

C37.115™

IEEE Standard Test Method for Use in the Evaluation of Message Communications Between Intelligent Electronic Devices in an Integrated Substation Protection, Control, and Data Acquisition System

IEEE Power Engineering Society

Sponsored by the
Power System Relaying Committee



3 Park Avenue, New York, NY 10016-5997, USA

30 June 2004

Print: SH95208

PDF: SS95208



Documents

Document Sharing Hub

IEEE Standard Test Method for Use in the Evaluation of Message Communications between Intelligent Electronic Devices in an Integrated Substation Protection, Control, and Data Acquisition System

Sponsor

**Power System Relaying Committee
of the
IEEE Power Engineering Society**

Approved 10 November 2003

IEEE-SA Standards Board

Approved 3 May 2004

American National Standards Institute

Abstract: A standard test method, including evaluation criteria and performance measures, test scenarios for communication between intelligent electronic devices (IEDs) that implement substation integrated protection, control, and data acquisition is defined. Test scenarios are used to describe what data is exchanged between IEDs to perform a specified function that may be distributed between IEDs. All test scenarios use a standard Unified Modeling Language (UML) and a core reference model to build out an implementation.

Keywords: automatic control, communication, communication evaluation criteria, communication performance measures, data acquisition, distributed processing, IEDs, intelligent electronic devices, SCADA, supervisory control, test scenario, UML, Unified Modeling Language.

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2004 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 30 June 2004. Printed in the United States of America.

IEEE and 802 are registered trademarks in the U.S. Patent and Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Inc.

UCA[®] (Utility Communications Architecture) is a registered trademark of the Electric Power Research Institute (EPRI).

Print: ISBN 0-7381-3971-8 SH95208
PDF: ISBN 0-7381-3972-6 SS95208

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

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 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. When a document is more than five years old and has not been reaffirmed, 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 the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

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. 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. Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this 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 shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

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 the 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 C37.115-2003, IEEE Standard Test Method for Use in the Evaluation of Message Communications between Intelligent Electronic Devices in an Integrated Substation Protection, Control, and Data Acquisition System.)

This standard applies to systems used to communicate between intelligent electronic devices (IEDs) for substation integrated protection, control, and data acquisition. The requirements of this standard are in addition to those contained in standards for individual devices (e.g., relays, switchgear).

This standard applies to a rapidly changing technology. It is anticipated that frequent revision may be desirable.

Notice to users

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 this 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 shall not be responsible for identifying patents or patent applications for which a license may be required to implement an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Participants

At the time this standard was completed, Working Group H4, Standards for Evaluation of Message Communication between IEDs of the Relaying Communications H Subcommittee had the following membership:

Dennis K. Holstein, *Chair*

Mark Adamiak
Alex Apostolov
John Burger
Mason Clark
Douglas Dawson
George Gresko
Ameen Hamdon

Wayne Hartman
Jerry Hohn
Kazik Kuras
Jeff McElray
Gary Michel
Fredric Planchon
Jacques Prevost

Tevfik Sezi
Dilip Shroff
Charles Sufana
John T. Tengdin
Mike Thompson
Eric Udren
Murty Yalla

At the time this standard was completed, the Relaying Communications H Subcommittee had the following membership:

Ken Fodero, *Chair*

Mark Adamiak	William Lowe	Miriam Sanders
John Burger	Ken Martin	Chuck Shank
Paul Drum	Jeff McElray	Mark Simon
Jerry Hohn	Gary Michel	John T. Tengdin
Dennis K. Holstein	Brad Nelson	Eric Udren
Jim Huddleston	Tim Phillippe	Murty Yalla
Jim Ingleson	Roger Ray	John Zip
	Robert Ryan	

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

William Ackerman	Tony Giuliani	Arun Phadke
David Bassett	Daniel Gregory	Paul Pillitteri
Mike Bio	Erich Gunther	Barry Pokorney
Stuart Borlase	Roger Hedding	Peter Raschio
Stuart Bouchey	Jerry Hohn	Paulo Ribeiro
Gustavo Brunello	Dennis K. Holstein	Jeanjean Robert
John Burger	James Hrabliuk	James Ruggieri
James T. Carlo	Hermann Koch	Thomas Schossig
John Chadwick	Joseph L. Koepfinger	Robert Sodergren
Mason Clark	Terry Krummrey	Richard Taylor
Robert Dempsey	Robert Landman	John T. Tengdin
Guru Dutt Dhingra	Daniel Love	Demetrios Tziouvaras
Paul Drum	Greg Luri	Eric Udren
Gary R. Engmann	John McDonald	Benton Vandiver
Ernest Duckworth	Jeff McElray	James Wardin
Kenneth Fodero	Gary Michel	Peter Wong
James Gardner	Gary L. Nissen	Murty Yalla
Jeffrey Gilbert	Daniel Nordell	

When the IEEE-SA Standards Board approved this guide on 10 November 2003, it had the following membership:

Don Wright, *Chair*

Howard M. Frazier, *Vice Chair*

Judith Gorman, *Secretary*

H. Stephen Berger	Donald N. Heirman	Daleep C. Mohla
Joseph A. Bruder	Laura Hitchcock	William J. Moylan
Bob Davis	Richard H. Hulett	Paul Nikolich
Richard DeBlasio	Anant Kumar Jain	Gary S. Robinson
Julian Forster*	Lowell G. Johnson	Malcolm V. Thaden
Toshio Fukuda	Joseph L. Koepfinger*	Geoffrey O. Thompson
Arnold M. Greenspan	Thomas J. McGean	Doug Topping
Raymond Hapeman	Steve M. Mills	Howard L. Wolfman

*Member Emeritus

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

Alan Cookson, *NIST Representative*
Satish K. Aggarwal, *NRC Representative*

Savoula Amanatidis,
IEEE Standards Managing Editor

Contents

1. Overview.....	1
1.1 Scope.....	1
1.2 Purpose	1
1.3 Document organization	1
2. References	2
3. Definitions and acronyms	2
3.1 Definitions.....	2
3.2 Acronyms and abbreviations	12
4. Communication network structures.....	13
4.1 Substation LAN architecture.....	13
4.2 Utility enterprise WAN architecture.....	15
5. Network management and application development tools.....	17
5.1 Network analyzer.....	18
5.2 Network management stations	19
5.3 Monitoring software	19
5.4 Server-based databases.....	19
6. Communication test method.....	19
6.1 Functional and performance requirements.....	20
6.2 Test concepts and requirements.....	21
Annex A (informative) Bibliography	27
A.1 Modeling tools	27
A.2 Text books.....	27
A.3 EPRI reports	28
A.4 CIGRE reports.....	29
A.5 IEEE technical reports, papers, standards, and draft standards	29
A.6 IEC and ISO/IEC specifications	30
A.7 RFC specifications	30
Annex B (informative) Synch check to close breaker.....	31
B.1 Performance requirements	31

B.2 Evaluation criteria	31
B.3 Functional configuration	31
B.3.1 Overview	31
B.3.2 Communication configuration	32
B.3.3 Allocation of functions	35
B.4 Object model.....	35
B.4.1 Remote operation	35
B.4.2 Substation operation.....	37
B.5 Transaction sequences	38
B.5.1 Operator-initiated select for breaker close.....	39
B.5.2 Operator-initiated permissive close.....	42
B.5.3 Select synch check parameters, VTs, and SynchCheckRelay	42
B.5.4 Initiate synch check	44
B.5.5 Check for dead line or dead bus	44
B.5.6 Check for high-voltage difference.....	47
B.5.7 Perform synch check.....	47
B.5.8 Update RemoteController	49
Annex C (informative) Load tap changer control with remote voltage measurements	50
C.1 Performance requirements	50
C.2 Evaluation criteria	50
C.3 Functional configuration	50
C.4 Object model	51
C.4.1 Remote operations.....	52
C.4.2 Internal substation operation	54
C.5 Transaction sequences	54
C.5.1 System configuration engineer selects remote LoadCenterVT	54
C.5.2 Substation host initializes LoadCenterVT input to TCCR.....	54
C.5.3 System configuration engineer sets operating parameters.....	55
C.5.4 Tap changer control operations	55
C.5.5 Alarm generation.....	57
Annex D (informative) Distributed generation on utility feeders.....	61
D.1 Performance requirements.....	61
D.2 Functional configuration.....	61
D.3 Object model	62
D.3.1 Communication interface object model	62
D.3.2 Human interface components.....	62
D.3.3 Power system components	62
D.4 Transaction sequences.....	65
D.4.1 System configuration.....	65
D.4.2 System operation	68

List of figures

Figure 1—An example of substation LAN functional architecture	14
Figure 2—Extension of local area networks via bridges and gateways.....	15
Figure 3—Data exchange via a router.....	17
Figure 4—Network management and application development tools.....	18
Figure 5—Test configuration	22
Figure 6—Application-to-application communication times	23
Figure B.1—Functional configuration	33
Figure B.2—Remote operation for synch check scenario.....	36
Figure B.3—Voltage transformer object model	38
Figure B.4—Virtual device build-out for relay and breaker IEDs	39
Figure B.5—Substation LAN operation for synch check scenario	40
Figure B.6—Operator-initiated select for breaker close	41
Figure B.7—Operator-initiated permissive close.....	43
Figure B.8—Select synch check parameters and devices	44
Figure B.9—Initiate synch check	45
Figure B.10—Check for dead line or dead bus	46
Figure B.11—Report selected breaker state change	47
Figure B.12—Verify synch check conditions and close breaker	48
Figure B.13—Report synch check status and stop sending zero crossing data.....	49
Figure C.1—LTC control functional configuration.....	51
Figure C.2—LTC control object model.....	52
Figure C.3—Remote engineer control object model.....	53
Figure C.4—Load center VT object model	53
Figure C.5—Select load center VT for tap changer control.....	54
Figure C.6—Substation host initializes LoadCenterVT inputs to TapChangerControllerRelay	55
Figure C.7—Initialize LoadCenterVT to send voltage information	56
Figure C.8—Tap changer operation	57
Figure C.9—Alarm generated when measured voltage exceeds specified value	58
Figure C.10—Operator response to alarm.....	59
Figure C.11—Alarm generated when communication is lost	59
Figure C.12—Operator response to communication loss	60
Figure D.1—Distributed generation functional configuration.....	62
Figure D.2—Communication interface object model	63
Figure D.3—Human interface components	64
Figure D.4—Power system components	65
Figure D.5—Instantiation of breaker and relay components.....	66
Figure D.6—Configure U.OCR setting group.....	67
Figure D.7—Configure C.OCR reporting.....	69
Figure D.8—System operation to monitor U.GCB and U.FCB states.....	70
Figure D.9—System operation to trip C.MCB.....	71

IEEE Standard Test Method for Use in the Evaluation of Message Communications between Intelligent Electronic Devices in an Integrated Substation Protection, Control, and Data Acquisition System

1. Overview

1.1 Scope

This standard defines standard communication modeling, terminology, evaluation criteria, and performance measures for communication test scenarios, which specify messages to be exchanged between electrical power substation intelligent electronic devices (IEDs). These scenarios define message transactions between applications within the substation, and between substation IEDs and remotely located applications. The scenarios do not specify the communication protocol required to implement the transactions.

1.2 Purpose

There are currently no coherent communication modeling, terminology, and communication test scenarios for the evaluation of one or more implementation concepts for communication between substation IEDs within a substation, or between a substation and remote IEDs. Utilities and vendors will use this standard to evaluate, on a common basis, one or more implementation solutions.

1.3 Document organization

In addition to the overview, described in this clause, normative references are listed in Clause 2, and definitions and acronyms are listed in Clause 3.

Communication network structures are described in Clause 4 for the substation LAN architecture and for the utility enterprise WAN architecture. The WAN architecture is further described in terms

of the requirements for bridging communication networks, switching between communication networks, and routing between communication networks.

Network management and application development tools are described in Clause 5. These tools include the network analyzer, network management workstations, monitoring software, server-based databases, and test methods and measurements.

Communication test method is formally defined in Clause 6. This method is specified in terms of functional and performance requirements for conformance testing, interoperability testing, and operability testing. Test concepts are specified in terms of requirements for performance measures, evaluation requirements, procurement specification, and vendor requirements.

Annexes are included to add informative specifications. Annex A is a bibliography listing references that are commonly used in power system engineering and communications between IEDs; Annex B describes a synch check to close breaker; Annex C describes a load tap changer control with remote voltage measurements; and Annex D describes the distributed generation on utility feeders.

2. References

This standard shall be used in conjunction with the following publications. When the following standards are superseded by an approval revision, the revision shall apply unless noted otherwise.

IEEE Std 802.1D™, 1998 Edition (ISO/IEC 15802-3:1998), IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Networks—Common Specifications—Part 3: Media Access Control (MAC) Bridges.^{1,2}

IEEE Std 1588™-2002, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.

ISO/IEC 9646-7:1995, Information Technology—Open Systems Interconnection—Conformance Testing Methodology and Framework—Part 7: Implementation Conformance Statements.³

3. Definitions and acronyms

3.1 Definitions

Some of the terms used in technical descriptions in this standard are listed as follows. The Unified Modeling Language (UML) descriptions related to the Object Management Architecture (OMA) object model are identified by the designation [OMA].

¹The IEEE standards or products referred to in Clause 2 are trademarks owned by the Institute of Electrical and Electronics Engineers, Inc.

²IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, Piscataway, NJ 08854, USA (<http://www.standards.ieee.org/>).

³ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Geneva 20, Switzerland (<http://www.iso.ch/>). ISO/IEC publications are also available in the United States from Global Engineering Documents, 15 Inverness Way East, Englewood, CO 80112, USA (<http://www.global.ihs.com/>). Electronic copies are available in the United States from the American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (<http://www.ansi.org/>).

When brackets enclose a multiword term (one or more words), it indicates that those words are optional when referring to the term. For example, use case [class] may be referred to as use case.

The following conventions are used.

Contrast: <term>. Refers to a term that has an opposed or substantively different meaning.

See: <term>. Refers to a related term that has a similar, but not synonymous meaning.

Synonym: <term>. Indicates that the term has the same meaning as another term, which is referenced.

Acronym: <term>. This indicates that the term is an acronym. The reader is usually referred to the spelled out term for the definition, unless the spelled-out term is rarely used.

3.1.1 action: A computational or algorithmic procedure.

3.1.2 action state: A state with an internal action and one or more outgoing transitions involving the completion of an internal action.

3.1.3 active class: A class whose instances are active objects. *See:* active object.

3.1.4 active object: An object that owns a thread and can initiate control activity. An instance of active class. *See:* active class.

3.1.5 activity diagram: A special case of a state diagram in which all or most of the states are action states and in which all or most of the transitions are triggered by completion of actions in the source states. *Contrast:* state diagram.

3.1.6 actor [class]: A predefined stereotype of type denoting an entity outside the system that interacts with use cases.

3.1.7 addressing: Means to identify the source and sink (recipients) of all information transfers.

3.1.8 address resolution protocol (ARP): An address used alongside IP to discover IP-to-MAC address mappings.

NOTE—ARP is described in RFC 826 [B59].

3.1.9 adjacent substation protection: Protection of power system equipment at one substation based on data measured at others. Examples are line differential protection and teleprotection schemes.

3.1.10 agent: Servers that are designated to work with compatible client stubs known as *user agents*, which share the same server protocol. Agents are responsible for picking up and delivering messages between senders and receivers.

3.1.11 aggregate [class]: A class that represents the *whole* in an aggregation (whole-part) relationship. *See:* aggregation.

3.1.12 aggregation: A special form of association that specifies a whole-part relationship between the aggregate (whole) and a component part. *Contrast:* composition.

3.1.13 alarm processing: Alarm analysis procedures to improve presentation of alarm data. It ranges from updating alarm lists and producing group alarms up to more intelligent evaluations.

3.1.14 artifact: A piece of information that is used or produced by a software development process. An artifact can be a model, a description, or software.

3.1.15 association role: The roles that type or class plays in an association.

3.1.16 asynchronous message: A message where the sending objects do not pause to wait for results. *Synonym:* asynchronous request [OMA]. *Contrast:* synchronous message.

3.1.17 asynchronous transmission: In data communications, a method of transmission in which sending and receiving of data is controlled by control characters rather than by a timing sequence. *Contrast:* synchronous transmission.

3.1.18 attribute description:

- a) Text describing an object's attribute. It is a filled-in template of information about an attribute, including: description, legal values, unit of measure, required (yes/no) get/set constraints, rules for getting a default value, applicable states, and tractability codes.
- b) For real-time systems, categorize each attribute as: state, state-dependent, or state independent.

3.1.19 automatic switching sequences: Automatic sequential operation of groups of power system devices to reduce operator workload and/or switching time and to avoid unsuccessful or unnecessary switching attempts.

3.1.20 availability of data: State in which data are where the user needs them, when the user needs them, and how the user needs them. *See also:* transparent data access.

3.1.21 bandwidth (data): The rate at which a communications link is capable of carrying data, usually measured in bits per second (bps). *See also:* latency.

3.1.22 binary association: An association between two classes. A special case of an n -ary association.

3.1.23 binary large object: Very large (may be several megabytes in size) binary representation of an image data type.

3.1.24 Boolean expression: An expression that evaluates to a unicast value.

3.1.25 breaker: A device that connects and disconnects power circuits, with fault-interrupting capability. (Synonymous with circuit breaker.)

3.1.26 breaker (health) monitoring: A function that measure breaker's parameters mainly for maintenance purpose.

3.1.27 breaker failure protection: Backup protection scheme to trip all connected breakers if a breaker fails to clear a detected fault.

3.1.28 broadcast mode: Concurrent transfer mode of information to all connected receivers with one message from the information source. *Contrast:* unicast and multicast modes.

3.1.29 circuit breaker: *See:* breaker.

3.1.30 class diagram: A diagram that shows a collection of declarative (static) model elements such as classes, types, and their contents and relationships.

3.1.31 collaboration diagram: A diagram that shows object interactions organized around the objects and their links to each other. Unlike a sequence diagram, a collaboration diagram shows the relationships among the objects. Sequence diagrams and collaboration diagrams express similar information but show it in different ways. *See:* sequence diagram.

3.1.32 cold load pickup: Restoration of a circuit where all load diversity is lost.

3.1.33 communication association: In a deployment diagram, an association between two nodes that implies a communication. *See:* deployment diagram.

3.1.34 communication interface: Serial interface of a device that allows exchange of (physical and logical) information among devices of the same or different functional levels in a hierarchical system.

An interface specifies the connection of a communication link, with regard to the mechanical connection as well as to the signal's physical and functional characteristics.

3.1.35 communication safety: Measures and controls to avoid any deterioration or losses of information (reliability).

3.1.36 communication security: Measures and controls taken to deny unauthorized persons access to and information derived from communication facilities; and, to ensure the authenticity of communication transactions.

3.1.37 component: An executable software module with identity and a well-defined interface. *Contrast:* component [OMA].

3.1.38 component diagram: A diagram that shows the organizations and dependencies among components.

3.1.39 composite [class]: A class that is related to one or more classes by a composition relationship. *See:* composition.

3.1.40 composite aggregation: *Synonym:* composition.

3.1.41 composite state: A state that consists of substates. *Contrast:* substate.

3.1.42 composition: A form of aggregation with strong ownership and coincident lifetime as part of the whole. Parts with nonfixed multiplicity may be created after the composite itself, but once created they live and die with it (i.e., they share lifetimes). Such parts can also be removed before the death of a composite. Composition may be recursive. *Synonym:* composite aggregation.

3.1.43 concurrent substate: A substate that can be held simultaneously with other concurrent substates contained in the same composite state. *See:* composite state. *Contrast:* disjoint substate.

3.1.44 connect function: Process of initiating communication between two applications.

3.1.45 core object model: Term used to describe components of the reference object model that are used and reused to build out the object model for each virtual device.

3.1.46 container:

- a) An object that exists to contain other objects and that provides operations to access or iterate over its contents. For example, arrays, sets, dictionaries.
- b) A component that exists to contain other components.

3.1.47 delegation: The ability of an object to issue a message to another object in response to a message. Delegation can be used as an alternative to inheritance. *Contrast:* inheritance.

3.1.48 deployment diagram: A diagram that shows the configuration of run-time processing nodes and the components, processes, and objects that live on them. Components represent run-time manifestations of code units. *See:* component diagrams.

3.1.49 device: Physical entity connected to the communication network composed of at least one communication element (the network element), which may have a control element, and/or a monitoring element (transducer, actuator, etc.).

3.1.50 digital fault recorder (DFR): Device that samples and stores analog and related binary data during power system transients. *See:* transient fault recorder (TFR).

3.1.51 directory services:

- a) Service for resolving user identifications of network components to network routing information.
- b) Automated services permitting users to determine network address of data and processing resources via symbolic names.

- 3.1.52 disconnect function:** Process of terminating communication between two applications.
- 3.1.53 disjoint substate:** A substate that cannot be held simultaneously with other concurrent substates contained in the same composite state. *See:* composite state. *Contrast:* concurrent substate.
- 3.1.54 distribution tree:** A set of routers and subnetworks that allows a (set of) group member(s) to receive traffic from any source. Depending on the algorithm in use by the multicast routing protocol, the tree may be routed at the source or at some central point in the network.
- 3.1.55 dynamic classification:** A semantic variation of generalization in which an object may change type or role. *Contrast:* static classification.
- 3.1.56 equipment clock synchronization:** Automated procedure to maintain consistent time data throughout the substation or power system; e.g., for time tagging or synchronized sampling.
- 3.1.57 equipment load monitoring:** Automated procedures to detect equipment overload. The goal is to tolerate temporary overload, support for adaptive protection, and maintenance on request.
- 3.1.58 embedded system:** An embedded system includes a microprocessor, involves the immediate control of hardware, and is limited to performing a specific function.
- 3.1.59 export:** In the context of packages, to make an element visible outside its enclosing namespace. *See:* visibility. *Contrast:* export [OMA], import.
- 3.1.60 extends:** A relationship from one use case to another, specifying how the behavior defined for the first use case can be inserted into the behavior defined for the second use case.
- 3.1.61 fire:** To cause a state transition; for example, to fire off a communication message. *See:* transition.
- 3.1.62 firewall:** A device or combination of hardware and software used to enforce security.
- 3.1.63 focus of control:** A symbol on a sequence diagram that shows the period of time during which an object is performing an action, either directly or through a subordinate procedure.
- 3.1.64 function block:** An autonomous function, such as auto-reclosing, operational units, breaker failure protection, disturbance recording, etc., that might be implemented in separate hardware.
- 3.1.65 gen-spec structure:** Generalization-specialization is a relationship between classes. A specialization class inherits the responsibilities of its generalization classes.
- 3.1.66 guard condition:** A condition that must be satisfied in order to cause an associated transition to fire.
- 3.1.67 hard real-time:** A system is hard real-time if failure to respond to an event within a specified time is considered a system failure. A hard real-time system must make all its deadlines all the time. This requirement is a result of the fact that the correctness of the system is tied to the timeliness of its response.
- 3.1.68 high-speed sampled data:** Raw sampled data provided by an analog-to-digital converter located close to the VTs and CTs. A process bus, or dedicated point-to-point links, transfers such data samples from the converter to protection IEDs. (*Synonym:* streaming data).
- 3.1.69 high-speed yard data:** Raw sampled data provided by an analog-to-digital converter located in the substation yard close to the VTs and CTs. Such data samples are transferred from the converter to protection IEDs via dedicated point-to-point links, or over a process bus.
- 3.1.70 implementation inheritance:** The inheritance of the implementation of a more specific element. Includes inheritance of the interface. *Contrast:* interface inheritance.
- 3.1.71 import:** In the context of packages, a dependency that shows the packages whose classes may be referenced within a given package (including packages recursively embedded within it). *Contrast:* import [OMA] and export.

3.1.72 instance connection: An association in the problem domain that maps one object needs with other objects in order to fulfill its responsibilities. An instance object is shown as a line drawn between objects. The end-points of an instance connection are positioned between individual objects (rather than between classes). Each object has amounts or range markings on each of its instance connection, reflecting its constraints with other objects.

3.1.73 integrity: Immunity requirements to the network of data transfer errors due to accidental or intentional interference. Three levels are defined:

- **High:** where a vanishing small probability of undetected error must be achieved.
- **Medium:** where inherent data redundancy provides adequate error immunity.
- **Low:** where errors are merely a nuisance to the data recipient.

3.1.74 interaction: A behavioral specification that comprises a set of message exchanges among a set of objects within a particular context to accomplish a specific purpose. One or more scenarios may illustrate an interaction.

3.1.75 interchangeability: Two IEDs are interchangeable when one can replace the other without changing external functionality or performances.

3.1.76 interface inheritance: The inheritance of the interface of a more specific element. Does not include inheritance of the implementation. *Contrast:* implementation inheritance.

3.1.77 internet control message protocol (ICMP): An Internet Protocol used to report errors in processing datagrams.

NOTE—ICMP is specified in RFC 792 [B57]. ICMPv6 is specified in RFC 1885 [B62]. ICMPv6 incorporates many of the functions from ICMP for Ipv4, and it also includes IGMP's functionality. There is no separate protocol known as IGMPv6.

3.1.78 internet engineering task force (IETF): Publishes standards for Internet operation, including TCP/IP and SNMP.

NOTE—The IETF is not an accredited standards development organization, although it writes standards.

3.1.79 internet group management protocol (IGMP): An Internet Protocol used to keep neighboring multicast routers informed of the host group memberships present on a particular local network. IP hosts report their host group memberships to any immediately neighboring multicast routers use the IGMP. IGMP is an asymmetric protocol and is specified here from the point of view of a host, rather than a multicast router. (IGMP may also be used, symmetrically or asymmetrically, between multicast routers. Such use is not specified here.)

NOTE—IGMPv1 is specified in Appendix 1 of RFC 1112 [B60].

3.1.80 internet protocol (IP): A communication protocol used to move datagrams through an interconnected set of networks. Passing the datagrams from one internet module to another until the destination is reached does this. The internet modules reside in hosts and gateways in the internet system. The datagrams are routed from one internet module to another through individual networks based on the interpretation of an internet address.

NOTE—IP is defined in RFC 791 [B56]. Ipv6 is specified in RFC 1883 [B61].

3.1.81 interoperability testing: Check whether two or more communication network implementations operate together for some range of applications over a specific medium.

3.1.82 journaling: Means to maintain an audit trail of all control activities.

3.1.83 latency (communications): The delay between the time the data is sent from its origin and received at its destination. Latency determines how responsive the link will be. *See also:* bandwidth.

3.1.84 legacy systems: Information systems built on older technology that is not very adaptable or scalable to integrate new microprocessor-based intelligent electronic devices.

3.1.85 link role: An instance of an association role. *See:* association role.

3.1.86 master/slave: Communication management scheme called *polling* in which one IED (the Master) requests one IED, or group of IEDs (Slaves), to deliver specified information. Only Masters, not Slaves, may issue unsolicited data or commands. Used where data flows primarily between the Slaves and the Master.

3.1.87 metaclass: A class whose instances are classes. Metaclasses are typically used to construct metamodels.

3.1.88 meta-metamodel: A model that defines the language for expressing a metamodel. The relationship between a meta-metamodel and metamodel is analogous to the relationship between a metamodel and a model.

3.1.89 metamodel: A model that defines the language for expressing a model. An instance of a meta-metamodel.

3.1.90 metaobject: A generic term for all metaentities in a metamodeling language. For example, metatypes, metaclasses, metaattributes, and meta associations.

3.1.91 model aspect: A dimension of modeling that emphasizes particular qualities of the metamodel. For example, the structural model aspect emphasizes the structural qualities of the metamodel.

3.1.92 model element: An element that is an abstraction drawn from the system being modeled.

3.1.93 monitor function: Process of repeatedly comparing the current value of one variable with a predefined value, and reporting detected changes in the comparison state.

3.1.94 multicast forwarding: For each multicast IP packet received, a forwarding decision must be made. Typically, each packet must arrive on a specific upstream interface (i.e., incoming interface), and then must be copied onto a (set of) downstream *outgoing interfaces*.

3.1.95 multicast mode: Concurrent transfer mode of information to a predefined subset of all connected receivers with one message from the information source. *Contrast:* unicast and broadcast modes.

3.1.96 multicast routing: Activities performed by IP routers in order to determine how to forward multicast IP packets, either from some particular source to a group, or from any source to a group.

3.1.97 multiple classification: A semantic variation of generalization in which an object may belong to more than one class. *See:* dynamic classification.

3.1.98 multiple inheritance: A semantic variation of generalization in which a type may have more than one supertype. *Contrast:* single inheritance.

3.1.99 multiplicity: A specification of the range of allowable cardinalities that a set may assume. Multiplicity specifications may be given for roles within associations, parts within composites, repetitions, and other purposes. Essentially a multiplicity is a (possible infinite) subset of the nonnegative integers. *Contrast:* cardinality.

3.1.100 *n*-ary association: An association among three or more classes. Each instance of the association in an *n*-tuple of values from the respective classes. *Contrast:* binary association.

3.1.101 non-repudiation: Method by which the sender of data is provided with proof of delivery and the recipient is assured of the sender's identity, so that neither can later deny having processed the data.

3.1.102 object diagram: A diagram that encompasses objects and their relationships at a point in time. An object diagram may be considered a special case of a class diagram or a collaboration diagram. *See:* class, diagram, collaboration diagram.

3.1.103 object lifetime: A line in a sequence diagram that represents the existence of an object over a period of time. *See:* sequence diagram.

3.1.104 operability testing: Checks functionality within a communication network implementation, using an application to test in a multilayer environment.

3.1.105 persistent object: An object that exists after the process or threads that created it are terminated. For example, the data that persists beyond the lifetime of a single program execution is stored in a permanent data store.

3.1.106 portability: Ability to move, with minimal changes, application software between computers.

3.1.107 primitive type: A predefined basic type, such as an integer or a string.

3.1.108 privileged user: Subject that is granted special discretionary access privileges.

3.1.109 process bus: The serial bus closest to the process providing raw data or data processed very near to the switchgear or instrument transformers. In the area of process automation, very often called the *field bus*.

3.1.110 projection: A mapping from a set to a subset of it.

3.1.111 pseudo-state: A vertex in a state machine that has the form of a state but doesn't behave as a state. Pseudo-states include initial, final, and history connections.

3.1.112 qualifier: An association attribute or tuple of attributes whose values partition the set of objects related to an object across an association.

3.1.113 receive [a message]: The handling of a message passed from a sender object. *See:* sender, receiver.

3.1.114 receiver [object]: The object handling a message passed from a sender object. *Contrast:* sender.

3.1.115 reference: A denotation of a model element.

3.1.116 reference object model: Term used to describe an object model for substation integrated protection, control, and data acquisition. The reference object model provides abstract views of state-of-the-art IEDs that can be integrated over a communication network. The model provides the basis for developing a conceptual design that maps object attributes to a data structure and to communication protocol services. It also provides the basis for comparing conceptual designs against a standard reference.

3.1.117 refinement: A relationship that represents the fuller specification of something that has already been specified at a certain level of detail. For example, a design class is a refinement of an analysis class.

3.1.118 Request For Comments (RFC): A document that is treated as pseudo-standard but is subject to change before it is published as a standard.

3.1.119 reuse: The use of a pre-existing artifact.

3.1.120 security: Immunity of network resources to accidental or intentional unauthorized access. Three levels are defined:

- **High:** where access is limited to predefined and validated clients.
- **Medium:** where access is granted to any client meeting simple criteria.
- **Low:** where access (usually read-only) is granted to any client.

3.1.121 semantic variation: A particular interpretation choice for a semantic variation point. *See:* semantic variation point.

3.1.122 semantic variation point: A point of variation in the semantics of a metamodel. It provides an intentional degree of freedom for the interpretation of the metamodel semantics. *See:* semantic variation.

3.1.123 sequence diagram: A diagram that shows object interactions arranged in time sequence. In particular, it shows the objects participating in the interaction and the sequence of messages exchanged. Unlike a collaboration diagram, a sequence diagram includes time sequences but does not include object relationships. A sequence diagram can exist in a generic form (describes all possible scenarios) and in an instance form (describes one actual scenario). Sequence diagrams and collaboration diagrams express similar information, but shows it in different ways. *See:* collaboration diagram.

3.1.124 sequence of events (SOE) recorder: Device that samples and stores events like contact status changes, trips, limit violations, etc., for later play and analysis. The events are time tagged.

3.1.125 single inheritance: A semantic variation of generalization in which a type may have only one supertype. *Contrast:* multiple inheritances.

3.1.126 soft real-time: A system in which timeliness of response is important but not a matter of complete system failure. The acceptable frequency of missed deadlines is dictated by design.

3.1.127 static classification: Semantic variations of generalization in which an object may not change type or may not change role. *Contrast:* dynamic classification.

3.1.128 stereotype: A type of modeling element that extends the semantics of the metamodel. Stereotypes must be based on certain existing types or classes in the metamodel. Stereotypes may extend the semantics, but not the structure of pre-existing types and classes. Certain stereotypes are predefined in UML; the user may define others. Stereotypes are one of three extendibility mechanisms in the UML.

3.1.129 structural model aspect: A model aspect that emphasizes the structure of the objects in a system, including their types, classes, relationships, attributes, and operations.

3.1.130 substate: A state that is part of a composite state. A substate can either be a concurrent or disjoint substate. *See:* concurrent state, disjoint state.

3.1.131 supplier: Others can invoke a type, class, or component that provides services. *Synonym:* server object [OMA]. *Contrast:* client.

3.1.132 switch (in data network): Network (communication) device that filters, forwards, and floods frames based on the destination address of each frame. The switch operates generally at the data link layer (layer 2) of the OSI model. Switches provide a unique network segment on each port, thereby separating collision domain. They are more and more used to replace hubs (concentrators) to increase network performance and bandwidth. Some switches perform network switching (layer 3) in order to alleviate the burden on centralized routers.

NOTES

1—IEEE Std 802.1D-1998 defines the protocol used by layer 2 switches.

2—IEEE Std C37.115-2003 uses the name *bridge* in the for data network *switch*.

3.1.133 synchronous message: A message where the sending-object pauses to wait for results. *Synonym:* synchronous request [OMA]. *Contrast:* asynchronous message.

3.1.134 system reconfiguration:

- a) Procedure to manage changes in power system connectivity.
- b) Procedure to overcome failures in redundant secondary systems.

3.1.135 tagged value: The explicit definition of a property as a named-value pair. In a tagged value, the name is referred as the tag. Certain tags are predefined in the UML; the user may define others. *See:* constraint, stereotype.

3.1.136 time event: An event that occurs at a particular time. It may be expressed as a time expression. *See:* event.

3.1.137 time expression: An expression that resolves to an absolute or relative value of time.

3.1.138 time to live (TTL): A field in the IP packet that controls how far an IP packet may travel. Each router decrements a packet's TTL field by one when forwarding it. The largest possible TTL is 255. If a router ever receives a packet whose TTL equals 1, it cannot forward the packet any further.

3.1.139 timing mark: A denotation for the time at which an event or message occurs. Timing marks may be used in constraints.

3.1.140 transient object: An object that exists only during the execution of the process or threads that created it.

3.1.141 transmission control protocol (TCP): A reliable connection-oriented communication protocol.

NOTE—TCP is defined by RFC 793 [B58].

3.1.142 transparent data access: Access to data from any point of the system without knowing the details of the data link.

3.1.143 type expression: An expression that evaluates to a reference to one or more types.

3.1.144 unified modeling language (UML): The modeling language is the (mainly graphical) notation that methods use to express designs. UML rigorously defines the semantics of the object metamodel and provides a notation for capturing and communication object structure and behavior.

3.1.145 use case [class]: A class that defines a set of use-case instances.

3.1.146 use case diagram: A diagram that shows the relationships among actors and used cases within a system.

3.1.147 use case instance: A sequence of actions a system performs that yields an observable result of value to a particular actor. Usually scenarios illustrate prototypical use case instances. An instance of a use case class. *See:* use case class.

3.1.148 use case model: A model that describes a system's functional requirements in terms of use cases.

3.1.149 user datagram protocol (UDP): A connectionless, unreliable, transport layer network protocol for the exchange of requests and replies between connected stations.

NOTE—UDP defers reliability issues to the next higher layers: session, presentation, or application layers of the OSI Reference Model, application layer of the Internet Reference Model as defined in RFC 768 [B55].

3.1.150 uses: A relationship from a concrete use case to an abstract use case in which the behavior defined for the concrete use case employs the behavior defined for the abstract use case.

3.1.151 view element: A textual and/or graphical projection of a collection of model elements.

3.1.152 view projection: A projection of model elements onto view elements. A projection provides a location and a style for each view element.

3.1.153 virtual device: Class of power system devices.

3.1.154 virus: Self-replicating, malicious program segment that attaches itself to an application program or other executable system component.

3.1.155 vulnerability: Weakness in an information system, or cryptographic system, or components (e.g., system security procedures, hardware design, internal controls) that could be exploited.

3.1.156 whole-part structure: A structure that portrays a mapping between objects that takes on three basic forms: assembly-parts, container-contents, and collection-members. The notation is directional; so the structure can be drawn at any angle. The end-points are positioned to reflect a mapping between the objects. Each end of the whole-part structure is marked with an amount or range, indicating the number of parts that a whole may have.

3.2 Acronyms and abbreviations

The following is a list of acronyms and abbreviations used in the technical descriptions in this standard.

ARP	address resolution protocol
ASE	application service element
BFI	breaker failure initiate
BLOB	binary large object
CAS	common application service
CASM	common application service model
CID	connection identifier
CL	connectionless
CLTP	connectionless-mode transport service
CO	connection-oriented
COMTRADE	common format for transient data exchange
CSOM	client-server object model
DCR	directional comparison relaying
DFR	digital fault recorder
DNP	distributed network protocol
DTT	direct transfer trip
DUT	device under test
ESP	encapsulating security protocol
ETP	engineering test platform
GMT	Greenwich mean time
GOMSFE	generic object models for substation and feeder equipment
GOOSE	generic object-oriented substation event
GPS	global positioning system
GSP	generic service provider
ICMP	internet control message protocol
ID	identifier (i.e., name)
IDL	interface definition language
IED	intelligent electronic device
IETF	internet engineering task force
IGMP	internet group management protocol
IKE	internet key exchange
IP	internet protocol
IRIG	inter range instrument group
ISAKMP	internet security association and key management protocol
MMS	manufacturing message specification
MICS	model implementation conformance specification
ODBMS	object-oriented data base management system
OMA	object management architecture
OOA	object-oriented analysis
PACE	priority access control enabled
PIXIT	protocol implementation extra information for testing
PMU	phase measurement unit
PSOM	power system object model
QoS	quality of service
RBE	report by exception
RCB	report control block
RCL	relay control logic
RDBMS	relational data base management system
RFC	Request For Comment

RLC	reactive line component
RPC	remote procedure call
SBO	select-before-operate
SCADA	supervisory control and data acquisition
SNMP	simple network management protocol
TAL	time allowed to live
TCP	transmission control protocol
TFR	transient fault recorder
TOE	thread of execution ⁴
TTL	time to live
UCA [®]	utility communication architecture ⁵
UDP	user datagram protocol
UML	unified modeling language
VMD	virtual manufacturing device

4. Communication network structures

Several communication network structures are used in this standard. Near real-time information transfer between IEDs within a substation, over the substation Local Area Network (LAN), is one communication network structure. Another structure for control and monitoring uses the Utility enterprise Wide Area Network (WAN) for communication between the control center or technical services center and the substation. And a third structure includes the transfer of information between two utility communication networks or between the utility WAN and the customer's WAN. All three structures are used in the scenarios described in the annexes.

4.1 Substation LAN architecture

Figure 1 shows an example reference architecture, which is used in this standard for substation-distributed communication. A firewall is shown between the Substation router and the Substation LAN. A transceiver, connected to the Substation router, is included to transmit or receive data from either a satellite or radio. System interfaces to the Utility WAN are described in IEC 61968 [B52]⁶ and IEC 61970 [B53], which use a common interface reference model. This example concentrates on the architecture of the substation. Although not shown, routers and firewall should be used for the connection of the remote SCADA Master and the Engineering Work Station, and DMS and EMS systems to the WAN.

If required for precise time synch, an IRIG-B time generator and timing wire, or the use of IEEE Std 1588-2002, may be used to synchronize the IED clocks so that data from multiple sources can be combined for post-fault analysis. If less time synch precision is needed, time synchronization may be achieved by sending time synch messages over the substation LAN. Time synch precision requirements are defined in IEEE P1646 [B43].⁷

A high-speed database server connected to the LAN is used to maintain all substation configuration data, and to record all substation event data.

⁴An alternate definition for TOE used in communication networks is TCP Offload Engine.

⁵UCA[®] (Utility Communication Architecture) is a registered trademark of the Electric Power Research Institute (EPRI).

⁶The numbers in brackets correspond to those of the bibliography in Annex A.

⁷Numbers preceded by a P are IEEE authorized standards projects that were not approved by the IEEE-SA Standards Board at the time this publication went to press.

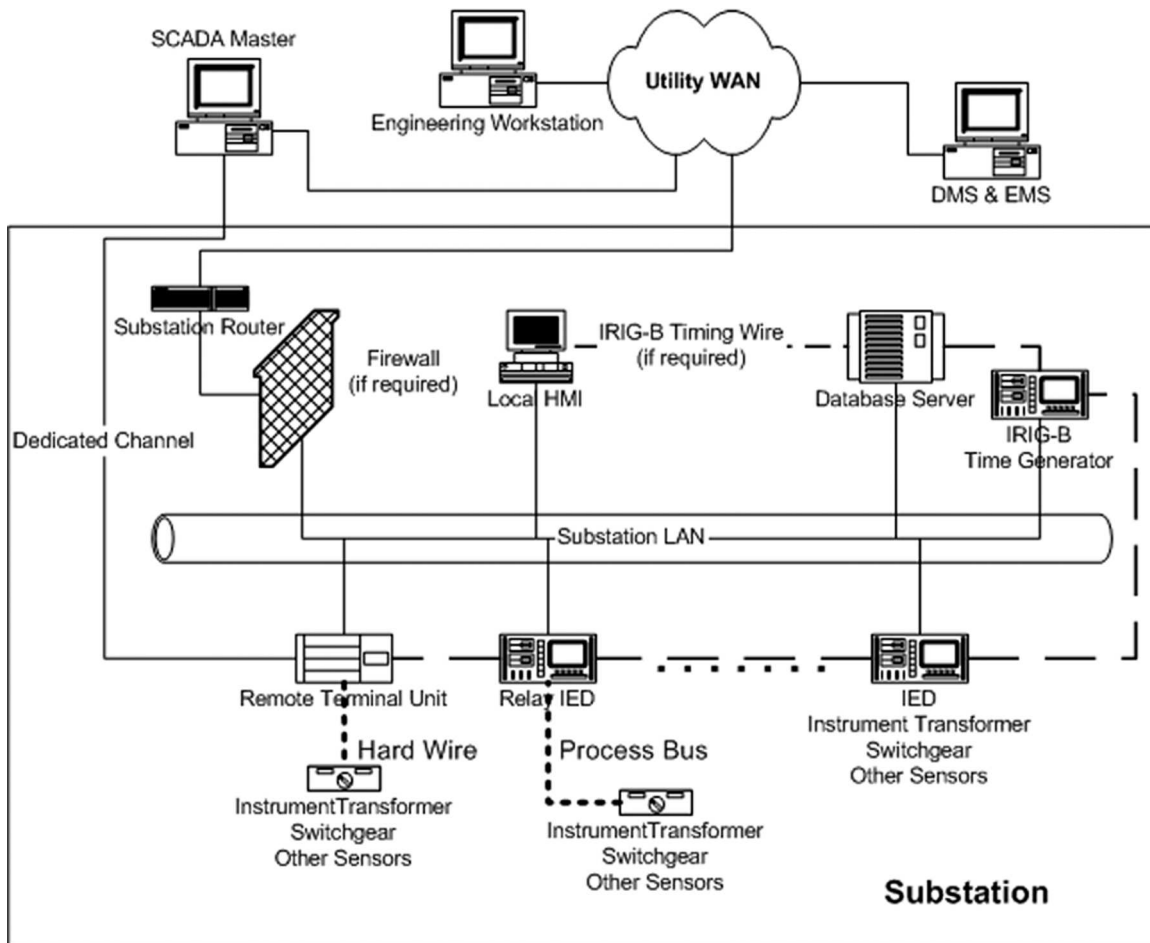


Figure 1—An example of substation LAN functional architecture⁸

Instrument transformers, switchgear and other sensors are connected by hardware to the IEDs or to a concentrator such as a Remote Terminal Unit (RTU), or through a process bus as described in IEC 61850 [B51]. Some instrument transformers, switchgear, and other sensors have IEDs that provide the capability to communicate over the LAN, which is extended from the control house into the switchyard. Figure 1 shows “...” to indicate that there may be many IEDs on the substation LAN.

Information security requires the use of a firewall, or other security provisions, to provide access protection to the substation IEDs. The firewall shown in Figure 1 provides a reasonable level of protection for all information passing through the substation router.⁹

However, a potential security problem exists through dedicated channels that by-pass the substation router. The dedicated channel, that is typical in today's substations, between the remote SCADA and the substation RTU is one example.

⁸The graphics for the Ethernet LANs are conceptual, not physical.

⁹A firewall may be implemented in a separate physical device, or it can be implemented within the substation router. If security is only required in one or two of the substation IEDs, such as a substation computer (host) or database server the firewall may be implemented on a board in the computer or database server that provides security to that device only.