

IEEE 1671.1 Corrigendum Change Summary

DRAFT 2 - 2023-04-13

PAR

- **Scope of the Proposed changes:** Make corrections to the XML Schema files to achieve full consistency with the Standard.
- **Additional Explanatory Notes:** The XML Schema files published with the standard are incorrect, which limits their use for document validation.

Question Re: Precedence

The *TestDescription* schema (specified in files “TestDescription.xsd”, “DatumTypes.xsd”, “IeeeStd1641.xsd”, and “Operations.xsd”) element, child element, and annotation information shall take precedence over the descriptive information contained in [Clause 4](#).

4. Test Description schema

?

4.1 Applicability

The *TestDescription* schema, specified in [Annex A](#) of this document, provides a standard format for the representation of test description information.

Annex A

(normative)

XML schemas

The *TestDescription* schema and the material contained in [Clause 4](#) shall take precedence over the example information represented in [Annex F](#).

The *ParallelDigital* schema (specified in file “ParallelDigitalTSFLib.xsd”) element, child element, and annotation information shall take precedence over the example information represented in [Annex F](#).

The *SerialDigital* schema (specified in file “SerialDigitalTSFLib.xsd”) element, child element, and annotation information shall take precedence over the example information represented in [Annex F](#).

Annex F

(informative)

Users information and examples

- The PAR relies on the assumption that “Annex A (normative)” takes precedence over the XML schemas.
- However, the text of the standard does not explicitly state this precedence relationship. It only specifies the relationship between schemas and the descriptive text in Annex 4. Note that this issue is not specific to 1671.1; the text appears in many (all?) of the “dot” standards.
- Were we correct in assuming that “Annex A (normative)” takes precedence over the XML schemas (ex. because the Annex is normative and the schemas are “support material”)?

Issue 1. Missing TestResult@ID uniqueness constraint (1)

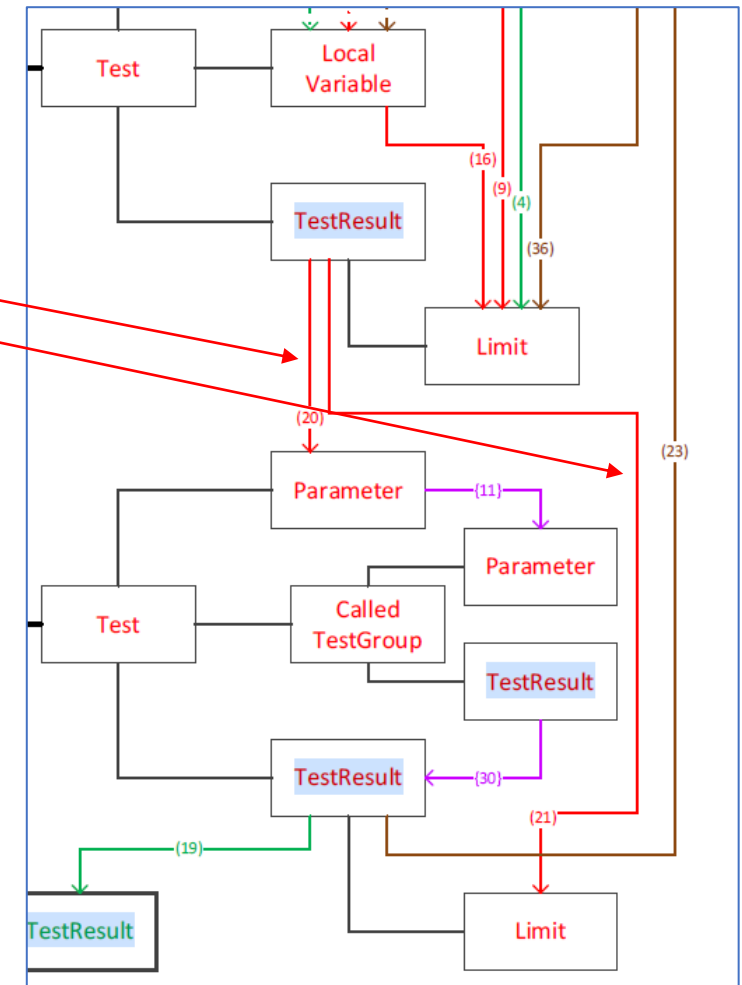
- TestResult@ID needs to be unique within the instance document in order to support unambiguous references from Parameters and Limits of other Tests
- The standard document identifies it as unique, but the schema does not.

The elements `td:Test/td:TestResults/td:TestResult` shall describe the test results of the test. The attributes of the `td:TestResult` type shall be used as follows:

- `ID`: The attribute shall contain a unique identifier. Refer to 5.3 for additional information on identifiers.

A.1.384 element Test/TestResults/TestResult

type	td:TestResult			
children	td:Description td:ValueDescription td:ValueTo td:Extension td:Transform td:TestLimits			
attributes	Name	Type	Use	Description
	ID	c:NonBlankString	required	Unique identifier, enables referencing the result.
	name	c:NonBlankString	required	Name of the test result.



Issue 1. Missing TestResult@ID uniqueness constraint (2)

- Implemented changes: make schema consistent with the standard text through the following changes:
 - In support file *TestDescription.xsd*:
 - 1. Add xs:key under td:TestDescription/td:DetailedTestInformation
 - 2. Increment schema version from 3.00 to 3.10. No namespace change because the constraint existed in the text of the standard. All compliant instances documents should pass validation. If they do not, they were not compliant.
 - In standard document
 - 1. Add xs:key to table in clause A.1.389, *element TestDescription/DetailedTestInformation*

```
<xs:key name="testKey">
  <xs:selector xpath="//td:Tests/td:Test"/>
  <xs:field xpath="@ID"/>
</xs:key>
<xs:key name="testResultKey">
  <xs:selector xpath="//td:Tests/td:Test/td:TestResults/td:TestResult"/>
  <xs:field xpath="@ID"/>
</xs:key>
<xs:keyref name="testInitializationRef" refer="td:testKey">
  <xs:selector xpath="//td:Initialization/td:Test"/>
  <xs:field xpath="@testID"/>
</xs:keyref>
```

▲ <> td:Test <1>

ID	test1
name	V _{CC} to GND Resistance Test
td:Behavior	<td:Description>With power off, measure resistance between V _{CC} and GND.
td:Outcomes	<td:Outcome> (2)
td:TestResults	
td:TestResult	
ID	t1tr1
name	V _{CC} Resistance
td:ValueDescription	
td:TestLimits	

▲ <> td:Test <2>

ID	test2
name	V _O DC Voltage Test
td:Parameters	
td:Behavior	<td:Description>With power on, measure DC voltage at the output
td:Outcomes	<td:Outcome> (2)
td:TestResults	
td:TestResult	
ID	t1tr1
name	V _O Resistance
td:ValueDescription	
td:TestLimits	

❗ Duplicate value: [t1tr1] was already matched by the <key> identity constraint 'td:testResultKey' within the scope of element <td:DetailedTestInformation>.

Issue 2. Missing TestGroupParameter@ID uniqueness constraint (1)

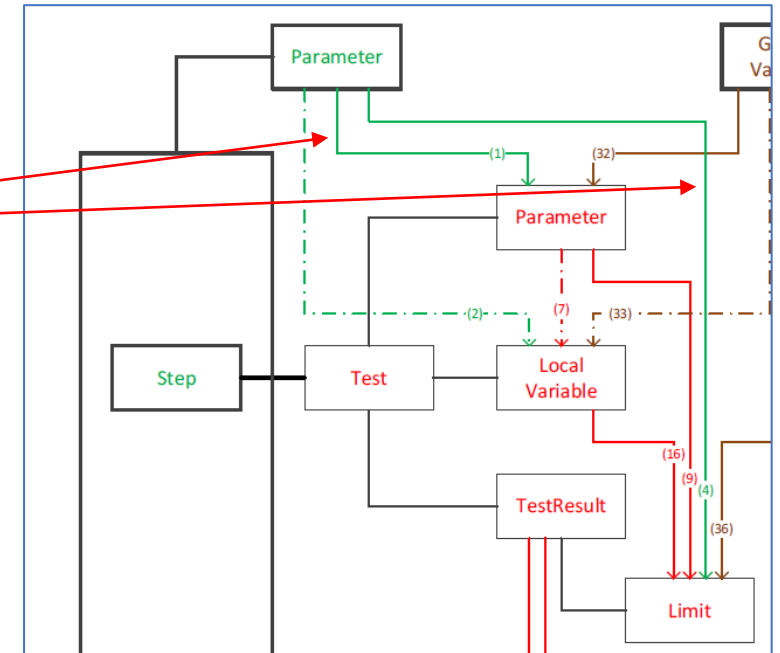
- TestGroupParameter@ID needs to be unique within the instance document in order to support unambiguous references from Parameters and Limits of Tests
- The standard document identifies it as unique , but the schema does not.

The elements *td:TestGroup/td:ParameterDescriptions/td:ParameterDescription* shall describe the parameters of the test group. The attributes of the *td:ParameterDescription* element shall be used as follows:

- a) 'ID': The attribute shall contain a unique identifier. Refer to 5.3 for additional information on identifiers.

A.1.397 element TestGroup/ParameterDescriptions/ParameterDescription

children	td:Description td:ValueDescription td:Extension			
attributes	Name	Type	Use	Description
	ID	c:NonBlankString	required	Unique identifier, enables referencing the parameter description.
	name	c:NonBlankString	required	Name of the parameter.



Issue 2. Missing TestGroupParameter@ID uniqueness constraint (2)

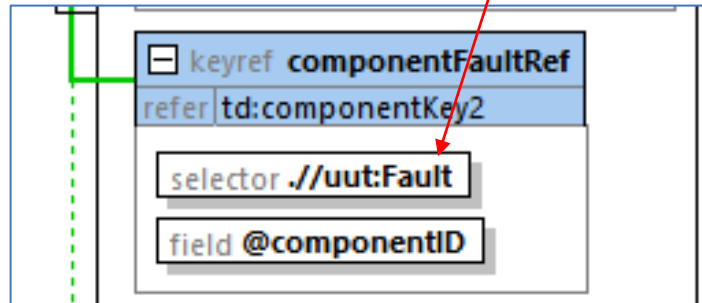
- Implemented change: make schema consistent with the standard text through the following changes:
 - In support file *TestDescription.xsd*:
 - 1. Add xs:key under td:TestDescription/td:DetailedTestInformation
 - 2. Increment schema version from 3.00 to 3.10. No namespace change because the constraint existed in the text of the standard. All compliant instances documents should pass validation. If they do not, they were not compliant.
 - In standard document
 - 1. Add xs:key to table in clause A.1.389, *element TestDescription/DetailedTestInformation*

```
<xs:key name="testGroupKey">
  <xs:selector xpath="//td:TestGroup"/>
  <xs:field xpath="@ID"/>
</xs:key>
<xs:key name="testGroupParameterDescriptionKey">
  <xs:selector xpath="//td:TestGroup/td:ParameterDescriptions/td:ParameterDescription"/>
  <xs:field xpath="@ID"/>
</xs:key>
<xs:keyref name="testGroupEntryPointRef" refer="td:testGroupKey">
  <xs:selector xpath="//td:EntryPoint"/>
  <xs:field xpath="@testGroupID"/>
</xs:keyref>
```

The screenshot displays an XML tree view of a test group structure. The first test group (seq1) has a parameter description with ID 'pd1'. The second test group (seq2) also has a parameter description with ID 'pd1'. A red error message is shown at the bottom of the second parameter description, stating: "Duplicate value: [pd1] was already matched by the <key> identity constraint 'td:testGroupParameterDescriptionKey' within the scope of element <td:DetailedTestInformation>." The error message is highlighted in yellow.

Issue 3. Incomplete Fault@componentID referential integrity constraint (1)

- Faults can be defined as
 - /td:TestDescription/td:UUT/td:Description/c:Definition/uut:Faults/uut:Fault
 - or as
 - /td:TestDescription/td:DetailedTestInformation/td:Faults /td:Fault
- The constraint currently specified through componentFaultRef works for the first definition only:



- Additionally, the XML schema definition of “componentFaultRef” is inconsistent with the table in Clause A.1.1 of the standard: the table specifies the selector “./td:Fault”, while the XML schema contains “./uut:Fault”.
- These two problems can be resolved together, through schema and standard changes.

Issue 3. Incomplete Fault@componentID referential integrity constraint (2)

- Implemented change:
 - In support file *TestDescription.xsd*:
 1. Correct existing xs:keyref "componentFaultRef" under td:TestDescription, to reference "../td:Fault" instead of "../uut:Fault"
 2. Add new xs:keyref "componentFaultRef2", referencing ../uut:Fault
 3. Increment schema version from 3.00 to 3.10. No namespace change because the this change adds a capability. The instance documents that passed validation before will still pass.
 - In standard document
 1. Add xs:keyref "componentFaultRef2" to table in clause A.1.1, *element TestDescription*

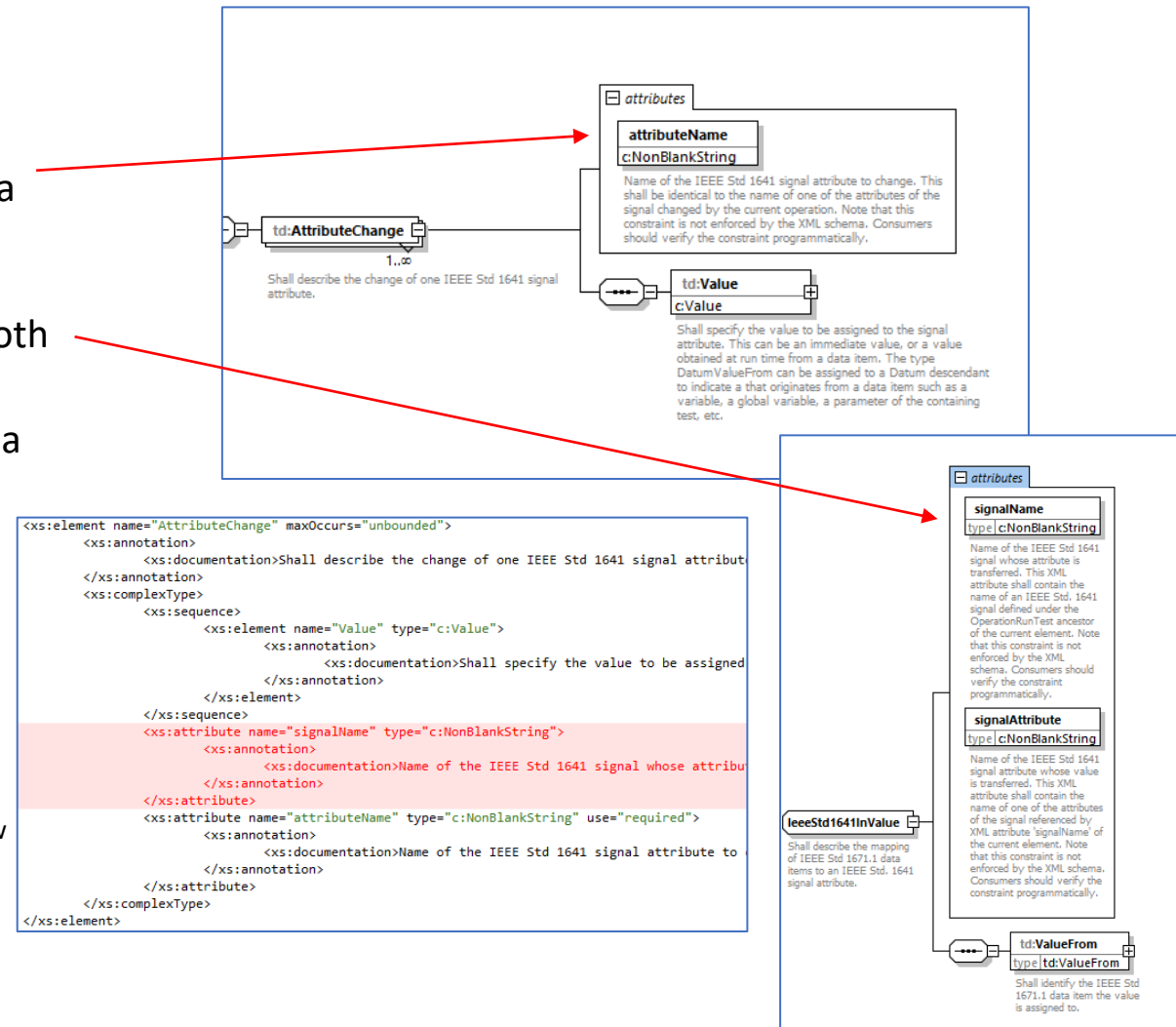
```
<xs:keyref name="componentFaultRef" refer="td:componentKey2">
  <xs:selector xpath="../uut:Fault"/>
  <xs:field xpath="@componentID"/>
</xs:keyref>
<xs:keyref name="componentFaultRef2" refer="td:componentKey2">
  <xs:selector xpath="../td:Fault"/>
  <xs:field xpath="@componentID"/>
</xs:keyref>
```

componentID	detectability	ID	uut:FailureMode	td:LocationItem
c1a	DetectableInsertable		FMT	

The value [c1a] matched by the <keyref> identity constraint 'td:componentFaultRef2' was not matched by the referenced identity constraint 'td:componentKey2' within the scope of element <td:TestDescription>.

Issue 4. Missing signal name reference in AttributeChange (1)

- The “td:ieeeStd1641AttributeChanges/ td:AttributeChange” element does not contain a reference to a *signal* name, only a reference to an *attribute* name.
- It should be possible to reference a signal name as well. This would be consistent with “ieeeStd1641InValue”, which has both signal name and attribute name references.
- The signal name attribute should be optional. In some cases, a signal reference is not necessary (ex. if the signal definition is regarded as an “anonymous TSF” and there is no ambiguity WRT attribute names).
- Implemented changes:
 - In support file *ieeeStd1641.xsd*:
 - 1. Add optional attribute “signalName” to element “td:ieeeStd1641AttributeChanges/td:AttributeChange”
 - 2. Increment schema version from 3.00 to 3.10
 - 3. No namespace change. This is a non-breaking change because the new attribute is optional.
 - In standard document:
 - 1. Add attribute to table in clause A.1.111, *element ieeeStd1641AttributeChanges/AttributeChange*

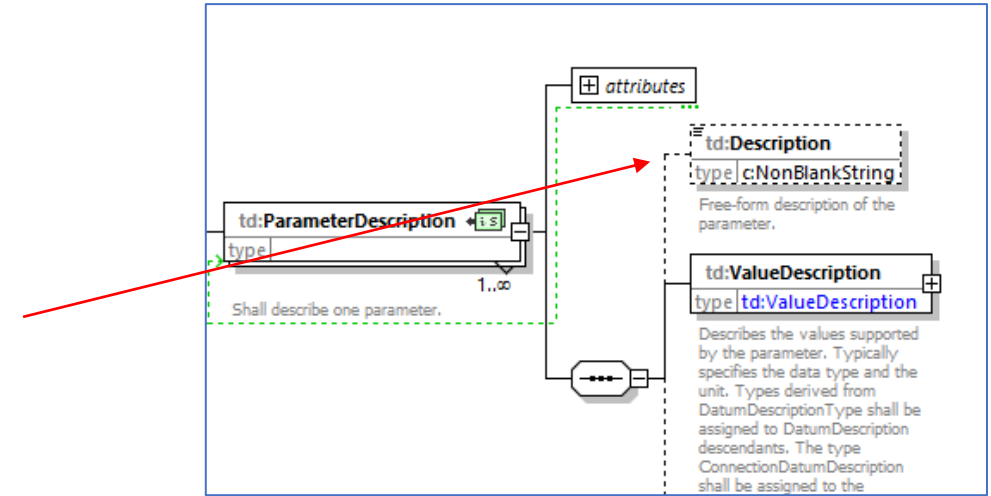


Issue 5. Incorrect annotation of @signalName attributes in XML schema

- In the XML schema, the annotations of attribute @signalName of complex types leeeStd1641InValue, leeeStd1641OutValue, and leeeStd1641PortValue contain the sentence “This XML attribute shall contain the name of an IEEE Std. 1641 signal defined under the OperationRunTest ancestor of the current element.”
- The descriptive text contained in the standard is correct: “This XML attribute shall contain the name of an IEEE Std. 1641 signal defined under the leeeStd1641 ancestor of the current element.”
- Implemented changes:
 - In support file *leeeStd1641.xsd*:
 - 1. change annotation to “This XML attribute shall contain the name of an IEEE Std. 1641 signal defined under the leeeStd1641 ancestor of the current element”
 - 2. Increment schema version from 3.00 to 3.10. No namespace change. This is a non-breaking change.

Issue 6. Missing “Description” Element in TestResultDescription

- Element “td:TestGroup/td:TestResultDescriptions/td:TestResultDescription” should have a “Description” child element , for consistency with “td:TestGroup/td:ParameterDescriptions/td:ParameterDescription”
- Implemented changes:
 - In support file *TestDescription.xsd*:
 - 1.Add element td:Description under td:TestGroup/td:TestResultDescriptions/td:TestResultDescription
 - 2. Increment schema version from 3.00 to 3.10
 - In standard document:
 - 1. Add new clause for element TestGroup/TestResultDescriptions/TestResultDescription. To be located after clause A.1.403.



```
<xs:element name="TestResultDescription" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>Shall describe one result produced by th
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Description" type="c:NonBlankStr
        <xs:annotation>
          <xs:documentation>Free-form descri
        </xs:annotation>
      </xs:element>
      <xs:element name="ValueDescription" type="td:Value
        <xs:annotation>
          <xs:documentation>Shall describe t
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Issue 7. Unnecessary Element td:TestGroupOutcomeReference in td:Result

- The td:Result/td:TestGroupOutcomeReference element is no longer necessary because Steps no longer describe the execution of a Test Group
- Proposed changes:
 - In schema file *TestDescription.xsd*:
 - The WG decided (in the August 2022 meeting) against removing element td:Result/td:TestGroupOutcomeReference and the “choice” element, because this would have been a breaking change. Instead, mark element as “deprecated and schedule for removal in the next revision of the standard.
 - 1. Add “Deprecated” to annotation.
 - 2. Increment schema version from 3.00 to 3.10. No namespace change. Although an element gets deleted, this is not a breaking change because the element could not have been used. The condition “... the ancestor Step element represents the execution of a test group” is never true with the current version of the schema.
 - In support file *IEEE 1671.1 - Example2.xml*:
 - 1. Replace all “td:TestGroupOutcomeReference” with “td:TestOutcomeReference” elements
 - 2. Replace all “testGroupOutcomeID” attribute with “testOutcomeID”
 - In standard document:
 - 1. Add to annotation in clause A.1.324 the following sentence: “Note that this element is deprecated and cannot be used when creating new instance documents.”

Issue 7a. Inconsistent ID format in Example 2

- IDs of Outcomes for Tests with ID = test8, test9, and test10 are prefixed with “s” instead of “t”
 - This is not an error, but is inconsistent with the ID formatting conventions used throughout the examples. These conventions improve readability.
- Implemented changes:
 - In support file *IEEE 1671.1 - Example2.xml*:
 - Change IDs of Outcomes for Tests with ID = test8, test9, and test10. Change ID references accordingly.

```
<td:Test ID="test8" name="Call_SafeToTurnOn">
  <td:Behavior>
    <td:TestGroupCall testGroupID="seq2"/>
  </td:Behavior>
  <td:Outcomes>
    <td:Outcome ID="t8o1" value="Passed"/>
    <td:Outcome ID="t8o2" value="Failed" qualifier="C3.SR"/>
  </td:Outcomes>
</td:Test>
```

```
<td:Step ID="step1" testID="test8">
  <td:Results>
    <td:Result>
      <td:TestOutcomeReference testOutcomeID="t8o1"/>
      <td:NextStep stepID="step2"/>
    </td:Result>
    <td:Result>
      <td:TestOutcomeReference testOutcomeID="t8o2"/>
      <td:SequenceOutcome sequenceOutcomeID="s1o2"/>
    </td:Result>
  </td:Results>
</td:Step>
```

Issue 8. Ambiguous “OutValue” Element in Example 2

- In td:Test with ID="test2", the first OutValue of OperationRunTest assigns the “Conn2” signal reference to a Global Variable. The meaning of this assignment is ambiguous:
 - The intent is to store a reference to the Connection signal in a Global Variable, in order to manipulate the connection in a subsequent Operation.
 - This assignment could also be interpreted as storing the value of the “Conn2.measurement” attribute in the Global Variable (which would not make sense because “Conn2” does not have a “measurement” attribute). Note that in IEEE 1641 “measurement” is the default name when an attribute name is not specified.
- All other Operation types use specialized “SignalRef” elements for describing the assignment of signal references
- WG discussion in the August 2022 meeting
 - The correct solution for specifying that a reference to the connection signal is stored is to add @attributeName = “Out” to the td:OutValue element with @signalName = “Conn2”.
 - However, this is against the purpose of OperationRunTest (i.e., run an encapsulated test).
 - For now, correct Example 2. In the next revision, add text clarifying the assignment of attributes.
- Implemented solution:
 - 1. Delete the “OutValue” element from Example 2.

```
<td:OutValues>
  <!--Example - referencing a global signal-->
  <td:OutValue signalName="Conn2">
    <td:ValueTo type="GlobalVariable" refID="gs1"/>
  </td:OutValue>
  <td:OutValue signalName="Meas1">
    <td:ValueTo type="TestResult" refID="t2tr1"/>
  </td:OutValue>
</td:OutValues>
```

Issue 9. Incorrect TSF class name in Example 2

- In td:Test with ID="test3", the "VO_3" signal is of type DC_SIGNAL. This is inconsistent with the following:
 - The code comment "Measure Amplitude of AC_SIGNAL at J1-3, J1-4"
 - The purpose of the signal - specify the expected amplitude range and frequency for the signal being measured.
- Implemented change:
 - In support file *IEEE 1671.1 - Example2.xml*, rename tsf:DC_SIGNAL to tsf:AC_SIGNAL.

```
<td:Operation ID="op1" xsi:type="td:OperationRunTest">
  <std:Signal Out="Meas4">
    <!--Apply AC_SIGNAL at J1-1, J1-2-->
    <tsf:AC_SIGNAL name="VIN1" ac_ampl="4.0mV" freq="1.0kHz"/>
    <std:TwoWire name="Conn10" hi="J1-1" lo="J1-2" In="VIN1"/>
    <!--Measure Amplitude of AC_SIGNAL at J1-3, J1-4-->
    <std:TwoWire name="Conn17" hi="J1-3" lo="J1-4"/>
    <tsf:AC_SIGNAL name="V_O3" ac_ampl="range 0mV to 100mV errlmt +- 0.1mV" freq="1000Hz"/>
    <std:Measure name="Meas4" As="V_O3" attribute="ac_ampl" In="Conn17"/>
  </std:Signal>
```


Issue 10. Missing “Out” attributes in signal definitions from examples

- The source signals do not have an Out attribute. This is currently needed when converting the XML to TPL (ex. for code generation).
 - Note that the Sensor signals do have Out attributes. The value of the attribute is the @name of the std:Measure signal.
- WG discussion in the August 2022 meeting
 - This is resolved in 1641-2022, where for signals without an “Out” attribute it is assumed that all contained signals become “Out” signals.
 - This is also resolved in the latest OSA-RTS releases.
 - For now, make no changes. Schedule the change for the next revision, along with changing all references to IEEE 1671-2022.

```
<td:Operation xsi:type="td:OperationSetup" ID="op1">
  <!--Example: source signal defined locally-->
  <td:Description>Define source signal for UUT DC power
  <td:Signal>
    <td:SignalRef type="GlobalVariable" refID="g
    <std:Signal Out="DcSignal1">
      <tsf:DC_SIGNAL name="DcSignal1" dc_e
    </std:Signal>
  </td:Signal>
</td:Operation>
```

Issue 11. Incorrect key in DetailedTestInformation/Tests/Test

- Selector of outcomeKey1 is printed in the standard as “./td:Outcomes/dotcom”. Should be “./td:Outcomes/td:Outcome”
- Note that the schema is correct.
- Implemented change:
 - In standard document, correct outcomeKey1

Editorial Issues

- Changes to tables are not easy to spot, especially added text.
 - Note that the format used follows the example from the Style Guide, where the entire table is copied and changes are identified through underline (additions) and strikethrough (deletions).
 - To facilitate revision, the changes are temporality highlighted in yellow
 - Is there a better way to describe changes in the Corrigendum? Ex. indicate a specific row in the table.

	Name	Refer	Selector	Field(s)
key	componentKey2		./hc:Component	@ID
keyref	adjustComponentRef	td:componentKey2	./td:AdjustComponent	@componentID
keyref	componentFaultRef	td:componentKey2	./td:Fault	@componentID
keyref	<u>componentFaultRef2</u>	<u>td:componentKey2</u>	<u>./uut:Fault</u>	<u>@componentID</u>

Course of Action

- The Corrigendum document and revised XML schemas prepared for this meeting contain all the changes enumerated in this presentation. Some of these changes are not in scope.
- Two possible courses of action:
 - 1. Retain for the Corrigendum only the changes that are in scope. Address the remaining changes in the upcoming revision.
 - 2. retire the current PAR and request a new PAR with extended scope.
- The summary table in the next slide shows the impact of deferring each of the changes.

Impact of Changes

(1) Means “substantive change”. Does not include changes to Annex A, made solely to reflect schema changes.

Issue No.	Schema Change	Example Change	Std. Doc. Change (1)	In Scope	Breaking	Notes	Impact, if omitted from Corrigendum
1	X			Y	N	Add xs:key to enforce uniqueness. Brings schema in sync with standard text.	Medium. Instance documents with ambiguous references could pass validation, but they can/should be detected programmatically by consumers.
2	X			Y	N	Add xs:key to enforce uniqueness. Brings schema in sync with standard text.	Medium. Same as above.
3.1	X			Y	N	Correct existing xs:keyref. Brings schema in sync with standard text.	Medium. Same as above.
3.2	X		X	N	N (?)	Add xs:key to improve validation.	Medium. Same as above.
4	X		X	N	N	Add attribute to remove restriction on modeling capabilities.	Medium. Limits usability of the standard in a particular use case of operation AttributeChange.
5	X			Y	N	Correct annotation. Brings schema in sync with standard text.	Low. Annotation could be confusing but cannot produce incorrect validation or incorrect instance documents.
6	X		X	N	N	Add element to improve modeling capabilities.	Low. Prevents the documentation of TestResultDescriptions in TestGroups.
7		X	X	N	N (?)	Deprecate unnecessary element	Medium. Instance documents that contain the element being deprecated could pass validation, but they can/should be detected programmatically by consumers because they are logically incorrect.
7a		X		N	N	Change attribute format in example, for readability.	None. Just an improvement of an example.
8		X		N	N	Change example to avoid ambiguous interpretation.	Low. Ambiguity can lead to incompatible implementations. However, the context is very localized (OperationRunTest)
9		X		N	N	Correct example.	None. Just a correction to an example.
10				N/A	N/A	No change in Corrigendum	Low. Instance documents from third-party producers could be incompatible with consumers in code generation solutions. However, a workaround can be implemented in consumers.
11			X	N	N (?)	Correct typo in key, in standard document. Schema is correct.	Low. Schema is correct and the standard text is obviously incorrect, so there is no ambiguity.