IEEE Standard for Common Format for Event Data Exchange (COMFEDE) for Power Systems

Sponsor
IEEE Power System Relaying Committee
of the
IEEE Power and Energy Society

Approved 30 September 2010
IEEE-SA Standards Board
Abstract: A common format for data files used for the interchange of various types of event data collected from electrical power systems or power system models is defined. Extensibility, extension mechanisms, and compatibility of future versions of the format are discussed. An XML schema is defined. A sample file is given.

Keywords: eXtensible Markup Language (XML) name spaces, instance file, payload data, XML schema
IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied “AS IS.”

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation, or every ten years for stabilization. When a document is more than five years old and has not been reaffirmed, or more than ten years old and has not been stabilized, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon his or her independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Recommendations to change the status of a stabilized standard should include a rationale as to why a revision or withdrawal is required. Comments and recommendations on standards, and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Authorization to photocopy portions of any individual standard for internal or personal use is granted by The Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.
Introduction

This introduction is not part of IEEE Std C37.239-2010, IEEE Standard for Common Format for Event Data Exchange (COMFEDE) for Power Systems.

Power network fault event data are indispensable to the analysis, testing, evaluation, and simulation of power systems and related protection schemes. The flexibility provided by digital devices in recording such event data has brought about a situation wherein the users of these records are confronted with the difficulty of dealing with the different formats used by each device to generate, store, and transmit the recorded data.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Standards Association web site at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA web site at http://standards.ieee.org.
Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/reading/ieee/updates/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

Attention is called to the possibility that implementation of standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.
Participants

At the time this standard was submitted to the IEEE-SA Standards Board for approval, the Common Data Format for IED Event Data Working Group had the following membership:

Mark Adamiak, Chair
Pierre Martin, Vice Chair

Eric Allen       Jim Hacket       Stanley Klein
Scott Anderson   Randy Hamilton   Larry Smith
Alex Apostolov   Juergen Holbach  Stan Thompson
Christoph Brunner Mladen Kezunovic Benton Vandiver

The following members of the [individual/entity] balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

William J. Ackerman       C. Huntley         Sukhbir Sachdev
Mark Adamiak              Akio Iso            Bob Saint
S. Aggarwal               R. Jackson         Miriam Sanders
James Ariza               Junghoon Jee      Steven Sano
Ali Al Awazi              Piotr Karocki     Bartien Sayogo
Hugh Barrass              Tanuj Khandelwal  Thomas Schossig
Martin Baur               Yuri Khersonsky    Lubomir Sevov
Kenneth Behrendt          Yongburn Kim      Devki Sharma
Oscar Bolado              James Kinney       Gil Shultz
Steven Brockschink        Stanley Klein     Mark Simon
Gustavo Brunello          J. Koeppfinger    Vesselin Skendzic
Suresh Channarasappa     Chung-Yiu Lam     James Smith
Arvind K. Chaudhary       Charles Lennon    Jerry Smith
Keith Chow                James Liming      John Spare
Stephen Conrad            G. Luri           Gary Stedter
R. W. Corlew             Pierre L. Martin   Michael Swearingen
James Cornelison          William McBride   Richard Taylor
Carlo Donati              John McDonald     John Tengdin
Gary Donner               Gary Michel       John Toth
Michael Dood              Keith Moore       Demetrios Tziouvaras
Neal Dowling              Jerry Murphy       Joe Uchiyama
Gary Engmann              R. Murphy        Eric Udren
Kenneth Fodero            Bruce Muschitz    Sterling Vaden
Jeffrey Gilbert           Anthony Napikoski  Srinivasa Vemuru
Manjinder Gill            Michael S. Newman  John Vergis
Jalal Gohari              Michael S. Newman  Jane Verner
Edwin Goodwin             Robert Orndorff    John Wang
Randall Groves            Chris Osterloh    Hung-Yu Wei
Roeer Hedding             Lorraine Padden   Kenneth White
Gary Heuston              Bansi Patel       Philip Winston
Werner Hoelzl              Farnoosh Rahmatian Paul Work
Stuart Holoman            R. Ray            Richard Young
David Horvath             Robert Robinson    Oren Yuen

Copyright © 2010 IEEE. All rights reserved.
When the IEEE-SA Standards Board approved this standard on 30 September 2010, it had the following membership:

Robert M. Grow, Chair  
Richard H. Hulett, Vice Chair  
Steve M. Mills, Past President  
Judith Gorman, Secretary

Karen Bartleson  
Victor Berman  
Ted Burse  
Clint Chaplin  
Andy Drozd  
Alexander Gelman  
Jim Hughes  
Young Kyun Kim  
Joseph L. Koepfinger*  
John Kulick  
David J. Law  
Hung Ling  
Oleg Logvinov  
Ted Olsen  
Ronald C. Petersen  
Thomas Prevost  
Jon Walter Rosdahl  
Sam Sciacca  
Mike Seavey  
Curtis Siller  
Don Wright

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish Aggarwal, NRC Representative  
Richard DeBlasio, DOE Representative  
Michael Janezic, NIST Representative

Don Messina  
IEEE Standards Program Manager, Document Development

Matthew J. Ceglia  
IEEE Standards Program Manager, Technical Program Development
Contents

1. Overview .................................................................................................................................................... 1  
   1.1 Scope ................................................................................................................................................... 1  
   1.2 Purpose ................................................................................................................................................ 1  

2. Normative references ............................................................................................................................. 2  

3. Definitions, acronyms, and abbreviations ............................................................................................. 2  
   3.1 Definitions ........................................................................................................................................... 2  
   3.2 Acronyms and abbreviations ............................................................................................................... 3  

4. Conformance .............................................................................................................................................. 3  

5. The COMFEDE format specification ........................................................................................................ 4  
   5.1 General .................................................................................................................................................. 4  
   5.2 Name spaces ....................................................................................................................................... 4  
   5.3 Specification method ........................................................................................................................... 5  
   5.4 Extensibility ....................................................................................................................................... 7  
   5.5 COMFEDE versions and compatibility ............................................................................................. 8  
   5.6 Main structure of the schema .......................................................................................................... 10  
   5.7 The COMFEDE and Log sections .................................................................................................... 12  
   5.8 EntryData .......................................................................................................................................... 17  
   5.9 PayloadData ..................................................................................................................................... 23  

Annex A (normative) The COMFEDE XML schema .................................................................................... 41  

Annex B (informative) COMFEDE instance file example ........................................................................... 60  

Annex C (informative) Bibliography ........................................................................................................... 63
IEEE Standard for Common Format for Event Data Exchange (COMFEDE) for Power Systems

IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection in all circumstances. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading “Important Notice” or “Important Notices and Disclaimers Concerning IEEE Documents.” They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

1. Overview

1.1 Scope

This standard defines a format for files containing event data such as sequence of events or fault summary reports collected from power systems or power system models. The format is intended to provide an easily interpretable form for use in exchanging data.

1.2 Purpose

This standard defines a common format for the data files needed for the exchange of various types of power network events in order to facilitate event data integration and analysis from multiple data sources and from different vendor devices. The flexibility provided by digital devices in recording network fault event data in the electric utility industry has generated the need for a standard format for the exchange of data. These data are being used with various devices to enhance and automate the analysis, testing, evaluation, and simulation of power systems and related protection schemes during fault and disturbance conditions. Since each source of data may use a different proprietary format, a common data format is necessary to facilitate the exchange of such data between applications. This will facilitate the use of proprietary data in diverse applications and allow users of one proprietary system to use digital data from other systems.
2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.


3. Definitions, acronyms, and abbreviations

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary: Glossary of Terms & Definitions* should be referenced for terms not defined in this clause.5

3.1 Definitions

**breaker operate time**: The time from energization of the trip coil until the current in the phases is interrupted.

**event**: An event is defined as either a change of state of a binary data value, exceeding or resetting of a measured threshold, or meeting or resetting a rate of change threshold.

**EntryData**: Basic information associated with an event: Data Reference, Value, Timestamp, Quality, and Reason Code.

**eXtensible Markup Language (XML) name space**: A mechanism used to avoid clashes between names from different markup vocabularies.

**fault clearing time**: The time from the fault inception until all currents are interrupted in the faulted phases.

**fault summary reports**: The type of recording containing a sequence of event history and extra information associated with each event.

**fault trigger**: Signals or internal logic that triggers an event.

**instance file**: File containing information that conforms to a specific format.

---

1 World Wide Web (W3) Consortium publications are available from the World Wide Web Consortium, Massachusetts Institute of Technology, 32 Vassar Street, Room 32-G515, Cambridge, MA 02139 (http://www.w3.org/).
4 http://www.w3.org/TR/2004/REC-xml-names11-20040204/.
**International Electrotechnical Commission (IEC) 61850 Log:** Periodic recordings, such as metering, or a sequence of event data, that conform to IEC 61850-7-2 Log standards.

**Payload Data:** Extra information associated with an event. For example, for an event recorded during a fault, payload data may include information such as distance to fault, fault currents, fault voltages, and so on.

**primary values:** The magnitude of a particular quantity, expressed as a number multiplied by a unit of measurement whose value corresponds to the unit used on the primary side of a power system.

**receiver:** An agent that is a recipient of the data to be processed.

**sender:** An agent that supplies the data for the intent of conveying information.

**sequence of events (SOE) report:** A recording of sequence of events (SOE) data.

**trigger options (TrgOpt):** The source of a trigger. Options include Data Change, Data Update, General Interrogation, and Integrity (periodic) update.

**Unified Modeling Language (UML) class diagram:** A visual expression of various static relations of class-related elements. Class Diagram can contain not only classes but also interfaces, enumerations, packages, various relations, instances, and their links.

### 3.2 Acronyms and abbreviations

- **ac** alternating current
- **COMFEDE** Common Format for Event Data Exchange
- **dc** direct current
- **IED**
- **UML** Unified Modeling Language
- **URI** Uniform Resource Identifier
- **UTC** Coordinated Universal Time
- **XML** eXtensible Markup Language

### 4. Conformance


Conformance to this standard refers to instance documents that shall be conformant to a schema document. The schema definition is contained within a set of files that exist separately from this document.

Clause 5 specifies the syntax definition of COMFEDE using Unified Modeling Language (UML) class diagrams, text, and excerpts from the XML schema files. In case of inconsistencies between this document and the XML schema files, the latter are those to which the reader is referred.

This standard also contains constraints that should be respected to maximize the interoperability between senders and receivers of COMFEDE files having to exchange in the future different versions of the format.

The extension file name of a COMFEDE instance document shall be CED.

---

6 Information on references can be found in Clause 2.
5. The COMFEDE format specification

5.1 General

The reader should have a general understanding of XML schemas or XML terminology.

COMFEDE has the following main characteristics:

a) It consists of a set of XML schema files identified with the extension XSD.

b) The main schema name space is www.pes-psrc.org/Subcommittee/H/COMFEDE.

c) The format is able to, at least, hold the information related to:
   1) Sequence of events (SOE) reports.
   2) Fault summary reports.
   3) IEC 61850 Logs.

d) The format includes an extension mechanism allowing the format to be extended in a formal way.

Information from these reports can be stored natively in the COMFEDE format or be translated via a tool into COMFEDE format.

5.2 Name spaces

A schema is a collection of definitions whose names belong to a particular name space. Name spaces are used to distinguish between definitions from different vocabularies. The conformance of an XML instance document can be checked with regard to the definitions found in a schema by using a name space declaration.

If many vocabularies in the same domain share common definitions, then it is good practice to split the definitions in smaller entities, each of them having its own schema and possibly its own name space. Figure 1 shows how COMFEDE uses this technique.
Although the two schemas “Common Format for Event Data” and “Other Standard” have their own definitions (name spaces), the definitions specified in the “Common Definitions” schema must also be imported. The “Common Definitions” schema includes definitions coming from a set of schema files, each of them sharing the same name space. These relations can be formally declared in an XML schema using the elements xs:import and xs:include.

The “Common Definitions” schema also reuses definitions conformant with IEC 61850-6 (Ed. 1) [B1]. However, this reuse is not formally declared in the schema.

The main name space for the COMFEDE shall be http://www.pes-psrc.org/Subcommittee/H/COMFEDE. This name space is reserved by this standard and shall not be used to represent any other data element (see W3C Recommendation, Name spaces in XML 1.1). This name space shall not be used to define extensions to a COMFEDE XML instance.

5.3 Specification method

This standard specifies the syntax definition of COMFEDE as a W3C XML schema.

Within the schema, the following conventions are used:

a) Schema type names start with the lowercase letter t (for example, tLog).

b) Enumeration names start with the lowercase letter t and end with the letters Enum (for example, tEventTypeEnum).

c) Element names start with an uppercase letter (for example, Location).
d) Attribute group definitions start with the acronym ag (for example, agQuality). An attribute group is used to define a reusable group of attributes.
e) Attribute names start with a lowercase letter (for example, name).

The schema has been segmented in different files containing type definitions as shown in Table 1.

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMFEDE.xsd</td>
<td>The main schema. Name space schema attribute declaration = xmlns:CED.</td>
</tr>
<tr>
<td>H_CommonTypes.xsd</td>
<td>The common complex types definitions used by the main schema. Name space schema attribute declaration in the main schema = xmlns:sch.</td>
</tr>
<tr>
<td>H_SimpleTypes.xsd</td>
<td>The common simple types definitions used by the main schema and included in the schema H_CommonTypes.xsd.</td>
</tr>
<tr>
<td>HEnums.xsd</td>
<td>The enumerations used by the main schema and included in the schema H_CommonTypes.xsd.</td>
</tr>
</tbody>
</table>

5.3.1 tAnyName, tName, and tNaming types

tAnyName and tName are simpleType elements (see schema H_SimpleTypes.xsd) reused in COMFEDE where naming by means of a character string datatype is required. tAnyName is an XML normalizedString type, which means that it does not contain any carriage return, line feed, or tab character. tName is of type tAnyName with the additional constraint that it cannot be empty. tNaming is used where an element requires a name attribute and an optional description attribute.

```xml
<xs:simpleType name="tAnyName">
    <xs:restriction base="xs:normalizedString"/>
</xs:simpleType>

<xs:simpleType name="tName">
    <xs:restriction base="tAnyName">
        <xs:minLength value="1"/>
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="tNaming">
    <xs:annotation>
        <xs:documentation>This type is reused where an element requires a name attribute and an optional description attribute.</xs:documentation>
    </xs:annotation>
    <xs:attribute name="name" type="tName" use="required"/>
    <xs:attribute name="desc" type="tAnyName"/>
</xs:complexType>
```

5.3.2 xs:dateTime and xs:time types

xs:dateTime and xs:time are datatypes defined in the W3C XML recommendation and are used in COMFEDE where information regarding date and time is required (e.g., the timestamp information). These datatypes were inspired by the standard ISO 8601 representations of dates and times. Among others, the Coordinated Universal Time (UTC) time specification may be represented by adding a Z to the end of the date/time value.
5.4 Extensibility

5.4.1 General

Extensibility is an important property of any data format. It allows the format to stay simple while enabling third-party solutions to be provided thanks to optional extensions. Extensibility also allows for the adding of more value to the data being exchanged. As it is not possible to anticipate how the data in a format may be augmented, it is good practice not to restrict which elements can be extended.

Thus, the format defined in this standard makes use of an extension mechanism that maximizes the interoperability between senders and receivers of COMFEDE files having to exchange relevant data that is not included in the standard.


5.4.2 Extension mechanism

5.4.2.1 tBaseElementForExtension

The type tBaseElementForExtension is the main mechanism used for extending the COMFEDE schema.

```
<xs:complexType name="tBaseElementForExtension" abstract="true">
    <xs:sequence>
        <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="Text" type="tText" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

tBaseElementForExtension uses the wildcard schema component xs:any namespace="##other" and xs:anyAttribute namespace="##other". This means that everywhere this type is declared in COMFEDE, any implementation of COMFEDE can add any new element or attribute definitions without requiring the schema be changed. Most of the elements defined in COMFEDE are based on the type tBaseElementForExtension as shown in the UML diagram in Figure 2.

tBaseElementForExtension also contains the optional element Text, which allows for adding content of any mixture of text and elements and attributes from other name spaces than the COMFEDE name space. The attribute source may be used to link to a Uniform Resource Identifier (URI) for large text insertions.

```
<xs:complexType name="tText" mixed="true">
    <xs:complexContent mixed="true">
        <xs:extension base="tAnyContentFromOtherNamespace">
            <xs:attribute name="source" type="xs:anyURI" use="optional"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```
The use of the type tBaseElementForExtension constrains extensions to belong to a defined name space with defined semantics for all extensions. A receiver can, thus, use the extended file for the following reasons:

a) To use the standardized portions of the data as they are intended to be used
b) To ignore the extended portions of the data
c) To use the extended portions of the data if the extended name space is known and available (e.g., in another XML schema)

The extension name spaces shall be defined at the main element Log level. The name space names shall not be the same as the names used for COMFEDE.

Figure 2 —The type tBaseElementForExtension

5.4.3 Extension example

An example of an extension to COMFEDE is given in Annex B. The name space used for the extension is www.company.com/ComfedeExtension.

5.5 COMFEDE versions and compatibility

5.5.1 General

With regard to COMFEDE, compatibility is the property of the data format allowing receivers to read different versions of the format without requiring upgrades of the receivers. Backward compatibility allows
receivers designed for a new version of COMFEDE to read older versions. Forward compatibility allows receivers designed for older versions of COMFEDE to read a new version.

There will inevitably be new versions of COMFEDE once this standard has been approved. These versions may be required for enhancement of the format (i.e., new features added) or for error corrections. Errors shall be fixed even if the modifications endanger compatibility and receivers shall be adapted accordingly. As for enhancements, the following rules shall be followed by creators of new versions:

a) The COMFEDE name space shall not be changed.
b) New attributes shall be added as options. The attribute default value, if needed, shall be the same as the value in the absence of the attribute in older versions.
c) Changing the default values of existing attributes shall not be allowed.
d) New elements shall be added at the end of existing type definitions.
e) Existing elements and attributes that are no longer required may be removed but shall still be allowed in older versions. However, their use should be discouraged.
f) Changing the semantics of existing elements and attributes shall not be allowed. New elements and attributes shall be added instead and the use of the older ones should be discouraged.


5.5.2 COMFEDE versions

As an eXtensible Markup Language (XML) schema, COMFEDE contains an attribute at the schema Element Information Item level to indicate which version of the schema is used.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:CED="http://www.pes-psrc.org/Subcommittee/H/COMFEDE"
    xmlns:sch="http://www.pes-psrc.org/Subcommittee/H/CommonTypes"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    targetNamespace="http://www.pes-psrc.org/Subcommittee/H/COMFEDE"
    elementFormDefault="qualified" attributeFormDefault="unqualified"
    version="1.0">
    ...
</xs:schema>
```

The W3C specification defines no semantics for the attribute version. In the COMFEDE context, the schema attribute version is the IEEE standard released version of COMFEDE, which is 1.0 for this standard.

However, to release intermediate versions, especially in the context of this standard, the attributes version and revision have been introduced at the root element Log.

```
<xs:element name="Log">
    <xs:complexType>
        <xs:complexContent>
            <xs:extension base="CED:tLog">
                <xs:attribute name="version" type="CED:tCedVersion" use="required" fixed="2010"/>
                <xs:attribute name="revision" type="CED:tCedRevision" use="required" fixed="A"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
</xs:element>
```
For this standard, version is fixed to 2010 and revision is fixed to A.

### 5.5.3 Versioning strategy

This subclause describes a processing model needed for handling new versions of COMFEDE in receivers. It also specifies rules to be used by the creators of future versions.

The Must Ignore Rules and MustUnderstand Constructs defined hereinafter should be used to allow receivers to read different versions of COMFEDE instance files with the same name space.

The Must Ignore All Rule is a substitution model that enables forwards-compatible changes by ignoring content that is not understood (World Wide Web Consortium [B7]). For COMFEDE, this means that the receiver is expected to ignore elements that are not understood and all their content as well. For attributes, only the attribute not understood is ignored.

For cases where a new version contains data that shall be understood by the receiver, the MustUnderstand Construct shall be used. This means that new declared elements shall contain the mustUnderstand attribute with a value equal to True. Receivers then know they cannot ignore the new element and related applications shall abort the processing of the instance file.

### 5.6 Main structure of the schema

This subclause specifies the syntax definition of COMFEDE using UML class diagrams, text, and excerpts from the XML schema files.

NOTE—Some optional element attributes in the schema have a default value, which means that in the absence of the attribute, a receiver shall use this value if needed. To avoid dealing with unnecessarily large files, it is recommended that those attributes be not included in an instance file if default values are used. This recommendation is in accordance with rule c) in 5.5.1.8

A UML class diagram of the main structure of COMFEDE is shown in Figure 3. The main element COMFEDE may contain one or more elements Log. The element Log is of type tLog and represents a record that comes from a location defined by the element type tLocation. In addition to the location information (station name, company name, etc.), tLocation identifies the IED that has made the record (element type tIEDNameplate) and the associated voltage level. The SettingFileName element contains the name(s) of the setting file(s) in force in the device when the report was extracted. As a result, a COMFEDE instance file may contain records coming from one particular IED, from different IEDs in the same substation, or from different IEDs in different substations.

---

8 Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.
Figure 3—Main structure of the schema

Each log is a sequence of events that contains entries (tEntry). Each Entry contains one or more EntryData elements (tEntryData). Each EntryData element has the attribute groups agTimeStamp and agQuality. The attribute group agTimeStamp refers to the time when the event occurred and the time quality code, whereas the attribute group agQuality holds information about the quality of the value (tVal). Each EntryData element also has the optional attributes faultTrigger (true or false) and event type (enumeration tEventTypeEnum). Each EntryData element contains a mandatory data name (tDataRef), an optional value
(tVal), or one-to-many optional data attribute values (tDA). In addition, it may optionally contain the reason for inclusion of the data into the report (ReasonCode) and extra information associated with the event (tPayloadData).

5.7 The COMFEDE and Log sections

The COMDEFE and Log elements are global elements. This means among other things that either element can be declared at the root level of an instance file, but not both in the same instance file.

A COMFEDE element can contain an unlimited number of Log elements. This is done by using the attribute Ref, which references the global element Log.

```
<xs:element name="COMFEDE">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="CED:Log" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

The element Log is of type tLog.

```
<xs:element name="Log">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="CED:tLog">
        <xs:attribute name="version" type="CED:tCedVersion" use="required" fixed="2010"/>
        <xs:attribute name="revision" type="CED:tCedRevision" use="required" fixed="A"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

The element Log also has two mandatory attributes as defined in Table 2 (the column M/O stands for Mandatory/Optional).

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Version number of the format. Fixed to 2010 for this version.</td>
<td>M</td>
</tr>
<tr>
<td>Revision</td>
<td>Revision number of the format. Fixed to A for this revision.</td>
<td>M</td>
</tr>
</tbody>
</table>

The version number corresponds to the year of publication of the format. The revision number identifies intermediate revisions between official publications, if required, to solve possible technical issues. The relationship between these attributes and the version attribute at the schema Element Information Item level is explained in 5.5.2.

The element Log is of type tLog, which contains three elements: one mandatory Location element and two optional elements, Entry and SettingFileName.

The type tLog section is as follows:

```
<xs:complexType name="tLog">
  <xs:complexContent>
    <xs:restriction base="CED:tLog">
      <xs:attribute name="version" type="CED:tCedVersion" use="required" fixed="2010"/>
      <xs:attribute name="revision" type="CED:tCedRevision" use="required" fixed="A"/>
    </xs:restriction>
  </xs:complexContent>
</xs:element>
```
The element type tLog has attributes as defined in Table 3.

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogName</td>
<td>Is the name of the Log. For IEC 61850 devices, it shall unambiguously identify a Log within the scope of a Logical Node.</td>
<td>O</td>
</tr>
<tr>
<td>LogRef</td>
<td>For IEC 61850 devices, it shall be the unique path-name of a Log.</td>
<td>O</td>
</tr>
<tr>
<td>OldEntrTm</td>
<td>Indicates the TimeOfEntry when the oldest log entry has been stored</td>
<td>O</td>
</tr>
<tr>
<td>NewEntrTm</td>
<td>Indicates the TimeOfEntry when the newest log entry has been stored</td>
<td>O</td>
</tr>
<tr>
<td>OldEntr</td>
<td>Indicates the EntryID for the oldest entry available in the log.</td>
<td>O</td>
</tr>
<tr>
<td>NewEntr</td>
<td>Indicates the EntryID for the newest entry available in the log.</td>
<td>O</td>
</tr>
</tbody>
</table>

In IEC 61850-7-2 (Ed. 1): 2003 [B2], entryID is specified as a counter that rolls over when the maximal value has been reached, whereas timeOfEntry is the time when the log entry is added to a LOG (see 5.7.2). EntryID together with timeOfEntry provides a unique identification of the entry.

### 5.7.1 Location

The type tLocation contains the information related to where comes the record (i.e., from which station, from which company, and from which IED) and to the local time of the record. It contains one mandatory element IEDNameplate identifying the IED that made the record.

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEDNameplate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>companyIdentificationCode</td>
<td></td>
<td>default=00</td>
</tr>
<tr>
<td>operatorName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stationName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>voltageLevelInkV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bayName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>latitude</td>
<td></td>
<td></td>
</tr>
<tr>
<td>longitude</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The type `tLocation` has attributes as defined in Table 4.

**Table 4—Attributes of the element type `tLocation`**

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>companyName</td>
<td>The name of the company that owns the station.</td>
<td>M</td>
</tr>
<tr>
<td>companyIdentificationCode</td>
<td>The identification code of the company.</td>
<td>O</td>
</tr>
<tr>
<td>operatorName</td>
<td>The name of the company that operates the station.</td>
<td>O</td>
</tr>
<tr>
<td>stationName</td>
<td>The name of the station.</td>
<td>M</td>
</tr>
<tr>
<td>voltageLevelInkV</td>
<td>Voltage level in kV with which the recording device is functionally associated.</td>
<td>O</td>
</tr>
<tr>
<td>bayName</td>
<td>The name of the bay.</td>
<td>O</td>
</tr>
<tr>
<td>Latitude</td>
<td>Geographical position of the device in WGS84 coordinates—latitude.</td>
<td>O</td>
</tr>
<tr>
<td>Longitude</td>
<td>Geographical position of the device in WGS84 coordinates—longitude.</td>
<td>O</td>
</tr>
<tr>
<td>tmOfsTmm</td>
<td>Offset of local time from UTC in minutes.</td>
<td>O</td>
</tr>
</tbody>
</table>

The companyIdentificationCode is the North American Electric Reliability Corporation unique utility identifier. It is of type `tCompanyIdentificationCode`, which is an XML normalizedString that does not contain any carriage return, line feed, or tab character; cannot be empty; and shall be limited to two to four characters. This is an optional attribute. If it is not present, the default value is assumed to be unknown.

The IEDNameplate element is of type `tIEDNameplate` and has attributes as defined hereafter and in Table 5.
Table 5—Attributes of the element type tIEDNameplate

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the IED that produced the COMFEDE file.</td>
<td>M</td>
</tr>
<tr>
<td>vendor</td>
<td>The name of the manufacturer of the IED.</td>
<td>M</td>
</tr>
<tr>
<td>model</td>
<td>The IED model.</td>
<td>O</td>
</tr>
<tr>
<td>serNum</td>
<td>The IED serial number.</td>
<td>O</td>
</tr>
<tr>
<td>hwRev</td>
<td>The IED hardware revision.</td>
<td>O</td>
</tr>
<tr>
<td>swRev</td>
<td>The IED firmware revision.</td>
<td>O</td>
</tr>
</tbody>
</table>

5.7.2 Entry

The Entry element identifies each entry in the Log and is of type tEntry.

```xml
<xs:complexType name="tEntry">
  <xs:complexContent>
    <xs:extension base="CED:tBaseElementForExtension">
      <xs:sequence>
        <xs:element name="EntryData" type="CED:tEntryData" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="entryId" type="xs:unsignedLong">
        <xs:annotation>
          <xs:documentation>Sequence number.</xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="timeOfEntry" type="xs:dateTime">
        <xs:annotation>
          <xs:documentation>Time when the Entry is added to the buffer</xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Each element Entry shall contain at least one EntryData element. Table 6 shows the two optional attributes of the element Entry.

Table 6—Attributes of the element type tEntry

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>entryId</td>
<td>Is a unique reference to all logged entries (elements EntryData) having the same value of TimeOfEntry.</td>
<td>O</td>
</tr>
<tr>
<td>timeOfEntry</td>
<td>Is the time when the logged entry (element EntryData) is added to a LOG. That time may be different to the time stamp of the data (element Timestamp), which is the time when the event occurred that caused the logged entry to be created.</td>
<td>O</td>
</tr>
</tbody>
</table>

5.7.3 SettingFileName

The SettingFileName element is of type tFileName. Each SettingFileName element identifies the name of a setting file in force in the device when the report was extracted. The type tFileName has one mandatory attribute extensionName corresponding to the name of the file extension.

```xml
<xs:complexType name="tFileName">
  <xs:simpleContent>
    <xs:extension base="tName">
      <xs:attribute name="extensionName" type="tFileExtensionEnum" use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```
Allowed file extension names are defined by the enumeration tFileExtensionEnum.

```
<xs:simpleType name="tFileExtensionEnum">
  <xs:union memberTypes="ns:tStandardizedFileExtensionEnum
  ns:tNonStandardizedFileExtensionEnum"/>
</xs:simpleType>
```

The content of the type tFileExtensionEnum is the union of the enumerations tStandardizedFileExtensionEnum and tNonStandardizedFileExtensionEnum.

tStandardizedFileExtensionEnum is the enumeration of the allowed standardized extension names.

```
<xs:simpleType name="tStandardizedFileExtensionEnum">
  <xs:restriction base="xs:Name">
    <xs:enumeration value="cid">
      <xs:annotation>
        <xs:documentation>Configured IED Description file as defined in IEC 61850-6</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="scd">
      <xs:annotation>
        <xs:documentation>System Configuration Description file as defined in IEC 61850-6</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="ssd">
      <xs:annotation>
        <xs:documentation>Substation Specification Description file as defined in IEC 61850-6</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="cfg">
      <xs:annotation>
        <xs:documentation>COMTRADE Configuration file as defined in IEEE C37.111</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="dat">
      <xs:annotation>
        <xs:documentation>COMTRADE Data file as defined in IEEE C37.111</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="inf">
      <xs:annotation>
        <xs:documentation>COMTRADE Information file as defined in IEEE C37.111</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="hdr">
      <xs:annotation>
      </xs:annotation>
  </xs:restriction>
</xs:simpleType>
```
tNonStandardizedFileExtensionEnum is an enumeration allowing any name with the following restrictions:

- It shall at least have three characters.
- The allowed characters are any mixture of letters and decimals.

```xml
<xs:simpleType name="tNonStandardizedFileExtensionEnum">
    <xs:restriction base="xs:Name">
        <xs:minLength value="3"/>
        <xs:pattern value="[A-Z,a-z,0-9]*"/>
    </xs:restriction>
</xs:simpleType>
```

### 5.8 EntryData

The EntryData element is of type tEntryData and contains two mandatory elements (DataRef and either Val or DA) and two optional elements (ReasonCode and PayloadData).

The type tEntryData is as follows:

```xml
<xs:complexType name="tEntryData">
    <xs:complexContent>
        <xs:extension base="CED:tBaseElementForExtension">
            <xs:sequence>
                <xs:element name="DataRef" type="CED:tDataRef"/>
                <xs:choice>
                    <xs:element name="Val" type="CED:tVal"/>
                    <xs:element name="DA" type="CED:tDA" maxOccurs="unbounded"/>
                </xs:choice>
                <xs:element name="ReasonCode" type="CED:tTrgOpt" minOccurs="0"/>
                <xs:element name="PayloadData" type="CED:tPayloadData" minOccurs="0"/>
            </xs:sequence>
            <xs:attributeGroup ref="sch:agTimeStamp"/>
            <xs:attributeGroup ref="sch:agQuality"/>
            <xs:attribute name="faultTrigger" type="xs:boolean" default="false"/>
            <xs:attribute name="eventType" type="sch:tEventTypeEnum"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

tEntryData has two attribute groups: agTimeStamp and agQuality. An attribute group is a named group of attribute declarations so that they may be incorporated as a group into complex type definitions. The attribute groups agTimeStamp and agQuality are described in 5.8.1 through 5.8.2.
tEntryData has also two optional attributes, faultTrigger and eventType, as shown in Table 7.

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>faultTrigger</td>
<td>Boolean to indicate if the event was triggered by a fault trigger.</td>
<td>O</td>
</tr>
<tr>
<td>evenType</td>
<td>Enumeration of types of events that may be used to filter events based on those types.</td>
<td>O</td>
</tr>
</tbody>
</table>

The attribute eventType is an enumeration of type tEventTypeEnum whose values are as follows:

```
<xs:simpleType name="tEventTypeEnum">
  <xs:restriction base="xs:Name">
    <xs:enumeration value="SysEvt">
      <!-- Power system event (fault detected; trip issued; breaker operation ...) -->
    </xs:enumeration>
    <xs:enumeration value="ComEvt">
      <!-- Communication event (association lost, buffered report overflow; ...) -->
    </xs:enumeration>
    <xs:enumeration value="SetChg">
      <!-- Setting Change Event (setting changed/edited, new setting group; adaptative setting change; ...) -->
    </xs:enumeration>
    <xs:enumeration value="IEDFail">
      <!-- IED Failures: A/D out of calibration; component failure; memory corrupt; ...) -->
    </xs:enumeration>
    <xs:enumeration value="IEDEvt">
      <!-- Internal IED events (event driven by logic state changes, oscillography triggered; ...) -->
    </xs:enumeration>
    <xs:enumeration value="TestEvt">
      <!-- Test events (IED in test mode, protection in test mode; ...) -->
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
```

5.8.1 The attribute group agTimeStamp

The attribute group agTimeStamp consists of the attribute t and the attribute group agTimeQuality.

```
<xs:attributeGroup name="agTimeStamp">
  <xs:attribute name="t" type="xs:dateTime" use="required"/>
  <xs:attributeGroup ref="#agTimeQuality"/>
</xs:attributeGroup>
```

The attribute t represents the time when the event occurred. Its type is the XML dateTime datatype. Note that, without the use of any further XML constraint, the attribute t allows a fractional second resolution well over the nanosecond.

The attribute group agTimeQuality provides information about the quality of the time source of the recording device when the event occurred. The attribute group has two optional attributes: timeQualityCode and leapSec, which are described in Table 8.

```
<xs:attributeGroup name="agTimeQuality">
  <xs:attribute name="timeQualityCode" use="optional" default="clockLocked"/>
</xs:attributeGroup>
```
Table 8 —Attributes of the attribute group agTimeQuality

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeQualityCode</td>
<td>It is the Time Quality indicator code of the recording device’s clock. It is an indication of synchronization relative to one source. It corresponds to the Time Quality indicator code as defined in IEEE Std C37.118™-2005 [B5]. The default value is “clockLocked”.</td>
<td>O</td>
</tr>
<tr>
<td>leapSec</td>
<td>It is the leap second indicator. It indicates that a leap second may have been added or deleted during the recording resulting either in two pieces of data having the same Second of Century time stamp or a missing second. The default value is “noLeapSec”.</td>
<td>O</td>
</tr>
</tbody>
</table>

The attributes timeQualityCode and leapSec are enumerations:

```xml
<xs:simpleType name="tTimeQualityCodeEnum">
  <xs:restriction base="xs:Name">
    <xs:enumeration value="clockLocked"/>
    <xs:enumeration value="clockUnlockedWithinOneNanoSec"/>
    <xs:enumeration value="clockUnlockedWithin10NanoSec"/>
    <xs:enumeration value="clockUnlockedWithin100NanoSec"/>
    <xs:enumeration value="clockUnlockedWithinOneMicroSec"/>
    <xs:enumeration value="clockUnlockedWithin10MicroSec"/>
    <xs:enumeration value="clockUnlockedWithin100MicroSec"/>
    <xs:enumeration value="clockUnlockedWithinOneMilliSec"/>
    <xs:enumeration value="clockUnlockedWithin10MilliSec"/>
    <xs:enumeration value="clockUnlockedWithin100MilliSec"/>
    <xs:enumeration value="clockUnlockedWithinOneSec"/>
    <xs:enumeration value="clockUnlockedWithin10Sec"/>
    <xs:enumeration value="clockfailure"/>
  </xs:restriction>
</xs:simpleType>
```

```xml
<xs:simpleType name="tLeapSecEnum">
  <xs:restriction base="xs:Name">
    <xs:enumeration value="noLeapSec"/>
    <xs:enumeration value="leapSecAdded"/>
    <xs:enumeration value="leapSecSubtracted"/>
    <xs:enumeration value="noLeapSecCapability"/>
  </xs:restriction>
</xs:simpleType>
```
5.8.2 The attribute group agQuality

The attribute group agQuality defines the quality information of the event. This definition corresponds to the quality type specified in IEC 61850-7-3 (Ed. 1):2003 [B3]. The attribute group agQuality is defined as follows:

```xml
<xs:attributeGroup name="agQuality">
  <xs:attribute name="validity" type="tValidityEnum" use="optional" default="good"/>
  <xs:attributeGroup ref="agDetailQuality"/>
  <xs:attribute name="source" type="tSourceEnum" use="optional" default="process"/>
  <xs:attribute name="test" type="xs:boolean" use="optional" default="0"/>
  <xs:attribute name="operatorBlocked" type="xs:boolean" use="optional" default="0"/>
</xs:attributeGroup>
```

Table 9 provides the five attributes of agQuality.

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>validity</td>
<td>The allowed values are good, invalid, questionable. The default value is good.</td>
<td>O</td>
</tr>
<tr>
<td>agDetailQuality</td>
<td>The group of attributes specifying in more detail the quality value whose validity is invalid or questionable.</td>
<td>O</td>
</tr>
<tr>
<td>source</td>
<td>Information related to the origin of a value: process or substituted.</td>
<td>O</td>
</tr>
<tr>
<td>test</td>
<td>When set to true, test indicates a value being a test value and not to be used for operational purposes.</td>
<td>O</td>
</tr>
<tr>
<td>operatorBlocked</td>
<td>When set to true, operatorBlocked indicates that the value has been blocked by an operator and there is no further update of the value.</td>
<td>O</td>
</tr>
</tbody>
</table>

The attribute validity is an enumeration, as follows:

```xml
<xs:simpleType name="tValidityEnum">
  <xs:restriction base="xs:Name">
    <xs:enumeration value="good"/>
    <xs:enumeration value="invalid"/>
    <xs:enumeration value="questionable"/>
  </xs:restriction>
</xs:simpleType>
```

The attribute group agDetailQuality gives detailed information about the quality information of the element EntryData.

```xml
<xs:attributeGroup name="agDetailQuality">
  <xs:attribute name="overflow" type="xs:boolean" use="optional" default="0"/>
  <xs:attribute name="outOfRange" type="xs:boolean" use="optional" default="0"/>
  <xs:attribute name="badReference" type="xs:boolean" use="optional" default="0"/>
  <xs:attribute name="oscillatory" type="xs:boolean" use="optional" default="0"/>
  <xs:attribute name="failure" type="xs:boolean" use="optional" default="0"/>
  <xs:attribute name="oldData" type="xs:boolean" use="optional" default="0"/>
  <xs:attribute name="inconsistent" type="xs:boolean" use="optional" default="0"/>
  <xs:attribute name="inaccurate" type="xs:boolean" use="optional" default="0"/>
</xs:attributeGroup>
```
5.8.3 DataRef

The element DataRef is a reference to the value of an EntryData. DataRef and Value may be used to describe a typical event into an SOE report. The value of DataRef is not standardized; it can be any tag attached to a value (Val), except for events recorded in the context of an IEC 61850 Log, where the value of DataRef shall follow specific rules (see 5.8.5). DataRef is of type tDataRef, which is based on the type tName.

tDataRef has two optional attributes: description and funcName (which is defined in 5.8.5). The attributes are of type tAnyName.

<xs:complexType name="tDataRef">
   <xs:simpleContent>
      <xs:extension base="sch:tName">
         <xs:attribute name="desc" type="sch:tAnyName"/>
         <xs:attribute name="funcName" type="sch:tAnyName"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>

5.8.4 Val

The element Val contains the value of the logged data referenced by DataRef. The element Val is based on the type tAnyName. It contains one optional attribute “type,” which is defined in 5.8.5.

<xs:complexType name="tVal">
   <xs:simpleContent>
      <xs:extension base="sch:tAnyName">
         <xs:attribute name="type" type="sch:tValTypeEnum" use="optional" default="xs:normalizedString"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>

5.8.5 The use of the elements DataRef, Val, and DA in the context of an IEC 61850 Log

A COMFEDE file used to exchange logged information coming from an IEC 61850 Log shall follow the rules defined hereafter.

The value of the element DataRef shall be the ObjectReference of a DataSet member, as defined in IEC 61850-7-2 (Ed. 1):2003 [B2], that is, it shall only contain allowed characters and not components (like the Functional Constraint) introduced by protocol mappings as in IEC 61850-8-1 (Ed. 1):2004 [B4]. The value of the element DataRef shall be an IED related name as defined in IEC 61850-6 (Ed. 1):2004 [B1]. However, for applications needing a functional view independent from the IED related name, the optional attribute funcName of the element DataRef may be used to specify a functional name (i.e., a name independent from any specific substation automation system implementation). The functional name should be based on the substation structure names as defined in IEC 61850-6 (Ed. 1):2004 [B1]. An example is as follows:

<EntryData t="2005-03-11T14:38:12.423852Z" eventType="SysEvt" faultTrigger="true">
   <DataRef funcName="Shannon_120kV_L1202/Z1PDIS1.Op" desc="Zone 1 Fault Operate (trip decision)">21L1202Prot/Z1PDIS1.Op</DataRef>
   ...
</EntryData>
The element Val has an optional attribute type that is an enumeration of type tValTypeEnum.

```xml
<xs:simpleType name="tValTypeEnum">
  <xs:restriction base="xs:normalizedString">
    <xs:enumeration value="xs:boolean"/>
    <xs:enumeration value="xs:integer"/>
    <xs:enumeration value="xs:double"/>
    <xs:enumeration value="xs:base64Binary"/>
    <xs:enumeration value="xs:normalizedString"/>
  </xs:restriction>
</xs:simpleType>
```

The value coding syntax of the Val element cannot be directly validated by the COMFEDE schema. However, the attribute type may be used by a receiver to understand the value coding syntax of the Val element. For example, if the declared type value is xs:Boolean, the Val element value must either be true, false, 1, or 0. An instance file should make use of the attribute type. In any case, the value coding syntax of the Val element shall follow the type mapping specified in IEC 61850-6 (Ed. 1):2004 [B1] (i.e., between the XML schema data type coding definitions and the IEC 61850-7 basic data types).

The element DA shall only be used in the context of an IEC 61850 Log. It has three mandatory attributes as defined in Table 10.

```xml
<xs:complexType name="tDA">
  <xs:attribute name="name" type="sch:tName" use="required"/>
  <xs:attribute name="val" type="sch:tAnyName" use="required"/>
  <xs:attribute name="valType" type="sch:tValTypeEnum" use="required"/>
</xs:complexType>
```

Table 10—Attributes of the element DA

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The remaining part of the ObjectReference of a DataSet member, the first part being the value of the element DataRef</td>
<td>M</td>
</tr>
<tr>
<td>val</td>
<td>The value of the logged data attribute referenced by DA. It is based on the type tAnyName.</td>
<td>M</td>
</tr>
<tr>
<td>type</td>
<td>enumeration of type tValTypeEnum. Is identical to the attribute type of the element Val.</td>
<td>M</td>
</tr>
</tbody>
</table>

The following examples show two possibilities of recording an IEC 61850 Log event. Both contain the same three-phase current measurements recorded by the IED named 21L1202.

```xml
<Entry entryId="4294967292" timeOfEntry="2005-03-11T14:38:13Z">
  <EntryData t="2005-03-11T14:38:12.423852Z">
    <DataRef>21L1202Meas/MMXU1.A</DataRef>
    <DA name="phsA.cVal.mag.f" val="1023" valType="xs:double"/>
    <DA name="phsB.cVal.mag.f" val="1022" valType="xs:double"/>
    <DA name="phsC.cVal.mag.f" val="1019" valType="xs:double"/>
  </EntryData>
</Entry>

<Entry entryId="4294967292" timeOfEntry="2005-03-11T14:38:13Z">
  <EntryData t="2005-03-11T14:38:12.423852Z">
    <DataRef>21L1202Meas/MMXU1.A.phsA.cVal.mag.f</DataRef><Val type="xs:double" val="1023"/>
  </EntryData>
</Entry>
```
5.8.6 ReasonCode

The optional element ReasonCode defines the reason for inclusion of the data recorded. This definition corresponds to the type TriggerConditions defined in IEC 61850-7-2 (Ed. 1):2003 [B2].

The element ReasonCode is of type tTrgOpt as follows:

```xml
<xs:element name="ReasonCode" type="CED:tTrgOpt" minOccurs="0"/>
```

The tTrgOpt type contains one attribute group, agTrgOpt, which corresponds to the different trigger options. Two of them are mandatory (dchg and qchg).

```xml
<xs:complexType name="tTrgOpt">
  <xs:attributeGroup ref="CED:agTrgOpt"/>
</xs:complexType>
```

```xml
<xs:attributeGroup name="agTrgOpt">
  <xs:attribute name="dchg" type="xs:boolean" use="required"/>
  <xs:attribute name="qchg" type="xs:boolean" use="required"/>
  <xs:attribute name="dupd" type="xs:boolean" use="optional"/>
  <xs:attribute name="period" type="xs:boolean" use="optional"/>
  <xs:attribute name="gi" type="xs:boolean" use="optional"/>
  <xs:attribute name="apptrg" type="xs:boolean" use="optional"/>
</xs:attributeGroup>
```

There is one ReasonCode per EntryData.

5.9 PayloadData

The element PayloadData is intended to be used to give additional information related to the recorded event. The UML class diagram in Figure 4 shows the structure of the PayloadData element.

The element PayloadData is of type tPayloadData and contains the following optional elements:

- ComtradeFileName, which is the name(s) of the COMTRADE file(s) triggered by the event. The name(s) should be in accordance with IEEE Std C37.232™-2007 [B6].

- ElectMeas, a set of electrical measurements recorded before, during, or after the event and expressed in primary values.

- Fq, the evaluated frequency of the power line or bus when the event occurred.

- FltRpt, additional information related to an electrical fault on a power network.
— ActTgtData, the device targets that are active after the event.
— RecloserRpt, information related to the status of the line recloser.
— BreakerRpt, information related to the breaker(s) involved in the event.

Figure 4—PayloadData UML diagram

tPayloadData is based on tBaseElementForExtension, which means it may be extended as explained in 5.4.2.1. An example of an extension is given in Annex B.

The tPayloadData section is as follows:

```xml
<xs:complexType name="tPayloadData">
  <xs:complexContent>
    <xs:extension base="CED:tBaseElementForExtension">
      <xs:sequence>
        <xs:element name="ComtradeFileName" type="sch:tFileName" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="ElectMeas" type="sch:tElectricalMeasurements" minOccurs="0"/>
        <xs:element name="Fq" minOccurs="0">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="sch:tLineFrequency"/>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```
The type `tPayloadData` has four optional attributes shown in Table 11 and seven optional elements, which are described later.

### Table 11 —Attributes of the element type `tPayloadData`

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sGroup</code></td>
<td>Is the setting group number in effect during the event.</td>
<td>O</td>
</tr>
<tr>
<td><code>payloadType</code></td>
<td>Gives an indication on the type of Payload.</td>
<td>O</td>
</tr>
<tr>
<td><code>faultType</code></td>
<td>Is an enumeration defining the type of the fault.</td>
<td>O</td>
</tr>
<tr>
<td><code>trip</code></td>
<td>Indicates if, following a triggered event, a trip action was executed by the device or not. True or false.</td>
<td>O</td>
</tr>
</tbody>
</table>

The attribute `payloadType` is intended to facilitate the interpretation of the values of the Payload element. For example, recorded measurements may have to be interpreted in the context of a line protection or a transformer protection. Taking into account that by definition, multifunction IEDs may implement a mixture of different types of functions, the `payloadType` attribute values are not standardized. Examples of values are line protection, distance protection, differential line protection, differential transformer protection, and so on.

The attribute `faultType` is an enumeration defining the phases involved during the fault occurrence; its values are as follows:

```xml
<xs:simpleType name="tFaultTypeEnum">
  <xs:restriction base="xs:Name">
    <xs:enumeration value="AG"/>
    <xs:enumeration value="BG"/>
    <xs:enumeration value="CG"/>
    <xs:enumeration value="AB"/>
    <xs:enumeration value="BC"/>
    <xs:enumeration value="CA"/>
    <xs:enumeration value="ABG"/>
    <xs:enumeration value="BCG"/>
    <xs:enumeration value="CAG"/>
    <xs:enumeration value="ABC"/>
    <xs:enumeration value="unknown"/>
  </xs:restriction>
</xs:simpleType>
```
5.9.1 Electrical measurements

The UML diagram in Figure 5 shows the type `tElectricalMeasurements`.

---

**Figure 5—tElectricalMeasurements UML diagram**

The `ElectMeas` element represents electrical analog values recorded at the time when the event occurred. Its type is `tElectricalMeasurements`, which contains the following optional elements:

- Three-phase voltage and current measurements
- Single-phase voltage and current measurements
- Direct current (dc) voltage and current measurements

The type `tElectricalMeasurements` may contain an unlimited number of the previous elements.
As shown in Figure 5, all measurements (alternating current [ac] and dc) inherit from the type tMeasurementCapture the attributes presented in Table 12.

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desc</td>
<td>A descriptive text of the measurement.</td>
<td>O</td>
</tr>
<tr>
<td>fltCorr</td>
<td>An enumeration whose values indicate the correlation of the recorded measurements with the instant of a fault detection. Enumeration values are prefault, fault, postfault, and none; none is the schema default value.</td>
<td>O</td>
</tr>
<tr>
<td>preTmms</td>
<td>Pretrigger time. The number of milliseconds prior to the fault trigger when the measurements were recorded.</td>
<td>O</td>
</tr>
<tr>
<td>pstTmms</td>
<td>Posttrigger time. The number of milliseconds following the fault trigger when the measurements were recorded.</td>
<td>O</td>
</tr>
<tr>
<td>xfmrRatPrimFact</td>
<td>Is the instrument transformer ratio primary factor.</td>
<td>O</td>
</tr>
<tr>
<td>xfmrRatSecFact</td>
<td>It is the instrument transformer ratio secondary factor.</td>
<td>O</td>
</tr>
</tbody>
</table>

The two last attributes in the table are defined by the attribute group agPrimaryToSecondaryRatios. Their default value is 1, meaning that if the attributes are not present, their value is implicitly interpreted as being 1.
5.9.1.1 tThreePhaseVoltage

The UML diagram in Figure 6 shows the type tThreePhaseVoltage.

It contains the following elements:

- Phase-to-neutral voltage measurements
- Phase-to-phase voltage measurements
- Voltage sequence component measurements

Figure 6—tThreePhaseVoltage UML diagram

The tThreePhaseVoltage section is as follows:

```xml
<xs:complexType name="tThreePhaseVoltage">
  <xs:complexContent>
    <xs:extension base="tACMeasurementCapture">
      <xs:sequence>
        <xs:element name="PhsA" type="tVoltageVector" minOccurs="0"/>
        <xs:element name="PhsB" type="tVoltageVector" minOccurs="0"/>
        <xs:element name="PhsC" type="tVoltageVector" minOccurs="0"/>
        <xs:element name="Neut" type="tVoltageVector" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Value of the measured neutral voltage at in independent input. Sometimes called Vx.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```
The type tThreePhaseVoltage is based on the type tACMeasurementCapture, which extends the content inherit from the type tMeasurementCapture by adding the attribute presented in Table 13.

**Table 13—Additional attribute of the complex type tACMeasurementCapture**

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>clcMth</td>
<td>An enumeration whose values specify how the data, which represent analog values, have been calculated.</td>
<td>O</td>
</tr>
</tbody>
</table>

The calculation method enumeration values are as follows:

```xml
<xs:simpleType name="tClecMthEnum">
  <xs:restriction base="xs:Name">
  <xs:enumeration value="true_RMS"/>
  <xs:enumeration value="peak_Fundamental"/>
  <xs:enumeration value="rms_Fundamental"/>
  <xs:enumeration value="rms_2ndHarmonic"/>
  <xs:enumeration value="rms_3rdHarmonic"/>
  <xs:enumeration value="rms_4thHarmonic"/>
  <xs:enumeration value="rms_5thHarmonic"/>
  <xs:enumeration value="other"/>
  </xs:restriction>
</xs:simpleType>
```

All phase measurements are voltage vector types (tVoltageVector), whereas the sequence component measurements are voltage sequence types (tSequenceV).

The type tSequenceV is used for voltage symmetrical component measurements. V1, V2, and V0, respectively, correspond to the positive, negative, and zero sequence measurements. Negative and zero sequence measurements also allow for 3V2 and 3V0 quantities. All measurements are voltage vector types (tVoltageVector).
The type tVoltageVector contains a mandatory element Mag, which has a mandatory attribute unit whose value is fixed to V and an optional attribute multiplier, which has the default value k.

The type tVoltageVector may also contain an optional element Ang, which has a mandatory attribute unit whose value is fixed to deg and an optional attribute multiplier whose value is intentionally left empty. When the clcMth attribute value is true_RMS, no Ang shall be declared.

5.9.1.2 tThreePhaseCurrent

The UML diagram in Figure 7 shows the type tThreePhaseCurrent. The type is a holder for the following elements:

— Phase current measurements
— Neutral, residual, and net current measurements
— Current sequence component measurements
The `tThreePhaseCurrent` section is as follows:

```xml
<xs:complexType name="tThreePhaseVoltage">
  <xs:complexContent>
    <xs:extension base="tACMeasurementCapture">
      <xs:sequence>
        <xs:element name="PhsA" type="tVoltageVector" minOccurs="0"/>
        <xs:element name="PhsB" type="tVoltageVector" minOccurs="0"/>
        <xs:element name="PhsC" type="tVoltageVector" minOccurs="0"/>
        <xs:element name="Neut" type="tVoltageVector" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Value of the measured neutral voltage at an independent input. Sometimes called Vx.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Res" type="tCurrentVector" minOccurs="0"/>
        <xs:element name="PhsAB" type="tVoltageVector" minOccurs="0"/>
        <xs:element name="PhsBC" type="tVoltageVector" minOccurs="0"/>
        <xs:element name="PhsCA" type="tVoltageVector" minOccurs="0"/>
        <xs:element name="Seq" type="tSequenceV" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

**Figure 7—tThreePhaseCurrent UML diagram**

The `tThreePhaseCurrent` section is as follows:
The type tThreePhaseCurrent is based on the type tACMeasurementCapture, which extends the content inherit from the type tMeasurementCapture by adding the clcMth attribute (see 5.9.1.1). The type tThreePhaseCurrent also add an optional attribute as shown in Table 14.

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>differentialInputId</td>
<td>A name intended to be used to identify the different current measurements in case of a differential protection PayLoadData.</td>
<td>O</td>
</tr>
</tbody>
</table>

All phase measurements are current vector types (tCurrentVector), whereas the sequence component measurements are current sequence types (tSequenceA).

The type tSequenceA is used for current symmetrical component measurements. I1, I2, and I0, respectively, correspond to the positive, negative, and zero sequence measurements. Negative and zero sequence measurements also allow for 3I2 and 3I0 quantities. All measurements are current vector types (tCurrentVector).

The type tCurrentVector contains a mandatory element Mag, which has a mandatory attribute unit whose value is fixed to A and an optional attribute multiplier whose value is intentionally left empty.

The type tCurrentVector may also contain an optional element Ang, which has a mandatory attribute unit whose value is fixed to deg and an optional attribute multiplier whose value is intentionally left empty. When the clcMth attribute value is true_RMS, no Ang shall be declared.
5.9.1.3 Single-phase and direct current (dc) measurements

The UML diagram in Figure 8 shows the single phase and the dc measurement definitions, which are part of the type tElectricalMeasurements. The main elements are as follows:

- Single-phase voltage measurement (tVoltage)
- Single-phase current measurement (tCurrent)
- DC voltage measurement (tDCVoltage)
- DC current measurement (tDCCurrent)
Figure 8—Single-phase and dc measurement UML diagram

The types tVoltage and tCurrent represent single-phase measurements and are based on the type tACMeasurementCapture (Figure 5).

The type tVoltage contains one mandatory element Mag, which has a mandatory attribute unit whose value is fixed to V and an optional attribute multiplier that has the default value k.

The type tCurrent contains a mandatory element Mag, which has a mandatory attribute unit whose value is fixed to A and an optional attribute multiplier whose value is intentionally left empty.

```xml
<xs:complexType name="tVoltage">
    <xs:complexContent>
        <xs:extension base="tACMeasurementCapture">
            <xs:sequence>
                <xs:element name="Mag">
                    <xs:complexType>
                        <xs:simpleContent>
                            <xs:restriction base="tValueWithUnit">
                                <xs:attribute name="unit" type="tSIUnitEnum" use="required" fixed="V"/>
                            </xs:restriction>
                        </xs:simpleContent>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```
The types tDCVoltage and tDCCurrent represent dc measurements and are based on the type tMeasurementCapture (Figure 5).

The type tDCVoltage contains the following three optional elements:

   a) Vol, the voltage between poles
   b) VolPsGnd, the voltage between the negative pole and ground
   c) VolNgGnd, the voltage between the negative pole and ground

Each of the elements contains one mandatory element Mag, which has a mandatory attribute unit whose value is fixed to V and an optional attribute multiplier that has the default value k.

The type tCurrent contains one mandatory element Mag, which has a mandatory attribute unit whose value is fixed to A and an optional attribute multiplier whose value is intentionally left empty.
<xs:restriction base="tValueWithUnit">
    <xs:attribute name="unit" type="tSIUnitEnum" use="required" fixed="V"/>
    <xs:attribute name="multiplier" type="tUnitMultiplierEnum" use="optional" default="k"/>
</xs:restriction>

<x:complexType>
    <xs:complexType>
        <xs:restriction base="tValueWithUnit">
            <xs:attribute name="unit" type="tSIUnitEnum" use="required" fixed="V"/>
            <xs:attribute name="multiplier" type="tUnitMultiplierEnum" use="optional" default="k"/>
        </xs:restriction>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<x:complexType name="tDCCurrent">
    <xs:complexContent>
        <xs:extension base="tMeasurementCapture">
            <xs:sequence>
                <xs:element name="Amp">
                    <xs:complexType>
                        <xs:complexContent>
                            <xs:restriction base="tValueWithUnit">
                                <xs:attribute name="unit" type="tSIUnitEnum" use="required" fixed="A"/>
                                <xs:attribute name="multiplier" type="tUnitMultiplierEnum" use="optional" default="m"/>
                            </xs:restriction>
                        </xs:complexContent>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
5.9.2 Frequency

The element Fq contains the value of the frequency during the event. Frequency is of type tLineFrequency, which has a mandatory attribute unit whose value is fixed to Hz and an optional attribute multiplier whose value is intentionally left empty.

```
<xs:complexType name="tLineFrequency">
    <xs:complexContent>
        <xs:extension base="xs:decimal">
            <xs:attribute name="unit" type="tSIUnitEnum" use="required" fixed="Hz" />
            <xs:attribute name="multiplier" type="tUnitMultiplierEnum" use="optional" default="" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

5.9.3 FaultReport

The FltRpt element is used to group diverse pieces of information related to a power system fault detection. Its type is tFaultReport, which contains the following four optional elements:

a) DistToFlt, the evaluated distance to the fault expressed in kilometers
b) FltRes, the evaluated resistance of the fault expressed in ohms
c) FltCirc, name of the circuit/component identified as being faulty
d) FltClrTm, the time to clear the fault expressed in milliseconds

```
<xs:complexType name="tFaultReport">
    <xs:complexContent>
        <xs:extension base="tBaseElementForExtension">
            <xs:sequence>
                <xs:element name="DistToFlt" type="tDistanceToFault" minOccurs="0" />
                <xs:element name="FltRes" type="tFaultResistance" minOccurs="0" />
                <xs:element name="FltCirc" type="tFaultedCircuit" minOccurs="0" />
                <xs:element name="FltClrTm" type="tFaultClearingTimeInMilliSec" minOccurs="0" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```
5.9.4 ActiveTargetData

The element ActTgtData identifies the relay targets that were active after the fault. Its type is tActiveTargetData, which contains an unlimited number of elements Target whose base type is tNaming.

The tRecloserReport has attributes as described in Table 15.
### Table 15—Attributes of the complex type tRecloserReport

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Enumeration of the different modes of the recloser.</td>
<td>M</td>
</tr>
<tr>
<td>status</td>
<td>Enumeration of the different statuses of the recloser.</td>
<td>M</td>
</tr>
<tr>
<td>shotCnt1ph</td>
<td>Counter of single-phase reclosing actions.</td>
<td>M</td>
</tr>
<tr>
<td>shotCnt3ph</td>
<td>Counter of three-phase reclosing actions.</td>
<td>M</td>
</tr>
</tbody>
</table>

```xml
<xs:simpleType name="tRecloserModeEnum">
    <xs:restriction base="xs:Name">
        <xs:enumeration value="disabled"/>
        <xs:enumeration value="singlePole"/>
        <xs:enumeration value="threePole"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tRecloserStatusEnum">
    <xs:restriction base="xs:Name">
        <xs:enumeration value="ready">
            <xs:annotation>
                <xs:documentation>Recloser is Ready: waiting for protection trigger</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="inProgress">
            <xs:annotation>
                <xs:documentation>Recloser is “In Progress” i.e., performing the number of predefined auto-reclose cycles</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="lockout">
            <xs:annotation>
                <xs:documentation>Recloser is locked out, e.g. if the number of trips exceeds the maximum number of reclose actions</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="blocked">
            <xs:annotation>
                <xs:documentation>Recloser is blocked, e.g., by a protection signal or by manual tripping</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
    </xs:restriction>
</xs:simpleType>
```

### 5.9.6 BreakerReport

The BkrRpt element contains recorded information related to breakers.

```xml
<xs:complexType name="tBreakerReport">
    <xs:complexContent>
        <xs:extension base="tBaseElementForExtension">
            <xs:attribute name="name" type="tAnyName" use="optional"/>
            <xs:attribute name="pos" type="tBreakerPositionEnum" use="optional"/>
            <xs:attribute name="operTmms" type="xs:integer" use="optional"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```
The `tBreakerReport` has attributes as described in Table 16.

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the breaker.</td>
<td>O</td>
</tr>
<tr>
<td>pos</td>
<td>The position of the breaker.</td>
<td>O</td>
</tr>
<tr>
<td>operTmms</td>
<td>The breaker operate time in milliseconds.</td>
<td>O</td>
</tr>
</tbody>
</table>
Annex A

(normative)

The COMFEDE XML schema

This annex includes the content of the different files describing the COMFEDE XML schema.

A.1 The file COMFEDE.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:CED="http://www.pes-psrc.org/Subcommittee/H/COMFEDE"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:sch="http://www.pes-psrc.org/Subcommittee/H/CommonTypes"
targetNamespace="http://www.pes-psrc.org/Subcommittee/H/COMFEDE"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0">
    <xs:annotation>
        <xs:documentation xml:lang="en"/>
    </xs:annotation>
    <xs:import namespace="http://www.pes-psrc.org/Subcommittee/H/CommonTypes"
schemaLocation="H_CommonTypes.xsd"/>
    <xs:complexType name="tBaseElementForExtension" abstract="true">
        <xs:sequence>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="Text" type="CED:tText" minOccurs="0"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="tAnyContentFromOtherNamespace" abstract="true" mixed="true">
        <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:any namespace="##other" processContents="lax"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="tText" mixed="true">
        <xs:complexContent mixed="true">
            <xs:extension base="CED:tAnyContentFromOtherNamespace"/>
            <xs:attribute name="source" type="xs:anyURI" use="optional"/>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="COMFEDE">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="CED:Log" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Log">
        <xs:complexType>
            <xs:complexContent>
                <xs:extension base="CED:tLog"/>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>
</xs:schema>
```
<xs:attribute name="version" type="CED:tCedVersion" use="required" fixed="2010"/>
<xs:attribute name="revision" type="CED:tCedRevision" use="required" fixed="A"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
<xs:complexType name="tLog">
<xs:complexContent>
<xs:extension base="CED:tBaseElementForExtension">
<xs:sequence>
<xs:element name="Location">
<xs:complexType>
<xs:complexContent>
<xs:extension base="sch:tLocation"/>
</xs:complexContent>
</xs:complexType>
</xs:element>
<xs:element name="Entry" type="CED:tEntry" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="SettingFileName" type="sch:tFileName" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="logName" type="xs:normalizedString"/>
<xs:attribute name="logRef" type="xs:normalizedString"/>
<xs:attribute name="oldEntrTm" type="xs:dateTime"/>
<xs:attribute name="newEntrTm" type="xs:dateTime"/>
<xs:attribute name="oldEntr" type="xs:unsignedLong"/>
<xs:attribute name="newEntr" type="xs:unsignedLong"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="tEntry">
<xs:complexContent>
<xs:extension base="CED:tBaseElementForExtension">
<xs:sequence>
<xs:element name="EntryData" type="CED:tEntryData" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="entryId" type="xs:unsignedLong"/>
<xs:annotation>
<xs:documentation>Sequence number.</xs:documentation>
</xs:annotation>
<xs:attribute name="timeOfEntry" type="xs:dateTime"/>
<xs:annotation>
<xs:documentation>Time when the Entry is added to the buffer</xs:documentation>
</xs:annotation>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="tEntryData">
<xs:complexContent>
<xs:extension base="CED:tBaseElementForExtension">
<xs:sequence>
<xs:element name="DataRef" type="CED:tDataRef"/>
<xs:choice>
<xs:element name="Val" type="CED:tVal"/>
</xs:choice>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:element name="DA" type="CED:tDA" maxOccurs="unbounded"/>
</xs:choice>
<xs:element name="ReasonCode" type="CED:tTrgOpt" minOccurs="0"/>
<xs:element name="PayloadData" type="CED:tPayloadData" minOccurs="0"/>
</xs:sequence>
</xs:extension>
</xs:complexType>
</xs:complexContent>
</xs:complexType>
</xs:complexType>
<xs:complexType name="tDataRef">
<xs:simpleContent>
<xs:extension base="sch:tName">
<xs:attribute name="desc" type="sch:tAnyName"/>
<xs:attribute name="funcName" type="sch:tAnyName"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="tVal">
<xs:simpleContent>
<xs:extension base="sch:tAnyName">
<xs:attribute name="type" type="sch:tValTypeEnum" use="optional" default="xs:normalizedString"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="tDA">
<xs:attribute name="name" type="sch:tName" use="required"/>
<xs:attribute name="val" type="sch:tAnyName" use="required"/>
<xs:attribute name="valType" type="sch:tValTypeEnum" use="required"/>
</xs:complexType>
<xs:complexType name="tTrgOpt">
<xs:attributeGroup ref="CED:agTrgOpt"/>
</xs:complexType>
<xs:attributeGroup name="agTrgOpt">
<xs:attribute name="dchg" type="xs:boolean" use="required"/>
<xs:attribute name="qchg" type="xs:boolean" use="required"/>
<xs:attribute name="dupd" type="xs:boolean" use="optional"/>
<xs:attribute name="period" type="xs:boolean" use="optional"/>
<xs:attribute name="gi" type="xs:boolean" use="optional"/>
<xs:attribute name="apptrg" type="xs:boolean" use="optional"/>
</xs:attributeGroup>
<xs:complexType name="tPayloadData">
<xs:complexContent>
<xs:extension base="CED:tBaseElementForExtension">
<xs:sequence>
<xs:element name="ComtradeFileName" type="sch:tFileName" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="ElectMeas" type="sch:tElectricalMeasurements" minOccurs="0"/>
<xs:element name="Fq" minOccurs="0"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:complexType>
A.2 The file H_CommonTypes.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.pes-psrc.org/Subcommittee/H/CommonTypes"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.pes-psrc.org/Subcommittee/H/CommonTypes" elementFormDefault="qualified"
  attributeFormDefault="unqualified" version="1.0">
  <xs:include schemaLocation="H_Enums.xsd"/>
  <xs:include schemaLocation="H_SimpleTypes.xsd"/>
  <xs:complexType name="tBaseElementForExtension" abstract="true">
    <xs:sequence>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="sGroup" type="xs:unsignedInt">
      <xs:annotation>
        <xs:documentation>Setting group number</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
  <xs:complexType name="tAnyContentFromOtherNamespace" abstract="true" mixed="true">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:any namespace="##other" processContents="lax"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="tCedVersion">
    <xs:restriction base="sch:tName"/>
  </xs:complexType>
  <xs:complexType name="tCedRevision">
    <xs:restriction base="xs:Name">
      <xs:pattern value="^[A-Z]+$"/>
    </xs:restriction>
  </xs:complexType>
  <xs:complexType name="tNaming">
    <xs:annotation>
      <xs:documentation>Setting group number</xs:documentation>
    </xs:annotation>
    <xs:attribute name="payloadType" type="sch:tName"/>
    <xs:attribute name="faultType" type="sch:tFaultTypeEnum"/>
    <xs:attribute name="trip" type="xs:boolean" default="false"/>
  </xs:complexType>
</xs:schema>
```
<xs:documentation>This type is reused where an element requires a name attribute and an optional
description attribute. </xs:documentation>
</xs:annotation>
<xs:attribute name="name" type="tAnyName" use="required"/>
<xs:attribute name="desc" type="xs:normalizedString"/>
</xs:complexType>
<xs:complexType name="tFileName">
  <xs:simpleContent>
    <xs:extension base="tName">
      <xs:attribute name="ext" type="tFileExtensionEnum" use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tLocation">
  <xs:complexContent>
    <xs:extension base="tBaseElementForExtension">
      <xs:sequence>
        <xs:element name="IEDNameplate" type="tIEDNameplate"/>
      </xs:sequence>
      <xs:attribute name="companyName" type="tName" use="required"/>
      <xs:attribute name="companyIdentificationCode" type="tCompanyIdentificationCode"
default="0"/>
      <xs:attribute name="operatorName" type="tName"/>
      <xs:attribute name="stationName" type="tName" use="required"/>
      <xs:attribute name="voltageLevelInkV" type="xs:decimal"/>
      <xs:attribute name="bayName" type="tName"/>
      <xs:attribute name="latitude" type="xs:float"/>
      <xs:attribute name="longitude" type="xs:float"/>
      <xs:attribute name="tmOfsTmm" default="0">Offset of local time from UTC in minutes</xs:annotation>
    </xs:sequence>
  </xs:complexType>
</xs:complexType>
<xs:complexType name="tIEDNameplate">
  <xs:complexContent>
    <xs:extension base="tBaseElementForExtension">
      <xs:attribute name="name" type="tName" use="required"/>
      <xs:attribute name="vendor" type="tName" use="required"/>
      <xs:attribute name="model" type="tName"/>
      <xs:attribute name="serNum" type="tName"/>
      <xs:attribute name="hwRev" type="tName"/>
      <xs:attribute name="swRev" type="tName"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tValueWithUnit">
  <xs:simpleContent>
<xs:extension base="xs:decimal">
  <xs:attribute name="unit" type="tSIUnitEnum" use="required"/>
  <xs:attribute name="mult" type="tUnitMultiplierEnum" use="optional"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="tElectricalMeasurements">
  <xs:complexContent>
    <xs:extension base="tBaseElementForExtension">
      <xs:sequence>
        <xs:element name="ThreePhsVolt" type="tThreePhaseVoltage" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="ThreePhsCurr" type="tThreePhaseCurrent" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="Volt" type="tVoltage" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="Curr" type="tCurrent" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="DCVolt" type="tDCVoltage" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="DC" type="tDCCurrent" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tMeasurementCapture" abstract="true">
  <xs:complexContent>
    <xs:extension base="tBaseElementForExtension">
      <xs:attribute name="desc" type="xs:normalizedString"/>
      <xs:attribute name="fltCorrel" default="none">
        <xs:simpleType>
          <xs:restriction base="tFaultCorrelationEnum"/>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="preTmms" type="xs:integer"/>
      <xs:attribute name="pstTmms" type="xs:integer"/>
      <xs:attributeGroup ref="agPrimaryToSecondaryRatios"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tACMeasurementCapture" abstract="true">
  <xs:complexContent>
    <xs:extension base="tMeasurementCapture">
      <xs:attribute name="clcMth" type="tClcMthEnum"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tVoltage">
  <xs:complexContent>
    <xs:extension base="tACMeasurementCapture">
      <xs:sequence>
        <xs:element name="Mag">
          <xs:complexType>
            <xs:simpleContent>
              <xs:restriction base="tValueWithUnit">
                <xs:attribute name="unit" type="tSIUnitEnum" use="required" fixed="V"/>
                <xs:attribute name="mult" type="tUnitMultiplierEnum" use="optional" default="k"/>
              </xs:restriction>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="tThreePhaseVoltage">
<xs:complexContent>
<xs:extension base="tACMeasurementCapture">
<xs:sequence>
<xs:element name="PhsA" type="tVoltageVector" minOccurs="0"/>
<xs:element name="PhsB" type="tVoltageVector" minOccurs="0"/>
<xs:element name="PhsC" type="tVoltageVector" minOccurs="0"/>
<xs:element name="Neut" type="tVoltageVector" minOccurs="0">
<xs:annotation>
<xs:documentation>Value of the measured neutral voltage at in independent input. Sometimes called Vx.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Res" type="tCurrentVector" minOccurs="0">
<xs:annotation>
<xs:documentation>Residual voltage calculated from the sum of the three phase-to-neutral voltages.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="PhsAB" type="tVoltageVector" minOccurs="0"/>
<xs:element name="PhsBC" type="tVoltageVector" minOccurs="0"/>
<xs:element name="PhsCA" type="tVoltageVector" minOccurs="0"/>
<xs:element name="Seq" type="tSequenceV" minOccurs="0"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="tDCVoltage">
<xs:complexContent>
<xs:extension base="tMeasurementCapture">
<xs:sequence>
<xs:element name="Vol" minOccurs="0">
<xs:complexType>
<xs:simpleContent>
<xs:restriction base="tValueWithUnit">
<xs:attribute name="unit" type="tSIUnitEnum" use="required" fixed="V"/>
<xs:attribute name="mult" type="tUnitMultiplierEnum" use="optional" default="k"/>
</xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="VolPsGnd" minOccurs="0">
<xs:complexType>
<xs:simpleContent>
<xs:restriction base="tValueWithUnit">
<xs:attribute name="unit" type="tSIUnitEnum" use="required" fixed="V"/>
<xs:attribute name="mult" type="tUnitMultiplierEnum" use="optional" default="k"/>
</xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
Sometimes called IG. Value of the measured neutral current at in independent input.

Residual current is the algebraic sum of the instantaneous values of currents flowing through all live conductors (i.e., sum of phase currents) of a circuit at a point of the electrical installation.

Net current is the algebraic sum of the instantaneous values of currents flowing through all live conductors (sum of phase currents) and neutral of a circuit at a point of the
electrical installation.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Seq" type="tSequenceA" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="differentialInputId" type="tName" use="optional"/>
</extension>
</complexType>
</complexContent>
</complexType>
<complexType name="tDCCurrent">
<complexContent>
<extension base="tMeasurementCapture">
<sequence>
<element name="Amp">
<complexType>
<simpleContent>
<restriction base="tValueWithUnit">
<attribute name="unit" type="tSIUnitEnum" use="required" fixed="A"/>
</restriction>
</simpleContent>
</complexType>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="tVoltageVector">
<sequence>
<element name="Mag">
<complexType>
<simpleContent>
<restriction base="tValueWithUnit">
<attribute name="unit" type="tSIUnitEnum" use="required" fixed="V"/>
<attribute name="mult" type="tUnitMultiplierEnum" use="optional" default="k"/>
</restriction>
</simpleContent>
</complexType>
</element>
<element name="Ang" minOccurs="0">
<complexType>
<simpleContent>
<restriction base="tValueWithUnit">
<attribute name="unit" type="tSIUnitEnum" use="required" fixed="deg"/>
</restriction>
</simpleContent>
</complexType>
</element>
</sequence>
</complexType>
<complexType name="tCurrentVector">
<sequence>
<element name="Mag">
<complexType>
<simpleContent>
<restriction base="tValueWithUnit">
<attribute name="unit" type="tSIUnitEnum" use="required" fixed="A"/>
</restriction>
</simpleContent>
</complexType>
</element>
</sequence>
</complexType>
<xs:complexType name="tSequenceA">
  <xs:sequence>
    <xs:element name="I1" type="tCurrentVector" minOccurs="0"/>
    <xs:element name="I2" type="tCurrentVector" minOccurs="0"/>
    <xs:element name="I2x3" type="tCurrentVector" minOccurs="0"/>
    <xs:element name="I0" type="tCurrentVector" minOccurs="0"/>
    <xs:element name="I0x3" type="tCurrentVector" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="tSequenceV">
  <xs:sequence>
    <xs:element name="V1" type="tVoltageVector" minOccurs="0"/>
    <xs:element name="V2" type="tVoltageVector" minOccurs="0"/>
    <xs:element name="V2x3" type="tVoltageVector" minOccurs="0"/>
    <xs:element name="V0" type="tVoltageVector" minOccurs="0"/>
    <xs:element name="V0x3" type="tVoltageVector" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="tDistanceToFault">
  <xs:simpleContent>
    <xs:restriction base="tValueWithUnit">
      <xs:attribute name="unit" type="tSIUnitEnum" use="required" fixed="m"/>
      <xs:attribute name="mult" type="tUnitMultiplierEnum" use="optional" fixed="k"/>
    </xs:restriction>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="tFaultResistance">
  <xs:simpleContent>
    <xs:restriction base="tValueWithUnit">
      <xs:attribute name="unit" type="tSIUnitEnum" use="required" fixed="ohm"/>
    </xs:restriction>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="tRecloserReport">
  <xs:complexContent>
    <xs:extension base="tBaseElementForExtension">
      <xs:attribute name="mode" type="tRecloserModeEnum" use="required"/>
      <xs:attribute name="status" type="tRecloserStatusEnum" use="required"/>
      <xs:attribute name="shotCnt1ph" type="xs:nonNegativeInteger" use="required"/>
      <xs:attribute name="shotCnt3ph" type="xs:nonNegativeInteger" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
A.3 The file H_Enums.xsd

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ns="http://www.pes-psrc.org/Subcommittee/H/CommonTypes" targetNamespace="http://www.pes-psrc.org/Subcommittee/H/CommonTypes" elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0">
  <xs:include schemaLocation="H_SimpleTypes.xsd"/>
  <xs:annotation>
    <xs:documentation>
      A.3 The file H_Enums.xsd
    </xs:documentation>
  </xs:annotation>
</xs:schema>
<xs:documentation xml:lang="en"/>
</xs:annotation>
<xs:simpleType name="tSIUnitEnum">
  <xs:annotation>
    <xs:documentation>none may be used for per unit (p.u.) values in combination with the unit multiplier empty value</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:token">
    <xs:enumeration value="none"/>
    <xs:enumeration value="m"/>
    <xs:enumeration value="kg"/>
    <xs:enumeration value="s"/>
    <xs:enumeration value="A"/>
    <xs:enumeration value="K"/>
    <xs:enumeration value="mol"/>
    <xs:enumeration value="cd"/>
    <xs:enumeration value="deg"/>
    <xs:enumeration value="rad"/>
    <xs:enumeration value="sr"/>
    <xs:enumeration value="Gy"/>
    <xs:enumeration value="q"/>
    <xs:enumeration value="°C"/>
    <xs:enumeration value="Sv"/>
    <xs:enumeration value="F"/>
    <xs:enumeration value="C"/>
    <xs:enumeration value="S"/>
    <xs:enumeration value="H"/>
    <xs:enumeration value="V"/>
    <xs:enumeration value="ohm"/>
    <xs:enumeration value="J"/>
    <xs:enumeration value="N"/>
    <xs:enumeration value="Hz"/>
    <xs:enumeration value="lx"/>
    <xs:enumeration value="lm"/>
    <xs:enumeration value="Wb"/>
    <xs:enumeration value="T"/>
    <xs:enumeration value="W"/>
    <xs:enumeration value="Pa"/>
    <xs:enumeration value="m²"/>
    <xs:enumeration value="m³"/>
    <xs:enumeration value="m³/s"/>
    <xs:enumeration value="m/m³"/>
    <xs:enumeration value="M"/>
    <xs:enumeration value="W/m K"/>
    <xs:enumeration value="J/K"/>
    <xs:enumeration value="ppm"/>
    <xs:enumeration value="1/s"/>
    <xs:enumeration value="rad/s"/>
    <xs:enumeration value="VA"/>
    <xs:enumeration value="Watts"/>
    <xs:enumeration value="VAr"/>
    <xs:enumeration value="phi"/>
  </xs:restriction>
</xs:simpleType>
<xs:enumeration value="cos(phi)"/>
<xs:enumeration value="Vs"/>
<xs:enumeration value="V²"/>
<xs:enumeration value="As"/>
<xs:enumeration value="A²"/>
<xs:enumeration value="VAh"/>
<xs:enumeration value="Wh"/>
<xs:enumeration value="VARh"/>
<xs:enumeration value="V/Hz"/>
<xs:enumeration value="Hz/s"/>
<xs:enumeration value="char"/>
<xs:enumeration value="char/s"/>
<xs:enumeration value="kgm²"/>
<xs:enumeration value="dB"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="tUnitMultiplierEnum">
<xs:restriction base="xs:normalizedString">
<xs:enumeration value="n"/>
<xs:enumeration value="m"/>
<xs:enumeration value="k"/>
<xs:enumeration value="M"/>
<xs:enumeration value="mu"/>
<xs:enumeration value="y"/>
<xs:enumeration value="z"/>
<xs:enumeration value="a"/>
<xs:enumeration value="f"/>
<xs:enumeration value="p"/>
<xs:enumeration value="n"/>
<xs:enumeration value="c"/>
<xs:enumeration value="d"/>
<xs:enumeration value="da"/>
<xs:enumeration value="h"/>
<xs:enumeration value="G"/>
<xs:enumeration value="T"/>
<xs:enumeration value="P"/>
<xs:enumeration value="E"/>
<xs:enumeration value="Z"/>
<xs:enumeration value="Y"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="tValidityEnum">
<xs:restriction base="xs:Name">
<xs:enumeration value="good"/>
<xs:enumeration value="invalid"/>
<xs:enumeration value="questionable"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="tEventTypeEnum">
<xs:restriction base="xs:Name">
<xs:enumeration value="SysEvt">
<!--Power system event (fault detected; trip issued; breaker operation ...)-->
</xs:enumeration>
<xs:enumeration value="ComEvt">
<!--Communication event (association lost, buffered report overflow; ...)-->
</xs:enumeration>
</xs:restriction>
</xs:simpleType>
<xs:enumeration value="clockUnlockedWithinOneSec"/>
<xs:enumeration value="clockUnlockedWithin10Sec"/>
<xs:enumeration value="clockFailure"/>
</xs:restriction>
</xs:simpleType>

<xs:simpleType name="tLeapSecEnum">
  <xs:restriction base="xs:Name">
    <xs:enumeration value="noLeapSec"/>
    <xs:enumeration value="leapSecAdded"/>
    <xs:enumeration value="leapSecSubstracted"/>
    <xs:enumeration value="noLeapSecCapability"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tRecloserStatusEnum">
  <xs:restriction base="xs:Name">
    <xs:enumeration value="ready">
      <xs:annotation>
        <xs:documentation>Recloser is Ready: waiting for protection trigger</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="inProgress">
      <xs:annotation>
        <xs:documentation>Recloser is “In Progress” i.e., performing the number of predefined auto-reclose cycles</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="lockout">
      <xs:annotation>
        <xs:documentation>Recloser is locked out, e.g., if the number of trips exceeds the maximum number of reclose actions</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="blocked">
      <xs:annotation>
        <xs:documentation>Recloser is blocked, e.g., by a protection signal or by manual tripping</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tRecloserModeEnum">
  <xs:restriction base="xs:Name">
    <xs:enumeration value="disabled"/>
    <xs:enumeration value="singlePole"/>
    <xs:enumeration value="threePole"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tFaultTypeEnum">
  <xs:restriction base="xs:Name">
    <xs:enumeration value="AG"/>
    <xs:enumeration value="BG"/>
    <xs:enumeration value="CG"/>
    <xs:enumeration value="AB"/>
    <xs:enumeration value="BC"/>
    <xs:enumeration value="CA"/>
    <xs:enumeration value="ABG"/>
  </xs:restriction>
</xs:simpleType>
<xs:enumeration value="BCG"/>
<xs:enumeration value="CAG"/>
<xs:enumeration value="ABC"/>
<xs:enumeration value="unknown"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="tStandardizedFileExtensionEnum">
<xs:restriction base="xs:Name">
<xs:enumeration value="cid">
<xs:annotation>
<xs:documentation>Configured IED Description file as defined in IEC 61850-6</xs:documentation>
</xs:annotation>
</xs:enumeration>
<xs:enumeration value="scd">
<xs:annotation>
<xs:documentation>System Configuration Description file as defined in IEC 61850-6</xs:documentation>
</xs:annotation>
</xs:enumeration>
<xs:enumeration value="ssd">
<xs:annotation>
<xs:documentation>Substation Specification Description file as defined in IEC 61850-6</xs:documentation>
</xs:annotation>
</xs:enumeration>
<xs:enumeration value="cfg">
<xs:annotation>
<xs:documentation>COMTRADE Configuration file as defined in IEEE C37.111</xs:documentation>
</xs:annotation>
</xs:enumeration>
<xs:enumeration value="dat">
<xs:annotation>
<xs:documentation>COMTRADE Data file as defined in IEEE C37.111</xs:documentation>
</xs:annotation>
</xs:enumeration>
<xs:enumeration value="inf">
<xs:annotation>
<xs:documentation>COMTRADE Information file as defined in IEEE C37.111</xs:documentation>
</xs:annotation>
</xs:enumeration>
<xs:enumeration value="hdr">
<xs:annotation>
<xs:documentation>COMTRADE Header file as defined in IEEE C37.111</xs:documentation>
</xs:annotation>
</xs:enumeration>
<xs:enumeration value="cff">
<xs:annotation>
<xs:documentation>COMTRADE single file as defined in IEEE C37.111</xs:documentation>
</xs:annotation>
</xs:enumeration>
</xs:restriction>
</xs:simpleType>
A.4 The file H_SimpleTypes.xsd

<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Pierre Martin (HYDRO-QUEBEC) -->
<xs:schema xmlns=http://www.pes-psrc.org/Subcommittee/H/CommonTypes
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xmlns:xs=http://www.w3.org/2001/XMLSchema
targetNamespace=http://www.pes-psrc.org/Subcommittee/H/CommonTypes" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.0">
  <xs:simpleType name="tAnyName">
    <xs:restriction base="xs:normalizedString"/>
  </xs:simpleType>
  <xs:simpleType name="tCompanyIdentificationCode">
    <xs:restriction base="tAnyName">
      <xs:minLength value="2"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tFaultCorrelationEnum">
    <xs:restriction base="xs:Name">
      <xs:enumeration value="prefault"/>
      <xs:enumeration value="fault"/>
      <xs:enumeration value="postfault"/>
      <xs:enumeration value="none"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tClcMthEnum">
    <xs:restriction base="xs:Name">
      <xs:enumeration value="true_RMS"/>
      <xs:enumeration value="peak_Fundamental"/>
      <xs:enumeration value="rms_Fundamental"/>
      <xs:enumeration value="rms_2ndHarmonic"/>
      <xs:enumeration value="rms_3rdHarmonic"/>
      <xs:enumeration value="rms_4thHarmonic"/>
      <xs:enumeration value="rms_5thHarmonic"/>
      <xs:enumeration value="other"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tBreakerPositionEnum">
    <xs:restriction base="xs:Name">
      <xs:enumeration value="intermediate-state"/>
      <xs:enumeration value="open"/>
      <xs:enumeration value="closed"/>
      <xs:enumeration value="bad-state"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
<xs:maxLength value="4"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="tName">
  <xs:restriction base="tAnyName">
    <xs:minLength value="1"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tFaultClearingTimeInMilliSec">
  <xs:restriction base="xs:integer"/>
</xs:simpleType>
<xs:simpleType name="tFaultedCircuit">
  <xs:restriction base="tAnyName"/>
</xs:simpleType>
</xs:schema>
Annex B

(informative)

COMFEDE instance file example

B.1 The file COMFEDEExample.xml

The following is an example of a COMFEDE file containing two Log elements, each element representing an event record from a different location.

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <Log logName="" version="2010" revision="A" xmlns:MyExtension="www.company.com/COMFEDEExtension">
    <Location companyName="Donnacon Power" companyIdentificationCode="00" stationName="Cap Rouge" voltageLevelInkV="120" bayName="L1202" latitude="497671893" longitude="937582165" tmOfsTmm="-300">
      <sch:IEDNameplate name="21L1202A" vendor="vendorX" model="ExtremeModel" serNum="ABC123456789" hwRev="H23.r01" swRev="TheGoodOne">
      </sch:IEDNameplate>
    </Location>
    <Entry entryId="4294967292">
      <EntryData t="2005-03-11T14:37:12.345852Z" eventType="IEDEvt">
        <DataRef>DEF S LOADED</DataRef><Val/>
      </EntryData>
    </Entry>
    <Entry entryId="4294967293">
      <EntryData t="2005-03-11T14:37:22.667852Z" eventType="IEDEvt">
        <DataRef>Relay newly powered up or settings changed</DataRef><Val/>
      </EntryData>
    </Entry>
    <Entry entryId="4294967294">
      <EntryData t="2005-03-11T14:38:13.998852Z" validity="invalid" eventType="SysEvt">
        <DataRef desc="Recloser in L0ckout mode">79LO</DataRef><Val type="xs:boolean">true</Val>
      </EntryData>
    </Entry>
    <Entry entryId="4294967295">
      <EntryData t="2005-03-11T14:39:13.998852Z" eventType="SysEvt" faultTrigger="true">
        <DataRef T="BCG" D="T"></DataRef><Val/>
        <PayloadData payloadType="Line Protection" sGroup="1" faultType="AB" trip="true">
          <ComtradeFileName ext="cfg">050311,143912998,CapRouge_120kV_Line1,21L1202A,Don.cfg</ComtradeFileName>
          <ComtradeFileName ext="dat">050311,143912998,CapRouge_120kV_Line1,21L1202A,Don.dat</ComtradeFileName>
        </PayloadData>
      </EntryData>
    </Entry>
  </Log>
</COMFEDE>
```
<ElectMeas>
  <sch:ThreePhsCurr fltCorrel="prefault">
    <PhsA>
      <Mag unit="A">276</Mag>
      <Ang unit="deg">22.1</Ang>
    </PhsA>
    <PhsB>
      <Mag unit="A">262</Mag>
      <Ang unit="deg">-91.7</Ang>
    </PhsB>
    <PhsC>
      <Mag unit="A">246</Mag>
      <Ang unit="deg">138.2</Ang>
    </PhsC>
    <Neut>
      <Mag unit="A">65</Mag>
      <Ang unit="deg">5.1</Ang>
    </Neut>
  </sch:ThreePhsCurr>
  <Fq unit="Hz">60.01</Fq>
  <FltRpt>
    <sch:DistToFlt unit="m" mult="k">48.17</sch:DistToFlt>
  </FltRpt>
  <sch:FltCirc>L7023</sch:FltCirc>
</ElectMeas>

<ActTgtData>
  <sch:Tgt name="INST TIME ZONE_1 A"/>
  <sch:Tgt name="INST TIME ZONE_1 B"/>
</ActTgtData>

<RecloserRpt shotCnt1ph="0" shotCnt3ph="0" status="inProgress" mode="threePole">
  <!-- Mechanical breaker operate time-->
</RecloserRpt>

<BrkrRpt name="120-1" pos="open" operTmms="26" MyExtension:BkrOpTimeMech="28">
  <!-- Mechanical breaker operate time-->
</BrkrRpt>

</PayloadData>
</EntryData>
</Entry>

<SettingFileName ext="set">21L1202A</SettingFileName>
</Log>

<Log version="2010" revision="A">
  <Location companyName="Wendake Electric" companyIdentificationCode="00"
    stationName="Stoneham" voltageLevelInkV="120" bayName="L1202"
    tmOfsTmm="-300">
    <sch:IEDNameplate vendor="vendorY" name="21L1202A"/>
  </Location>
  <Entry entryId="3256">
    <EntryData t="2005-03-11T14:39:13.997Z" eventType="SysEvt" faultTrigger="true">
      <DataRef>BCG T</DataRef>
      <Val/>
      <PayloadData payloadType="Line Protection" sGroup="1" faultType="AB" trip="true">
<ComtradeFileName ext="cfg">050311,143912997,Stoneham,21L1202A,Wen.cfg</ComtradeFileName>
<ComtradeFileName ext="dat">050311,143912997,Stoneham,21L1202A,Wen.dat</ComtradeFileName>
<FltRpt>
  <DistanceToFault multiplier="k" unit="m">56.4</DistanceToFault>
  <FaultedCircuit>L1202</FaultedCircuit>
</FltRpt>
</PayloadData>
</EntryData>
</Entry>
<SettingFileName ext="set">21L1202A</SettingFileName>
</Log>

<Log logName="LDName" logRef="LDName" newEntr="4294967296" newEntrTm="2005-03-11T14:39:12.345852Z" oldEntr="4294967292" oldEntrTm="2005-03-11T14:37:12.345852Z" version="2010" revision="A">
  <Location companyName="Hydro-Quebec" stationName="Shannon" voltageLevelInkV="120" bayName="L1202" tmOfsTmm="-300">
    <sch:IEDNameplate name="21L1202" vendor="vendorZ"/>
  </Location>
  <Entry entryId="35925" timeOfEntry="2005-03-11T14:38:12.423852Z">
    <EntryData t="2005-03-11T14:38:12.423852Z" eventType="SysEvt" faultTrigger="true">
      <!--This is an example of an event recorded by an IEC 61850 device-->
      <DataRef funcName="Shannon_120kV_L1202/Z1PDIS1.Op" desc="Zone 1 Fault Operate (trip decision)">21L1202Prot/Z1PDIS1.Op</DataRef>
      <DA name="general" val="true" valType="xs:boolean"/>
      <DA name="phsA" val="true" valType="xs:boolean"/>
      <DA name="phsB" val="false" valType="xs:boolean"/>
      <DA name="phsC" val="false" valType="xs:boolean"/>
      <ReasonCode dchg="true" qchg="false"/>
    </EntryData>
    <EntryData t="2005-03-11T14:38:13.173852Z" eventType="SysEvt">
      <!--This is an example of an event recorded by an IEC 61850 device-->
      <DataRef funcName="Shannon_120kV_L1202/RREC1.Op" desc="Recloser Operate (closing command issued)">21L1202Prot/RREC1.Op</DataRef>
      <DA name="general" val="true" valType="xs:boolean"/>
      <DA name="phsA" val="true" valType="xs:boolean"/>
      <DA name="phsB" val="false" valType="xs:boolean"/>
      <DA name="phsC" val="false" valType="xs:boolean"/>
      <ReasonCode dchg="true" qchg="false"/>
    </EntryData>
  </Entry>
</Log>

<SettingFileName ext="set">21L1202A</SettingFileName>
</Log>
</COMFEDE>
Annex C

(informative)

Bibliography


\(^{9}\) IEC publications are available from the Sales Department of the International Electrotechnical Commission, Case Postale 131, 3 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iec.ch/). IEC publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

\(^{10}\) The IEEE standards or products referred to in this clause are trademarks owned by the Institute of Electrical and Electronics Engineers, Incorporated.

\(^{11}\) IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

\(^{12}\) http://www.w3.org/2001/tag/doc/versioning.

\(^{13}\) http://www.w3.org/TR/REC-xml-names/#dt-expname.