlns:SimpleTypes="http://uri.etsi.org/ttcn‑3/tci/SimpleTypes\_v4\_10\_1.xsd" elementFormDefault="qualified">

<xsd:import namespace="http://uri.etsi.org/ttcn‑3/tci/Templates\_v4\_10\_1.xsd"

schemaLocation="Templates\_v4\_10\_1.xsd"/>

<xsd:import namespace="http://uri.etsi.org/ttcn‑3/tci/SimpleTypes\_v4\_10\_1.xsd"

schemaLocation="SimpleTypes\_v4\_10\_1.xsd"/>

<xsd:simpleType name="ValueModifier">

<xsd:restriction base="SimpleTypes:TString">  
      <xsd:enumeration value="lazy"/>  
      <xsd:enumeration value="fuzzy"/>

    </xsd:restriction>

</xsd:simpleType>

<xsd:attributeGroup name="ValueAtts">

<xsd:attribute name="name" type="SimpleTypes:TString" use="optional"/>

<xsd:attribute name="type" type="SimpleTypes:TString" use="optional"/>

<xsd:attribute name="module" type="SimpleTypes:TString" use="optional"/>

<xsd:attribute name="modifier" type="Values:ValueModifier" use="optional"/>

<xsd:attribute name="annotation" type="SimpleTypes:TString" use="optional"/> </xsd:attributeGroup>

<xsd:group name="BaseValue">

<xsd:choice>

<xsd:sequence>  
 <xsd:choice>  
     <xsd:element name="value" type="SimpleTypes:TString"/>  
     <xsd:element name="matching\_symbol" type="Templates:MatchingSymbol"/>  
 </xsd:choice>  
 <xsd:element name="ifpresent" type="SimpleTypes:TEmpty" minOccurs="0"/>  
 <xsd:element name="length" type="Values:LengthRestriction" minOccurs="0"/>  
 </xsd:sequence>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type=" SimpleTypes:TEmpty"/>  
 <xsd:element name="not\_evaluated" type=" SimpleTypes:TEmpty"/>

</xsd:choice>

</xsd:group>

<xsd:group name="Value">

<xsd:choice>

<xsd:element name="integer" type="Values:IntegerValue"/>

<xsd:element name="float" type="Values:FloatValue"/>

<xsd:element name="boolean" type="Values:BooleanValue"/>

<xsd:element name="verdicttype" type="Values:VerdictValue"/>

<xsd:element name="bitstring" type="Values:BitstringValue"/>

<xsd:element name="hexstring" type="Values:HexstringValue"/>

<xsd:element name="octetstring" type="Values:OctetstringValue"/>

<xsd:element name="charstring" type="Values:CharstringValue"/>

<xsd:element name="universal\_charstring" type="Values:UniversalCharstringValue"/>

<xsd:element name="record" type="Values:RecordValue"/>

<xsd:element name="record\_of" type="Values:RecordOfValue"/>

<xsd:element name="set" type="Values:SetValue"/>

<xsd:element name="set\_of" type="Values:SetOfValue"/>

<xsd:element name="enumerated" type="Values:EnumeratedValue"/>

<xsd:element name="union" type="Values:UnionValue"/>

<xsd:element name="anytype" type="Values:AnytypeValue"/>

<xsd:element name="address" type="Values:AddressValue"/>

<xsd:element name="component" type="Values:ComponentValue"/>

<xsd:element name="port" type="Values:PortValue"/>

<xsd:element name="default" type="Values:DefaultValue"/>

<xsd:element name="timer" type="Values:TimerValue"/>

</xsd:choice>

</xsd:group>

<xsd:group name="Values">

<xsd:choice>

<xsd:sequence>  
 <xsd:choice>

<xsd:element name="integer" type="Values:IntegerValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="float" type="Values:FloatValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="boolean" type="Values:BooleanValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="bitstring" type="Values:BitstringValue"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="hexstring" type="Values:HexstringValue"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="octetstring" type="Values:OctetstringValue"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="charstring" type="Values:CharstringValue"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="universal\_charstring"

type="Values:UniversalCharstringValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="record" type="Values:RecordValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="record\_of" type="Values:RecordOfValue"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="set" type="Values:SetValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="set\_of" type="Values:SetOfValue"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="enumerated" type="Values:EnumeratedValue"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="union" type="Values:UnionValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="anytype" type="Values:AnytypeValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="address" type="Values:AddressValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="component" type="Values:ComponentValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="port" type="Values:PortValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="default" type="Values:DefaultValue" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="timer" type="Values:TimerValue" minOccurs="0"

maxOccurs="unbounded"/>

    <xsd:element name="matching\_symbol" type="Templates:MatchingSymbol"/>  
 </xsd:choice>  
 <xsd:element name="ifpresent" type="SimpleTypes:TEmpty" minOccurs="0"/>  
 <xsd:element name="length" type="Values:LengthRestriction" minOccurs="0"/>  
 </xsd:sequence>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type="SimpleTypes:TEmpty"/>

<xsd:element name="not\_evaluated" type="SimpleTypes:TEmpty"/>

</xsd:choice>

</xsd:group>

<xsd:complexType name="Value" mixed="true">

<xsd:group ref="Values:Value"/> <xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="LengthRestriction">  
 <xsd:sequence>  
 <xsd:element name="lower" type="SimpleTypes:TInteger" />  
 <xsd:element name="upper" type="SimpleTypes:TInteger" minOccurs="0" />  
 </xsd:sequence>  
 </xsd:complexType>

<xsd:complexType name="Value" mixed="true">

<xsd:group ref="Values:Value"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<!‑‑ general event elements ‑‑>

<xsd:complexType name="IntegerValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="FloatValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="BooleanValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="VerdictValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="BitstringValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="HexstringValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="OctetstringValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="CharstringValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="UniversalCharstringValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="RecordValue">

<xsd:choice>

<xsd:sequence>  
 <xsd:choice>  
 <xsd:group ref="Values:Value" minOccurs="0" maxOccurs="unbounded"/>

    <xsd:element name="matching\_symbol" type="Templates:MatchingSymbol"/>  
 </xsd:choice>  
 <xsd:element name="ifpresent" type="SimpleTypes:TEmpty" minOccurs="0"/>

</xsd:sequence>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type="SimpleTypes:TEmpty"/>

<xsd:element name="not\_evaluated" type="SimpleTypes:TEmpty"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="RecordOfValue">

<xsd:group ref="Values:Values"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="ArrayValue">

<xsd:group ref="Values:Values"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="SetValue">

<xsd:choice>

<xsd:sequence>  
 <xsd:choice>  
 <xsd:group ref="Values:Value" minOccurs="0" maxOccurs="unbounded"/>

    <xsd:element name="matching\_symbol" type="Templates:MatchingSymbol"/>  
 </xsd:choice>  
 <xsd:element name="ifpresent" type="SimpleTypes:TEmpty" minOccurs="0"/>

</xsd:sequence>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type="SimpleTypes:TEmpty"/>

<xsd:element name="not\_evaluated" type="SimpleTypes:TEmpty"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="SetOfValue">

<xsd:group ref="Values:Values"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="EnumeratedValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="UnionValue">

<xsd:choice>

<xsd:sequence>  
 <xsd:choice>  
 <xsd:group ref="Values:Value"/>

    <xsd:element name="matching\_symbol" type="Templates:MatchingSymbol"/>  
 </xsd:choice>  
 <xsd:element name="ifpresent" type="SimpleTypes:TEmpty" minOccurs="0"/>

</xsd:sequence>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type="SimpleTypes:TEmpty"/>

<xsd:element name="not\_evaluated" type="SimpleTypes:TEmpty"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="AnytypeValue">

<xsd:choice>

<xsd:sequence>  
 <xsd:choice>  
 <xsd:group ref="Values:Value"/>

    <xsd:element name="matching\_symbol" type="Templates:MatchingSymbol"/>  
 </xsd:choice>  
 <xsd:element name="ifpresent" type="SimpleTypes:TEmpty" minOccurs="0"/>

</xsd:sequence>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type="SimpleTypes:TEmpty"/>

<xsd:element name="not\_evaluated" type="SimpleTypes:TEmpty"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="AddressValue">

<xsd:group ref="Values:Value"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="ComponentValue">

<xsd:group ref="Values:BaseValue"/>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="PortValue">

<xsd:choice>

<xsd:element name="value" type="SimpleTypes:TString"/>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type="SimpleTypes:TEmpty"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="DefaultValue">

<xsd:choice>

<xsd:element name="value" type="SimpleTypes:TString"/>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type="SimpleTypes:TEmpty"/>

<xsd:element name="not\_evaluated" type="SimpleTypes:TEmpty"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="TimerValue">

<xsd:choice>

<xsd:element name="value" type="SimpleTypes:TString"/>

<xsd:element name="null" type="SimpleTypes:TEmpty"/>

<xsd:element name="omit" type="SimpleTypes:TEmpty"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

</xsd:schema>

# B.4 TCI‑TL XML Schema for Templates

<?xml version="1.0" encoding="UTF‑8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

targetNamespace="http://uri.etsi.org/ttcn‑3/tci/Templates\_v4\_10\_1.xsd"

xmlns:Templates="http://uri.etsi.org/ttcn‑3/tci/Templates\_v4\_10\_1.xsd"

xmlns:Values="http://uri.etsi.org/ttcn‑3/tci/Values\_v4\_10\_1.xsd"

xmlns:SimpleTypes="http://uri.etsi.org/ttcn‑3/tci/SimpleTypes\_v4\_10\_1.xsd" elementFormDefault="qualified">

<xsd:import namespace="http://uri.etsi.org/ttcn‑3/tci/Values\_v4\_10\_1.xsd"

schemaLocation="Values\_v4\_10\_1.xsd"/>

<xsd:import namespace="http://uri.etsi.org/ttcn‑3/tci/SimpleTypes\_v4\_10\_1.xsd"

schemaLocation="SimpleTypes\_v4\_10\_1.xsd"/>

<xsd:group name="TypedTemplate">

<xsd:choice>

<xsd:element name="integer" type="Templates:SimpleTemplate"/>

<xsd:element name="float" type="Templates:SimpleTemplate"/>

<xsd:element name="boolean" type="Templates:SimpleTemplate"/>

<xsd:element name="verdicttype" type="Templates:SimpleTemplate"/>

<xsd:element name="bitstring" type="Templates:SimpleTemplate"/>

<xsd:element name="hexstring" type="Templates:SimpleTemplate"/>

<xsd:element name="octetstring" type="Templates:SimpleTemplate"/>

<xsd:element name="charstring" type="Templates:SimpleTemplate"/>

<xsd:element name="universal\_charstring" type="Templates:SimpleTemplate"/>

<xsd:element name="record" type="Templates:RecordTemplate"/>

<xsd:element name="record\_of" type="Templates:RecordOfTemplate"/>

<xsd:element name="array" type="Templates: RecordOfTemplate"/>

<xsd:element name="set" type="Templates:RecordTemplate"/>

<xsd:element name="set\_of" type="Templates:RecordOfTemplate"/>

<xsd:element name="enumerated" type="Templates:SimpleTemplate"/>

<xsd:element name="union" type="Templates:UnionTemplate"/>

<xsd:element name="anytype" type="Templates:AnytypeTemplate"/>

<xsd:element name="address" type="Templates:AddressTemplate"/>

</xsd:choice>

</xsd:group>

<xsd:group name="SpecialTemplate">

<xsd:choice>

<xsd:element name="omit" type="Templates:omit"/>

<xsd:element name="any" type="Templates:any"/>

<xsd:element name="anyoromit" type="Templates:anyoromit"/>

<xsd:element name="templateDef" type="SimpleTypes:TString"/>

</xsd:choice>

</xsd:group>

<xsd:complexType name="TciValueTemplate">

<xsd:choice>

<xsd:group ref="Values:Value"/>

<xsd:group ref="Templates:TypedTemplate"/>

<xsd:group ref="Templates:SpecialTemplate"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="omit">

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="any">

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="anyoromit">

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="MatchingSymbol">

<xsd:choice>

<xsd:element name="any\_value" type="SimpleTypes:TEmpty"/>

<xsd:element name="any\_value\_or\_none" type="SimpleTypes:TEmpty"/>

<xsd:element name="any\_element " type="SimpleTypes:TEmpty"/>

<xsd:element name="any\_element\_or\_none" type="SimpleTypes:TEmpty"/>

<xsd:element name="range" type="Templates:Range"/>

<xsd:element name="list" type="Templates:MatchingList"/>

<xsd:element name="complement" type="Templates:MatchingList"/>

<xsd:element name="subset" type="Templates:MatchingList"/>

<xsd:element name="superset" type="Templates:MatchingList"/>

<xsd:element name="permutation" type="Templates:MatchingList"/>

<xsd:element name="pattern" type="Templates:Pattern"/>

<xsd:element name="decmatch" type="Templates:DecMatch"/>

</xsd:choice>

</xsd:complexType>

<xsd:complexType name="Range">

<xsd:sequence>

<xsd:element name="excludeLower" minOccurs="0"/>

<xsd:element name="lower" type="Values:Value" minOccurs="0"/>

<xsd:element name="excludeUpper" minOccurs="0"/>

<xsd:element name="upper" type="Values:Value" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="MatchingList">

<xsd:sequence>

<xsd:group ref="Values:Value" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="Pattern">

<xsd:sequence>

<xsd:choice>

<xsd:element name="charstring" type="Values:CharstringValue"/>

<xsd:element name="universal\_charstring" type="Values:UniversalCharstringValue"/>

</xsd:choice>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="DecMatch">

<xsd:sequence>

<xsd:group ref="Values:Value"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="TciNonValueTemplate">

<xsd:sequence>

<xsd:choice>

<xsd:element name="any" type="Templates:any"/>

<xsd:element name="all" type="Templates:all"/>

<xsd:element name="templateDef" type="SimpleTypes:TString"/>

<xsd:element name="null" type="Templates:null"/>

<xsd:group ref="Values:Value"/>

</xsd:choice>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="all">

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="null">

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="TciValueDifference">

<xsd:sequence>

<xsd:element name="val" type="SimpleTypes:xpath"/>

<xsd:element name="tmpl" type="SimpleTypes:xpath"/>

</xsd:sequence>

<xsd:attributeGroup ref="Values:ValueAtts"/>

<xsd:attribute name="desc" type="SimpleTypes:TString" use="optional"/>

</xsd:complexType>

<xsd:complexType name="TciValueDifferenceList">

<xsd:sequence>

<xsd:element name="diff" type="Templates:TciValueDifference"

maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="SimpleTemplate">

<xsd:choice>

<xsd:element name="value" type="SimpleTypes:TString"/>

<xsd:group ref="Templates:SpecialTemplate"/>

<xsd:element name="null" type="Templates:null"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="RecordTemplate">

<xsd:choice>

<xsd:group ref="Templates:TypedTemplate" minOccurs="0" maxOccurs="unbounded"/>

<xsd:group ref="Templates:SpecialTemplate"/>

<xsd:element name="null" type="Templates:null"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="RecordOfTemplate">

<xsd:choice>

<xsd:element name="integer" type="Templates:SimpleTemplate" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="float" type="Templates:SimpleTemplate" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="boolean" type="Templates:SimpleTemplate" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="verdicttype" type="Templates:SimpleTemplate" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="bitstring" type="Templates:SimpleTemplate"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="hexstring" type="Templates:SimpleTemplate"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="octetstring" type="Templates:SimpleTemplate"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="charstring" type="Templates:SimpleTemplate"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="universal\_charstring"

type="Templates:SimpleTemplate" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="record" type="Templates:RecordTemplate" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="record\_of" type="Templates:RecordOfTemplate"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="array" type="Templates:RecordOfTemplate" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="set" type="Templates:RecordTemplate" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="set\_of" type="Templates:RecordOfTemplate"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="enumerated" type="Templates:SimpleTemplate"

minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="union" type="Templates:UnionTemplate" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="anytype" type="Templates:AnytypeTemplate" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:element name="address" type="Templates:AddressTemplate" minOccurs="0"

maxOccurs="unbounded"/>

<xsd:group ref="Templates:SpecialTemplate"/>

<xsd:element name="null" type="Templates:null"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="UnionTemplate">

<xsd:choice>

<xsd:group ref="Templates:TypedTemplate" />

<xsd:group ref="Templates:SpecialTemplate"/>

<xsd:element name="null" type="Templates:null"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="AnytypeTemplate">

<xsd:choice>

<xsd:group ref="Templates:TypedTemplate" />

<xsd:group ref="Templates:SpecialTemplate"/>

<xsd:element name="null" type="Templates:null"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

<xsd:complexType name="AddressTemplate">

<xsd:choice>

<xsd:group ref="Templates:TypedTemplate" />

<xsd:group ref="Templates:SpecialTemplate"/>

<xsd:element name="null" type="Templates:null"/>

</xsd:choice>

<xsd:attributeGroup ref="Values:ValueAtts"/>

</xsd:complexType>

</xsd:schema>

# B.5 TCI‑TL XML Schema for Events

<?xml version="1.0" encoding="UTF‑8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

targetNamespace="http://uri.etsi.org/ttcn‑3/tci/Events\_v4\_10\_1.xsd"

xmlns:Events="http://uri.etsi.org/ttcn‑3/tci/Events\_v4\_10\_1.xsd"

xmlns:Types="http://uri.etsi.org/ttcn‑3/tci/Types\_v4\_10\_1.xsd"

xmlns:Templates="http://uri.etsi.org/ttcn‑3/tci/Templates\_v4\_10\_1.xsd"

xmlns:SimpleTypes="http://uri.etsi.org/ttcn‑3/tci/SimpleTypes\_v4\_10\_1.xsd"

xmlns:Values="http://uri.etsi.org/ttcn‑3/tci/Values\_v4\_10\_1.xsd"

elementFormDefault="qualified">

<xsd:import namespace="http://uri.etsi.org/ttcn‑3/tci/SimpleTypes\_v4\_10\_1.xsd"

schemaLocation="SimpleTypes\_v4\_10\_1.xsd"/>

<xsd:import namespace="http://uri.etsi.org/ttcn‑3/tci/Types\_v4\_10\_1.xsd" schemaLocation="Types\_v4\_10\_1.xsd"/>

<xsd:import namespace="http://uri.etsi.org/ttcn‑3/tci/Values\_v4\_10\_1.xsd"

schemaLocation="Values\_v4\_10\_1.xsd"/>

<xsd:import namespace="http://uri.etsi.org/ttcn‑3/tci/Templates\_v4\_10\_1.xsd"

schemaLocation="Templates\_v4\_10\_1.xsd"/>

<!‑‑ common definition for all events ‑‑>

<xsd:complexType name="Event" mixed="true">

<xsd:sequence>

<xsd:element name="am" type="SimpleTypes:TString"/>

</xsd:sequence>

<xsd:attribute name="ts" type="xsd:long" use="required"/>

<xsd:attribute name="src" type="SimpleTypes:TString" use="optional"/>

<xsd:attribute name="line" type="SimpleTypes:TInteger" use="optional"/>

<!‑‑ general identifier structure for test components, ports and timer ‑‑>

<xsd:attribute name="name" type="SimpleTypes:TString" use="required"/>

<xsd:attribute name="id" type="SimpleTypes:TString" use="required"/>

<xsd:attribute name="type" type="SimpleTypes:TString" use="required"/>

</xsd:complexType>

<!‑‑ this event is extended by all port configuration events ‑‑>

<xsd:complexType name="PortConfiguration">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="port1" type="Types:TriPortIdType" />

<xsd:element name="port2" type="Types:TriPortIdType" />

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ this event is extended by all port status events ‑‑>

<xsd:complexType name="PortStatus">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="port" type="Types:TriPortIdType"/>

<xsd:element name="stat" type="SimpleTypes:PortStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ testcases ‑‑>

<xsd:complexType name="tliTcExecute">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="tcId" type="Types:TciTestCaseIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="dur" type="SimpleTypes:TriTimerDurationType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTcStart">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="tcId" type="Types:TciTestCaseIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="dur" type="SimpleTypes:TriTimerDurationType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTcStop">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event"/>

<xsd:sequence>

<xsd:element name="reason" type="SimpleTypes:TString" minOccurs="0"/>

</xsd:sequence>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTcStarted">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="tcId" type="Types:TciTestCaseIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="dur" type="SimpleTypes:TriTimerDurationType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTcTerminated">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="tcId" type="Types:TciTestCaseIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="verdict" type="Values:VerdictValue"/>

<xsd:element name="reason" type="SimpleTypes:TString" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ control ‑‑>

<xsd:complexType name="tliCtrlStart">

<xsd:complexContent>

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCtrlStop">

<xsd:complexContent>

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCtrlTerminated">

<xsd:complexContent>

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ asynchronous communication ‑‑>

<xsd:complexType name="tliMSend\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="msgValue" type="Values:Value"/>

<xsd:element name="addrValue " type="Values:Value" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="msg" type="Types:TriMessageType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMSend\_m\_BC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="msgValue" type="Values:Value"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="msg" type="Types:TriMessageType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMSend\_m\_MC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="msgValue" type="Values:Value"/>

<xsd:element name="addrValues" type="Types:TciValueListType" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="msg" type="Types:TriMessageType" minOccurs="0"/>

<xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMSend\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="msgValue" type="Values:Value"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMSend\_c\_BC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdListType" minOccurs="0"/>

<xsd:element name="msgValue" type="Values:Value"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMSend\_c\_MC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdListType" minOccurs="0"/>

<xsd:element name="msgValue" type="Values:Value"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMDetected\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="msgValue" type="Types:TriMessageType"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMDetected\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="msgValue" type="Values:Value"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMMismatch\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="msgValue" type="Values:Value"/>

<xsd:element name="msgTmpl" type="Templates:TciValueTemplate"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMMismatch\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="msgValue" type="Values:Value"/>

<xsd:element name="msgTmpl" type="Templates:TciValueTemplate"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMReceive\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="msgValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="msgTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMReceive\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="msgValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="msgTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ synchronous communication ‑‑>

<xsd:complexType name="tliPrCall\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCall\_m\_BC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCall\_m\_MC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="addrValues" type="Types:TciValueListType" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCall\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCall\_c\_BC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdListType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCall\_c\_MC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdListType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetCallDetected\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetCallDetected\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetCallMismatch\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetCallMismatch\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates: TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetCall\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetCall\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrReply\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType"

minOccurs="0"/>

<xsd:sequence>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

<xsd:element name="transmission‑failure"

type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrReply\_m\_BC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType"

minOccurs="0"/>

<xsd:sequence>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/>

<xsd:element name="transmission‑failure"

type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrReply\_m\_MC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addrValues" type="Types:TciValueListType" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType"

minOccurs="0"/>

<xsd:sequence>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/>

<xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/>

<xsd:element name="transmission‑failure"

type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrReply\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrReply\_c\_BC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdListType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrReply\_c\_MC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdListType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetReplyDetected\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="triPars" type="Types:TriParameterListType" minOccurs="0"/>

<xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetReplyDetected\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetReplyMismatch\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="replTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetReplyMismatch\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="replTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetReply\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="replTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetReply\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="replTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrRaise\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrRaise\_m\_BC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrRaise\_m\_MC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addrValues" type="Types:TciValueListType" minOccurs="0"/>

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:sequence>

<xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/>

<xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrRaise\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrRaise\_c\_BC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdListType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrRaise\_c\_MC">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="to" type="Types:TriPortIdListType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="transmission‑failure" type="SimpleTypes:TriStatusType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatchDetected\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/>

<xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatchDetected\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriPortIdType" minOccurs="0"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatchMismatch\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="excTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatchMismatch\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="excTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatch\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="excTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatch\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="excTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatchTimeoutDetected">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatchTimeout">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ components ‑‑>

<xsd:complexType name="tliCCreate">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType"/>

<xsd:element name="name" type="SimpleTypes:TString"/>

<xsd:element name="hostId" type="Values:Value" minOccurs="0"/>

<xsd:element name="alive" type="SimpleTypes:TBoolean"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCStart">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType"/>

<xsd:element name="name" type="Types:TciBehaviourIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCRunning">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType"/>

<xsd:element name="status" type="SimpleTypes:ComponentStatusType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCAlive">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType"/>

<xsd:element name="status" type="SimpleTypes:ComponentStatusType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCStop">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCKill">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCDoneMismatch">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType"/>

<xsd:element name="compTmpl" type="Templates:TciNonValueTemplate"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCKilledMismatch">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="comp" type="Types:TriComponentIdType"/>

<xsd:element name="compTmpl" type="Templates:TciNonValueTemplate"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCDone">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="compTmpl" type="Templates:TciNonValueTemplate"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCKilled">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="compTmpl" type="Templates:TciNonValueTemplate"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCTerminated">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="verdict" type="Values:VerdictValue" />

<xsd:element name="reason" type="SimpleTypes:TString" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ ports ‑‑>

<xsd:complexType name="tliPConnect">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:PortConfiguration"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPDisconnect">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:PortConfiguration"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPMap">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:PortConfiguration"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPMapParam">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:tliPMap">

<xsd:sequence>

<xsd:element name="tciPars" type="Types:TciParameterListType" />

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:element name="triPars" type="Types:TriParameterListType" />

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPUnmap">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:PortConfiguration"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPUnmapParam">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:tliPUnmap">

<xsd:sequence>

<xsd:element name="tciPars" type="Types:TciParameterListType" />

<xsd:choice>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:element name="triPars" type="Types:TriParameterListType"/>

</xsd:choice>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPClear">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:PortStatus"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPStart">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:PortStatus"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPStop">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:PortStatus"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPHalt">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:PortStatus"/>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ codec ‑‑>

<xsd:complexType name="tliEncode">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="val" type="Values:Value"/>

<xsd:choice>

<xsd:element name="msg" type="Types:TriMessageType"/>

<xsd:element name="encoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

</xsd:choice>

<xsd:element name="codec" type="SimpleTypes:TString"

minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliDecode" mixed="true">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="msg" type="Types:TriMessageType"/>

<xsd:choice>

<xsd:element name="decoder‑failure" type="SimpleTypes:TciStatusType" minOccurs="0"/>

<xsd:element name="val" type="Values:Value"/>

</xsd:choice>

<xsd:element name="codec" type="SimpleTypes:TString"

minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ timers ‑‑>

<xsd:complexType name="tliTTimeoutDetected">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="timer" type="Types:TriTimerIdType" />

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTTimeoutMismatch">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="timer" type="Types:TriTimerIdType" />

<xsd:element name="timerTmpl" type="Templates:TciNonValueTemplate" />

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTTimeout">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="timer" type="Types:TriTimerIdType" />

<xsd:element name="timerTmpl" type="Templates:TciNonValueTemplate" />

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTStart">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="timer" type="Types:TriTimerIdType"/>

<xsd:element name="dur" type="SimpleTypes:TriTimerDurationType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTStop">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="timer" type="Types:TriTimerIdType"/>

<xsd:element name="dur" type="SimpleTypes:TriTimerDurationType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTRead">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="timer" type="Types:TriTimerIdType"/>

<xsd:element name="elapsed" type="SimpleTypes:TriTimerDurationType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliTRunning">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="timer" type="Types:TriTimerIdType"/>

<xsd:element name="status" type="SimpleTypes:TimerStatusType"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ scope ‑‑>

<xsd:complexType name="tliSEnter">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="name" type="Types:QualifiedName" />

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="kind" type="SimpleTypes:TString"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliSLeave">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="name" type="Types:QualifiedName" />

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="returnValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="kind" type="SimpleTypes:TString"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ variables and module parameter ‑‑>

<xsd:complexType name="tliVar">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="name" type="Types:QualifiedName" />

<xsd:element name="val" type="Values:Value" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliModulePar">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="name" type="Types:QualifiedName" />

<xsd:element name="val" type="Values:Value" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ verdicts ‑‑>

<xsd:complexType name="tliGetVerdict">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="verdict" type="Values:VerdictValue"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliSetVerdict">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="verdict" type="Values:VerdictValue"/>

<xsd:element name="reason" type="SimpleTypes:TString" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ log ‑‑>

<xsd:complexType name="tliLog">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="log" type="SimpleTypes:TString"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<!‑‑ alt ‑‑>

<xsd:complexType name="tliAEnter">

<xsd:complexContent>

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliALeave">

<xsd:complexContent>

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliADefaults">

<xsd:complexContent>

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliAActivate">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="name" type="Types:QualifiedName" />

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="ref" type="Values:Value"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliADeactivate">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="ref" type="Values:Value"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliANomatch">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliARepeat">

<xsd:complexContent>

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliAWait">

<xsd:complexContent>

<xsd:extension base="Events:Event"/>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliAction">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="action" type="SimpleTypes:TString"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMatch">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="expr" type="Values:Value"/>

<xsd:element name="tmpl" type="Templates:TciValueTemplate"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMatchMismatch">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="expr" type="Values:Value"/>

<xsd:element name="tmpl" type="Templates:TciValueTemplate"/>

<xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliInfo">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="level" type="SimpleTypes:TInteger"/>

<xsd:element name="info" type="SimpleTypes:TString"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMChecked\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="msgValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="msgTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliMChecked\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="msgValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="msgTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetCallChecked\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetCallChecked\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetReplyChecked\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="replTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrGetReplyChecked\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="tciPars" type="Types:TciParameterListType" minOccurs="0"/>

<xsd:element name="parsTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="replValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="replTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatchChecked\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="excTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliPrCatchChecked\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="signature" type="Types:TriSignatureIdType"/>

<xsd:element name="excValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="excTmpl" type="Templates:TciValueTemplate" minOccurs="0"/>

<xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCheckedAny\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate"

minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCheckedAny\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriComponentIdType"

minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate"

minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCheckMismatch\_m">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="addrValue" type="Values:Value" minOccurs="0"/>

<xsd:element name="addressTmpl" type="Templates:TciValueTemplate"

minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliCheckMismatch\_c">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="at" type="Types:TriPortIdType"/>

<xsd:element name="from" type="Types:TriComponentIdType"

minOccurs="0"/>

<xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate"

minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliRnd">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="val" type="Values:FloatValue"/>

<xsd:element name="from" type="Values:FloatValue"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="tliEvaluate">

<xsd:complexContent mixed="true">

<xsd:extension base="Events:Event">

<xsd:sequence>

<xsd:element name="name" type="Types:QualifiedName" />

<xsd:element name="evalResult" type="Values:Value" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

</xsd:schema>

# B.6 TCI‑TL XML Schema for a Log

<?xml version="1.0" encoding="UTF‑8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

targetNamespace="http://uri.etsi.org/ttcn‑3/tci/TLI\_v4\_10\_1.xsd"

xmlns:TLI="http://uri.etsi.org/ttcn‑3/tci/TLI\_v4\_10\_1.xsd"

xmlns:Events="http://uri.etsi.org/ttcn‑3/tci/Events\_v4\_10\_1.xsd"

elementFormDefault="qualified">

<xsd:import namespace="http://uri.etsi.org/ttcn‑3/tci/Events\_v4\_10\_1.xsd"

schemaLocation="Events\_v4\_10\_1.xsd"/>

<xsd:element name="logfile" type="TLI:LogModule"/>

<xsd:complexType name="LogModule">

<xsd:sequence>

<xsd:element name="header" type="TLI:Header"/>

<xsd:element name="body" type="TLI:Body"/>

<xsd:element name="trailer" type="TLI:Trailer" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="Header">

<xsd:sequence>

<!‑‑ logging version ‑‑>

<xsd:element name="version" type="xsd:string"/>

<!‑‑ begin of the log ‑‑>

<xsd:element name="ts" type="xsd:long"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="Trailer">

<xsd:choice>

<xsd:any namespace="##any" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>

</xsd:choice>

</xsd:complexType>

<xsd:complexType name="Body">

<xsd:choice maxOccurs="unbounded">

<!‑‑ test cases operations ‑‑>

<xsd:element name="tliTcExecute" type="Events:tliTcExecute"/>

<xsd:element name="tliTcStart" type="Events:tliTcStart"/>

<xsd:element name="tliTcStop" type="Events:tliTcStop"/>

<xsd:element name="tliTcStarted" type="Events:tliTcStarted"/>

<xsd:element name="tliTcTerminated" type="Events:tliTcTerminated"/>

<!‑‑ control operations ‑‑>

<xsd:element name="tliCtrlStart" type="Events:tliCtrlStart"/>

<xsd:element name="tliCtrlStop" type="Events:tliCtrlStop"/>

<xsd:element name="tliCtrlTerminated" type="Events:tliCtrlTerminated"/>

<!‑‑ asynchronous communication ‑‑>

<xsd:element name="tliMSend\_m" type="Events:tliMSend\_m"/>

<xsd:element name="tliMSend\_c" type="Events:tliMSend\_c"/>

<xsd:element name="tliMSend\_m\_BC" type="Events:tliMSend\_m\_BC"/>

<xsd:element name="tliMSend\_c\_BC" type="Events:tliMSend\_c\_BC"/>

<xsd:element name="tliMSend\_m\_MC" type="Events:tliMSend\_m\_MC"/>

<xsd:element name="tliMSend\_c\_MC" type="Events:tliMSend\_c\_MC"/>

<xsd:element name="tliMDetected\_m" type="Events:tliMDetected\_m"/>

<xsd:element name="tliMDetected\_c" type="Events:tliMDetected\_c"/>

<xsd:element name="tliMMismatch\_m" type="Events:tliMMismatch\_m"/>

<xsd:element name="tliMMismatch\_c" type="Events:tliMMismatch\_c"/>

<xsd:element name="tliMReceive\_m" type="Events:tliMReceive\_m"/>

<xsd:element name="tliMReceive\_c" type="Events:tliMReceive\_c"/>

<!‑‑ synchronous communication ‑‑>

<xsd:element name="tliPrCall\_m" type="Events:tliPrCall\_m"/>

<xsd:element name="tliPrCall\_c" type="Events:tliPrCall\_c"/>

<xsd:element name="tliPrCall\_m\_BC" type="Events:tliPrCall\_m\_BC"/>

<xsd:element name="tliPrCall\_c\_BC" type="Events:tliPrCall\_c\_BC"/>

<xsd:element name="tliPrCall\_m\_MC" type="Events:tliPrCall\_m\_MC"/>

<xsd:element name="tliPrCall\_c\_MC" type="Events:tliPrCall\_c\_MC"/>

<xsd:element name="tliPrGetCallDetected\_m" type="Events:tliPrGetCallDetected\_m"/>

<xsd:element name="tliPrGetCallDetected\_c" type="Events:tliPrGetCallDetected\_c"/>

<xsd:element name="tliPrGetCallMismatch\_m" type="Events:tliPrGetCallMismatch\_m"/>

<xsd:element name="tliPrGetCallMismatch\_c" type="Events:tliPrGetCallMismatch\_c"/>

<xsd:element name="tliPrGetCall\_m" type="Events:tliPrGetCall\_m"/>

<xsd:element name="tliPrGetCall\_c" type="Events:tliPrGetCall\_c"/>

<xsd:element name="tliPrReply\_m" type="Events:tliPrReply\_m"/>

<xsd:element name="tliPrReply\_c" type="Events:tliPrReply\_c"/>

<xsd:element name="tliPrReply\_m\_BC" type="Events:tliPrReply\_m\_BC"/>

<xsd:element name="tliPrReply\_c\_BC" type="Events:tliPrReply\_c\_BC"/>

<xsd:element name="tliPrReply\_m\_MC" type="Events:tliPrReply\_m\_MC"/>

<xsd:element name="tliPrReply\_c\_MC" type="Events:tliPrReply\_c\_MC"/>

<xsd:element name="tliPrGetReplyDetected\_m" type="Events:tliPrGetReplyDetected\_m"/>

<xsd:element name="tliPrGetReplyDetected\_c" type="Events:tliPrGetReplyDetected\_c"/>

<xsd:element name="tliPrGetReplyMismatch\_m" type="Events:tliPrGetReplyMismatch\_m"/>

<xsd:element name="tliPrGetReplyMismatch\_c" type="Events:tliPrGetReplyMismatch\_c"/>

<xsd:element name="tliPrGetReply\_m" type="Events:tliPrGetReply\_m"/>

<xsd:element name="tliPrGetReply\_c" type="Events:tliPrGetReply\_c"/>

<xsd:element name="tliPrRaise\_m" type="Events:tliPrRaise\_m"/>

<xsd:element name="tliPrRaise\_c" type="Events:tliPrRaise\_c"/>

<xsd:element name="tliPrRaise\_m\_BC" type="Events:tliPrRaise\_m\_BC"/>

<xsd:element name="tliPrRaise\_c\_BC" type="Events:tliPrRaise\_c\_BC"/>

<xsd:element name="tliPrRaise\_m\_MC" type="Events:tliPrRaise\_m\_MC"/>

<xsd:element name="tliPrRaise\_c\_MC" type="Events:tliPrRaise\_c\_MC"/>

<xsd:element name="tliPrCatchDetected\_m" type="Events:tliPrCatchDetected\_m"/>

<xsd:element name="tliPrCatchDetected\_c" type="Events:tliPrCatchDetected\_c"/>

<xsd:element name="tliPrCatchMismatch\_m" type="Events:tliPrCatchMismatch\_m"/>

<xsd:element name="tliPrCatchMismatch\_c" type="Events:tliPrCatchMismatch\_c"/>

<xsd:element name="tliPrCatch\_m" type="Events:tliPrCatch\_m"/>

<xsd:element name="tliPrCatch\_c" type="Events:tliPrCatch\_c"/>

<xsd:element name="tliPrCatchTimeoutDetected"

type="Events:tliPrCatchTimeoutDetected "/>

<xsd:element name="tliPrCatchTimeout" type="Events:tliPrCatchTimeout"/>

<!‑‑ components ‑‑>

<xsd:element name="tliCCreate" type="Events:tliCCreate"/>

<xsd:element name="tliCStart" type="Events:tliCStart"/>

<xsd:element name="tliCRunning" type="Events:tliCRunning"/>

<xsd:element name="tliCAlive" type="Events:tliCAlive"/>

<xsd:element name="tliCStop" type="Events:tliCStop"/>

<xsd:element name="tliCKill" type="Events:tliCKill"/>

<xsd:element name="tliCDoneMismatch" type="Events:tliCDoneMismatch"/>

<xsd:element name="tliCDone" type="Events:tliCDone"/>

<xsd:element name="tliCKilledMismatch" type="Events:tliCKilledMismatch"/>

<xsd:element name="tliCKilled" type="Events:tliCKilled"/>

<xsd:element name="tliCTerminated" type="Events:tliCTerminated"/>

<!‑‑ ports ‑‑>

<xsd:element name="tliPConnect" type="Events:tliPConnect"/>

<xsd:element name="tliPDisconnect" type="Events:tliPDisconnect"/>

<xsd:element name="tliPMap" type="Events:tliPMap"/>

<xsd:element name="tliPMapParam" type="Events:tliPMapParam"/>

<xsd:element name="tliPUnmap" type="Events:tliPUnmap"/>

<xsd:element name="tliPUnmapParam" type="Events:tliPUnmapParam"/>

<xsd:element name="tliPClear" type="Events:tliPClear"/>

<xsd:element name="tliPStart" type="Events:tliPStart"/>

<xsd:element name="tliPStop" type="Events:tliPStop"/>

<xsd:element name="tliPHalt" type="Events:tliPHalt"/>

<!‑‑ codec ‑‑>

<xsd:element name="tliDecode" type="Events:tliDecode"/>

<xsd:element name="tliEncode" type="Events:tliEncode"/>

<!‑‑ timers ‑‑>

<xsd:element name="tliTTimeoutDetected" type="Events:tliTTimeoutDetected"/>

<xsd:element name="tliTTimeoutMismatch" type="Events:tliTTimeoutMismatch"/>

<xsd:element name="tliTTimeout" type="Events:tliTTimeout"/>

<xsd:element name="tliTStart" type="Events:tliTStart"/>

<xsd:element name="tliTStop" type="Events:tliTStop"/>

<xsd:element name="tliTRead" type="Events:tliTRead"/>

<xsd:element name="tliTRunning" type="Events:tliTRunning"/>

<!‑‑ scopes ‑‑>

<xsd:element name="tliSEnter" type="Events:tliSEnter"/>

<xsd:element name="tliSLeave" type="Events:tliSLeave"/>

<!‑‑ statements ‑‑>

<xsd:element name="tliVar" type="Events:tliVar"/>

<xsd:element name="tliModulePar" type="Events:tliModulePar"/>

<xsd:element name="tliGetVerdict" type="Events:tliGetVerdict"/>

<xsd:element name="tliSetVerdict" type="Events:tliSetVerdict"/>

<xsd:element name="tliLog" type="Events:tliLog"/>

<!‑‑ alt ‑‑>

<xsd:element name="tliAEnter" type="Events:tliAEnter"/>

<xsd:element name="tliALeave" type="Events:tliALeave"/>

<xsd:element name="tliADefaults" type="Events:tliADefaults"/>

<xsd:element name="tliAActivate" type="Events:tliAActivate"/>

<xsd:element name="tliADeactivate" type="Events:tliADeactivate"/>

<xsd:element name="tliANomatch" type="Events:tliANomatch"/>

<xsd:element name="tliARepeat" type="Events:tliARepeat"/>

<xsd:element name="tliAWait" type="Events:tliAWait"/>

<!‑‑ action ‑‑>

<xsd:element name="tliAction" type="Events:tliAction"/>

<!‑‑ match ‑‑>

<xsd:element name="tliMatch" type="Events:tliMatch"/>

<xsd:element name="tliMatchMismatch" type="Events:tliMatchMismatch"/>

<!‑‑ info ‑‑>

<xsd:element name="tliInfo" type="Events:tliInfo"/>

<!‑‑ check ‑‑>

<xsd:element name="tliMChecked\_m" type="Events:tliMChecked\_m"/>

<xsd:element name="tliMChecked\_c" type="Events:tliMChecked\_c"/>

<xsd:element name="tliPrGetCallChecked\_m" type="Events:tliPrGetCallChecked\_m"/>

<xsd:element name="tliPrGetCallChecked\_c" type="Events:tliPrGetCallChecked\_c"/>

<xsd:element name="tliPrGetReplyChecked\_m" type="Events:tliPrGetReplyChecked\_m"/>

<xsd:element name="tliPrGetReplyChecked\_c" type="Events:tliPrGetReplyChecked\_c"/>

<xsd:element name="tliPrCatchChecked\_m" type="Events:tliPrCatchChecked\_m"/>

<xsd:element name="tliPrCatchChecked\_c" type="Events:tliPrCatchChecked\_c"/>

<xsd:element name="tliCheckedAny\_m" type="Events:tliCheckedAny\_m"/>

<xsd:element name="tliCheckedAny\_c" type="Events:tliCheckedAny\_c"/>

<xsd:element name="tliCheckAnyMismatch\_m" type="Events:tliCheckAnyMismatch\_m"/>

<xsd:element name="tliCheckAnyMismatch\_c" type="Events:tliCheckAnyMismatch\_c"/>

<!-- rnd -->

<xsd:element name="tliRnd" type="Events:tliRnd"/>

<!-- evaluation of @lazy and @fuzzy variables -->

<xsd:element name="tliEvaluate" type="Events:tliEvaluate"/>

</xsd:choice>

</xsd:complexType>

</xsd:schema>