INTERNATIONAL STANDARD

ISO/IEC/IEEE 18880

First edition
2015-04-15

Information technology — Ubiquitous green community control network protocol

Technologies de l'information — Protocole de contrôle de la communauté verte omniprésente

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Reference number
ISO/IEC/IEEE 18880:2015(E)

© IEEE 2015

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

COPYRIGHT PROTECTED DOCUMENT

© IEEE 2015

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from ISO, IEC or IEEE at the respective address below.

ISO copyright office
Case postale 56
CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
E-mail inmail@iec.ch
Web www.iec.ch

Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York
NY 10016-5997, USA
E-mail stds.ipr@ieee.org
Web www.ieee.org

Published in Switzerland

© IEEE 2015 – All rights reserved

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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.

The main task of ISO/IEC JTC 1 is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is called to the possibility that implementation of this standard may require the 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. ISO/IEEE is not responsible for identifying essential patents or patent claims for which a license may be required, for conducting inquiries into the legal validity or scope of patents or patent claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance or a Patent Statement and Licensing Declaration Form, 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 ISO or the IEEE Standards Association.

ISO/IEC/IEEE 18880 was prepared by the Corporate Advisory Group of the IEEE SA Board of Governors (as IEEE 1888:2014). It was adopted by Joint Technical Committee ISO/IEC JTC 1, *Information technology,* Subcommittee SC 6, *Telecommunications and information exchange between systems*, in parallel with its approval by the ISO/IEC national bodies, under the “fast-track procedure” defined in the Partner Standards Development Organization cooperation agreement between ISO and IEEE. IEEE is responsible for the maintenance of this document with participation and input from ISO/IEC national bodies.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

IEEE STANDARDS ASSOCIATION IEEE

IEEE Standard for Ubiquitous Green Community Control Network Protocol

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

IEEE-SA Board of Governors

Sponsored by the
Corporate Advisory Group

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

IEEE Std 1888™-2014
(Revision of
IEEE Std 1888-2011)

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

IEEE Std 1888™-2014
(Revision of
IEEE Std 1888-2011)

IEEE Standard for Ubiquitous Green Community Control Network Protocol

IEEE-SA Board of Governors

Sponsored by the
Corporate Advisory Group

Approved 27 March 2014

IEEE-SA Standards Board

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Abstract: The standard identifies gateways for field-bus networks, data storage for archiving and developing data sharing platforms, and application units as important system components for developing digital communities, i.e., building-scale and city-wide ubiquitous facility networking infrastructure. The standard defines a data exchange protocol that generalizes and interconnects these components (gateways, storage, application units) over the IPv4/v6-based networks. This enables integration of multiple facilities, data storage, application services such as central management, energy saving, environmental monitoring, and alarm notification systems.

Keywords: access interface, actuator, APP, application, communication protocol, component, data structure, energy management, energy saving, facility networking, gateway, GW, IEEE 1888™, point, registry, sensor, SOAP, storage

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

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

Copyright © 2014 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 28 May 2014. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

W3C is a trademark (registered in numerous countries) of the World Wide Web Consortium; marks of W3C are registered and held by its host institutions MIT, ERCIM, Keio, and Beihang.

PDF: ISBN 978-0-7381-9063-1 STD98619
Print: ISBN 978-0-7381-9064-8 STDPD98619

IEEE prohibits discrimination, harassment, and bullying.
For more information, visit http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html.
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 2015 – All rights reserved

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Standards Documents."

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE-SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, 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.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of an IEEE standard is wholly voluntary. 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.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own 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.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of 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 of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive 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 comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

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

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document 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

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted 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 these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, 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.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

Updating of IEEE Standards documents

Users of IEEE Standards documents 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.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, 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 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-SA Website at http://ieeexplore.ieee.org/xpl/standards.jsp or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at http://standards.ieee.org.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

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 by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. 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.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

Participants

At the time this IEEE standard was submitted to the IEEE-SA Standards Board for approval, the UGCCNet Working Group had the following membership:

Beijing Jiaotong University
BII Group Holdings Ltd.
China Telecommunications Corporation
Cisco Systems, Inc.
Intel Corporation
Qingdao Gaoxiao Information Industry Co. Ltd.
The University of Tokyo

The P1888 Working Group gratefully acknowledges the contributions of the following participants. Without their assistance and dedication, this standard would not have been completed.

Dong Liu, *Chair*

Hiroshi Esaki
Ming Feng
Shuai Gao
Chen Gu
Xiaochuan Gu
Lianshan Jiang
Wenjie Li
Hideya Ochiai
Shoichi Sakane
Yang Song
Guoquan Tan
Xiuying Tan
Hongke Zhang
Huiling Zhao
Xiaopeng Zhao
Ning Zou

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

Beijing Jiaotong University
Beijing University of Posts and Telecommunications (BUPT)
BII Group Holdings Ltd.
China Datang Corporation
China Telecommunications Corporation
Cisco Systems, Inc.
Guangzhou Smart Home Technology Standards Promotion Center
Guangzhou Video-star Electronics Industrial Co. Ltd.
Institute of Software, Chinese Academy of Sciences (CAS)
Marvell Semiconductor, Inc.
Nippon Telegraph and Telephone Corporation (NTT)
Qingdao Gaoxiao Information Industry Co. Ltd.
Raisecom Technology Co., Ltd.
Renesas Electronics Corporation
State Grid Corporation of China (SGCC)
The University of Tokyo
ZTE Corporation

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

When the IEEE-SA Standards Board approved this standard on 27 March 2014, it had the following membership:

John Kulick, *Chair*
Jon Walter Rosdahl, *Vice-chair*
Richard H. Hulett, *Past Chair*
Konstantinos Karachalios, *Secretary*

Peter Balma
Farooq Bari
Ted Burse
Clint Chaplain
Stephen Dukes
Jean-Phillippe Faure
Gary Hoffman
Michael Janezic
Jeffrey Katz
Joseph L. Koepfinger*
David J. Law
Hung Ling
Oleg Logvinov
Ted Olsen
Glenn Parsons
Ron Peterson
Adrian Stephens
Peter Sutherland
Yatin Trivedi
Phil Winston
Don Wright
Yu Yuan

*Member Emeritus

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

Richard DeBlasio, *DOE Representative*
Michael Janezic, *NIST Representative*

Don Messina
IEEE Standards Program Manager, Document Development

Joan Woolery
IEEE Standards Program Manager, Technical Program Development

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

Introduction

This introduction is not part of IEEE Std 1888-2014, IEEE Standard for Ubiquitous Green Community Control Network Protocol.

The standard identifies gateways for field-bus networks, data storage for archiving and developing data sharing platforms, and application units to be important system components for developing digital communities, i.e., building-scale and city-wide ubiquitous facility networking infrastructure. The standard defines a data exchange protocol that generalizes and interconnects these components (gateways, storage, application units) over the IPv4/v6-based networks. This enables integration of multiple facilities, data storage, application services such as central management, energy saving, environmental monitoring, and alarm notification systems.

Facility networking in buildings, houses, and factories is now considered to be a promising tool for energy management or energy saving, and networking of facilities with TCP/IP protocols has certainly enabled building-scale or city-wide energy management. However, most of the systems are proprietarily and independently developed, deployed, and operated, which makes the installation and running costs quite high.

Traditionally, in order to extend access reachability to sensors and actuators at the field-bus level via the Internet, gateway design has been introduced. However, recent applications of facility networking for such scales are required beyond just a simple access to devices. In most of the practical implementations, they have (1) large storage to archive the history of sensor readings, (2) user interface for interactive operation, (3) reporting systems, and (4) a data analyzer.

Collaboration of these system components is mandatory, especially in energy-aware facility networking. However, they cannot simply collaborate or interoperate with each other without understanding the intended analysis, integration, and operation of systems, because these system components have been independently developed and proprietarily integrated.

Interoperability of these system components by a common communication protocol certainly increases the efficiency of facility networking deployment. It reduces the cost of system integration and interoperability management, allowing installation of them in small and medium-sized buildings and even houses. For vendors, their developed components can be sold worldwide without any customized implementation, sometimes resulting in mass production with reasonable cost.

Targeting building-scale and city-wide energy management, the IEEE P1888 Working Group initiated the project named Ubiquitous Green Community Control Network (UGCCNet), which specifies integration architecture of remote facilities. The scope and purpose of this project is to establish facility networking infrastructure over the Internet by specifying an interoperable communication protocol among the common (building-scale) facility networking components (i.e., device access gateways, data storage, and application units). This standard specifies UGCCNet in order to allow interoperability and open development of those facility networking components. First, the standard generalizes all the facility networking components by a simple component model. Then, the standard defines communication protocol among them. The standard also introduces a registry mechanism to support autonomous collaboration of these components.

It is a communication infrastructure that aims to construct a new network for the renewal of the facilities, next generation's facility management, and the energy conservation including small and medium-sized facilities. The aspect is expanded from a past facility management to the operation management that targets energy conservation and the integration of the management platform. This infrastructure will be used for some system-level collaborations in addition to the energy conservation.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

Contents

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

IEEE Standard for Ubiquitous Green Community Control Network Protocol

IMPORTANT NOTICE: IEEE Standards documents are not intended to ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

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.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

1. Overview

1.1 Scope

The standard identifies gateways for field-bus networks, data storage for archiving and developing data sharing platforms, and application units such as those providing user interfaces of analysis and for knowing that environmental information is an important system component for developing digital communities, i.e., building-scale and city-wide ubiquitous facility networking infrastructure. The standard defines a data exchange protocol that generalizes and interconnects these components (gateways, storage, application units) over the IPv4/v6-based networks. This opens the application interface to handle the statuses of multi-vendor facilities on a generalized digital infrastructure. The standard assumes distributed operation of the infrastructure by multiple service providers and integrators, and defines a component management protocol that autonomously interoperates such distributed infrastructure. Security requirements are taken into consideration in this standard to ensure the integrity and confidentiality of data.

1.2 Purpose

The standard aims to develop green communities whose energy usage is well managed and highly efficient, by allowing the interconnection of facilities of multiple buildings including small and medium-sized on different converged networks, data sharing platforms, and application units. The products based on the standard implement the functions of sensing, archiving, sharing and presenting of ubiquitous information such as power usage and generation, environmental status and signals, human behavior, heating, ventilation, and air conditioning (HVAC) working status, lighting systems, weather, warnings, analyzed data, forecasted data, and so on. With the capability of interoperable integration of these system

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

components, the standard aims to provide and share rich data platforms, including cooperation with control-oriented 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.

IETF RFC 793, Transmission Control Protocol (TCP), Information Sciences Institute.[1]

IETF RFC 2460, Internet Protocol, Version 6 (IPv6) Specification, S. Deering and R. Hinden, editors.

IETF RFC 3986, Uniform Resource Identifiers (URI): Generic Syntax, T. Berners-Lee, R. Fielding, and L. Masinter, editors.

ISO 8601, Data elements and interchange formats—Information interchange—Representation of dates and times.[2]

W3C®, Extensible Markup Language (XML) 1.0, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and François Yergeau, editors.[3, 4]

W3C, Namespaces in XML 1.1, Tim Bray, Dave Hollander, Andrew Layman, Richard Tobin, editors.

W3C, SOAP Version 1.2 Part 1: Messaging Framework, Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen, Anish Karmarkar, and Yves Lafon, editors.

W3C, XML Schema Part 1: Structures, David Beech, Murray Maloney, Henry S. Thompson and Noah Mendelsohn, editors.

W3C, XML Schema Part 2: Datatypes, Ashok Malhotra and Paul V. Biron, editors.

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause. [5]

access control: The means to allow authorized entry and usage of resources.

actuator: A transducer that accepts a data sample or samples and converts them into physical action.

[1] IETF documents (i.e., RFCs) are available for download at http://www.rfc-archive.org/.
[2] ISO publications are available from the ISO Central Secretariat (http://www.iso.org/). ISO publications are also available in the United States from the American National Standards Institute (http://www.ansi.org/).
[3] W3C publications are available from the World Wide Web Consortium (W3C) at http://www.w3.org/.
[4] W3C® is a trademark (registered in numerous countries) of the World Wide Web Consortium; marks of W3C are registered and held by its host institutions MIT, ERCIM, Keio, and Beihang.
[5] *IEEE Standards Dictionary Online* subscription is available at:
http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html.

Copyright © 2014 IEEE. All rights reserved.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

eXtensible Markup Language (XML) namespace: A method for distinguishing XML elements and attributes that may have the same name but different meanings. A URL is used as a prefix to a "local name." This combination ensures the uniqueness of the element or attribute name. The URL is used only as a way to create a unique prefix and does not have to resolve to a real page on the Internet.

sensor: A transducer that converts a physical, biological, or chemical parameter into a digital signal.

universally unique identifier (UUID): An identifier that has a unique value within some defined universe.

NOTE—In this standard, the query-expression and lookup-expression of transport data structure has a UUID unless otherwise stated.[6]

3.2 Acronyms and abbreviations

AAA	authentication, authorization, and accounting
API	application programming interface
EPR	end-point reference
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
IPv6	Internet Protocol Version 6
NAT	Network Address Translation
RPC	remote procedure call
SIP	Session Initiation Protocol
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TTL	time to live
UGCCNet	Ubiquitous Green Community Control Network Protocol
URI	Uniform Resource Identifier
UUID	Universally Unique Identifier
VPN	Virtual Private Network
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
XML	eXtensible Markup Language

[6] Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

Copyright © 2014 IEEE. All rights reserved.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

4. Architecture

4.1 UGCCNet networking architecture

This protocol specification applies to a TCP/IP-based facility networking architecture shown in Figure 1. One of the main goals of this specification is to enable interoperability among facility networking components. Thus gateway (GW), storage, and application (APP), what we call "Component" in this document, have the same generalized communication interface. Registry has a different communication interface with these components, so it is distinguished with a dashed frame as shown in Figure 1. A Component works as a part of the data plane, and Registry works as a part of control plane. In Figure 1, an emphasis is on authentication, authorization, and accounting (AAA) functionality provided for operational management support. This document does not cover AAA, but related work may be taken on in the future.

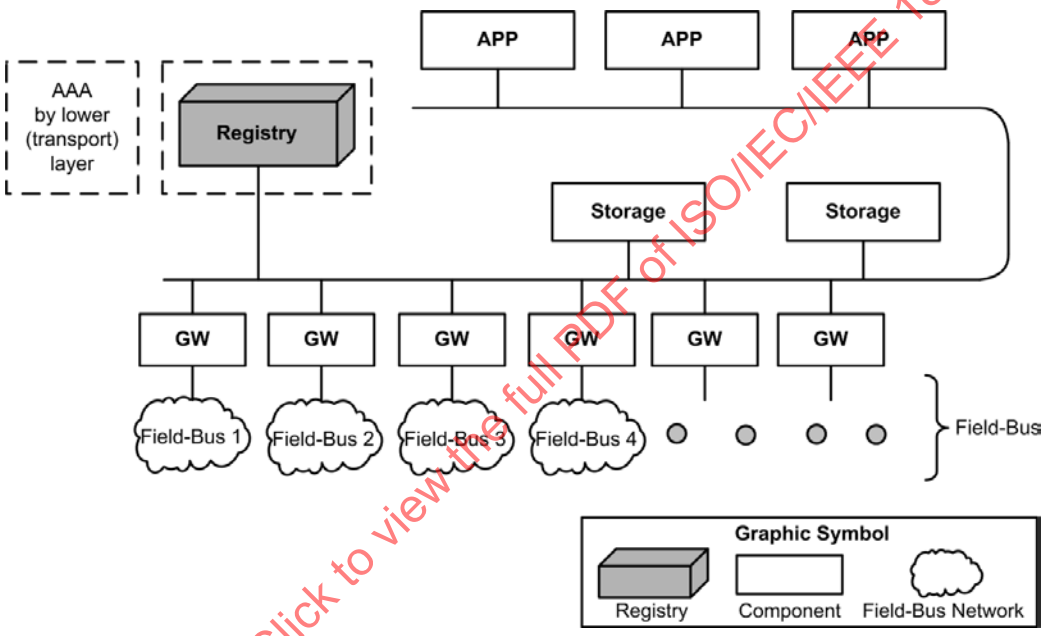

Figure 1—UGCCNet networking architecture

In the UGCCNet networking environment, Registry works as a broker of Components. It manages meta information, e.g., the role of each component and the semantics of Point ID, in order to bind components appropriately and autonomously. We describe them here in more detail and show how they collaborate with each other.

4.1.1 Gateway (GW)

The gateway component has physical sensors and actuators. It generalizes the data model and the access method for those devices, encapsulating each physical (field-bus) data model and access method. It acts on its actuator according to the written value from a component (e.g., APP), and it provides physical sensor readings for other components (e.g., storage and APP).

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

4.1.2 Storage

Storage component archives the history of data sequences. The written values from other components should be permanently stored in the backend disks. It provides the archived values to the components that have requested them.

4.1.3 Application (APP)

APP component provides some particular works on sensor readings and actuator commands. It can have user interface to display the latest environmental state. It can also allow a user to input some schedules of actuator settings, and it can as well analyze some sensor data in real time and provide the result as a virtual device.

4.1.4 Registry

The Registry works as a broker of GW, storage, and APPs. The main role of registry is to bind those components appropriately and autonomously. It is separated from the data-plane. It does not work on sensor readings or actuator settings directly. The UGCCNet should allow system operation without Registry.

4.2 Typical sequence

Figure 2 shows typical protocol sequences among components and registry. The bold arrows indicate component-to-component communication, and the dashed arrows indicate component-to-registry communication.

Registry manages the information of Point IDs and the role of components. When components need to access the data or the interface bound to a certain Point ID, it first refers to the Registry. Then, the Registry replies with the access URIs [e.g., SOAP end-point references (EPRs)] of the components that manage the data or the interface of the Point ID. Interaction with the Registry is optional. If the requesting component knows the target access URIs, for example by configuration, it is not necessary to consult to Registry.

Here, we describe the details of communication from case A to case I.

NOTE—Annex A provides more detailed explanations for each case.

Case A: Registration of Components. The registration includes the information of (1) what Point IDs this GW has, (2) what Point IDs this storage manages, (3) what Point IDs this APP reads and provides. See 5.3.2 for details.

Case B: Searching appropriate Points by semantic-query. See 5.3.3 for details.

Case C: Searching a Storage component for a specified Point. It returns the access URI (see IETF RFC 3986) for the resolved component. [7] See 5.3.3 for details.

Case D: Transmission of data; GW is sending observed sensor readings to Storage. See 5.2.3 for details.

Case E: Sequential retrieval of data; APP is retrieving data from Storage. If the retrieving dataset is large, data should be read sequentially. See 5.2.2 for details.

[7] Information on references can be found in Clause 2.

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

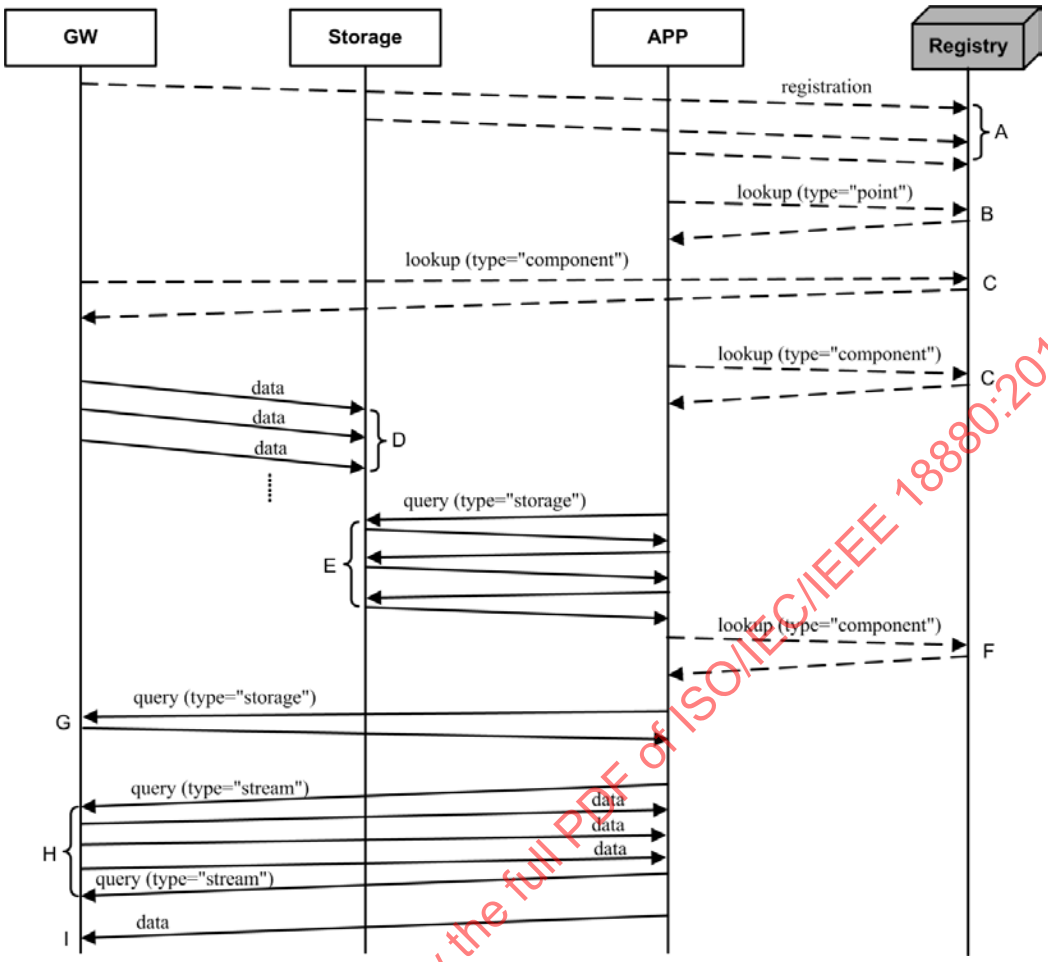

Figure 2—A typical sequence of UGCCNet communication

Case F: Searching a GW component for a specified Point. It returns the access URI for the resolved component. See 5.3.3 for details.

Case G: Retrieval of Data; APP is retrieving data from GW. If the retrieving dataset is small, data can be read in one remote procedure call (RPC). See 5.2.2 for details.

Case H: Notification of Data; APP is putting event query to GW periodically, and the GW submits updates to the APP. See 5.2.4 for details.

Case I: Writing of Data; APP is setting a command to a GW actuator. See 5.2.3 for details.

4.3 Concerns of UGCCNet network design

As we see in the preceding sequences, all components can behave both as the TCP (see IETF RFC 793) initiator and receiver at once. It implies the components should be put on a flat network. A flat network means there are no middle boxes that could disturb bidirectional communications such as NAT routers and firewalls.

To avoid the issue, we strongly recommend building an IPv6 (see IETF RFC 2460) network. There could be other solutions than IPv6 such as http proxy based or NAT traversal solutions. However, they would depend on the requirements of the network configuration. This specification does not reject such solutions,

Copyright © 2014 IEEE. All rights reserved.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

but they are required to be made interoperable with other UGCCNet-based systems and not to disturb the specification. Proposing specific solutions other than IPv6 is out of scope of this document.

4.4 System model and deployment

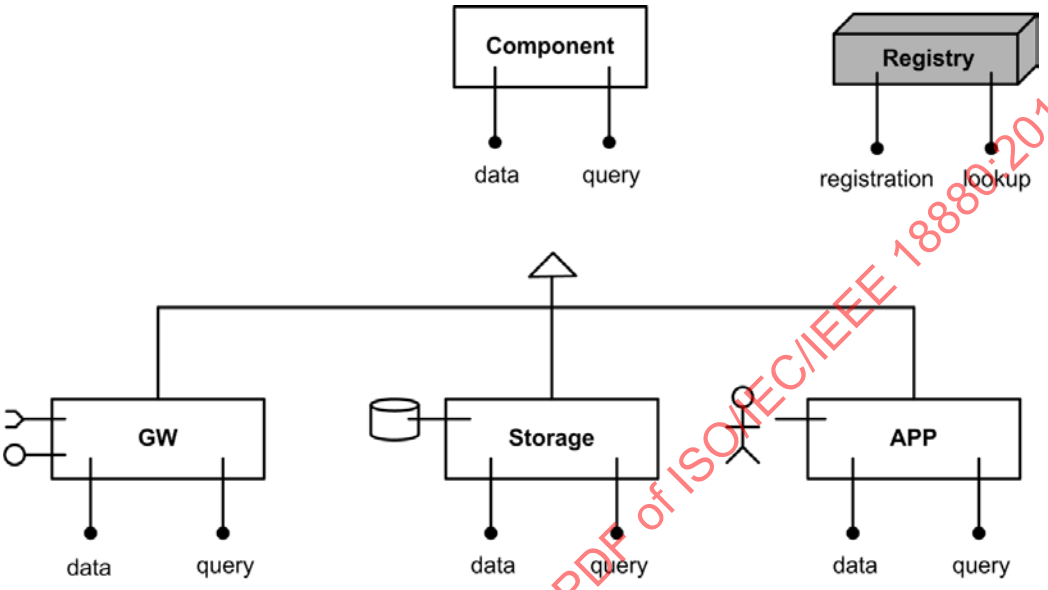

Figure 3—UGCCNet system model

Figure 3 provides the UGCCNet system model. Component is the basic unit for all the GWs, storage, and APPs. The interface of Component provides data and query method. GW, storage, and APP are the inherited classes of Component. Thus, they have the same interface (i.e., data and query method), and they communicate with each other using the same protocol, as follows:

— Query is a method for retrieving data (including event-based data transfer) from Component.

— Data is a method for pushing data into Component.

Registry works as a broker of Components with another type of interface (i.e., registration and lookup method). The interface of Registry provides registration and lookup method, as follows:

— Registration is a method for registering the role of components and semantics of Points.

— Lookup is a method for finding appropriate components and Points.

Typical implementations for GWs, storages, APPs, and Registry would be the following:

— GW implementations encapsulate field-buses and provide INPUT/OUTPUT access for physical devices (by query and data method).

— Storage implementations archive the history of data posted by data method and provide the historical data by query method.

— APP implementations provide other functionalities. For example, they can have user interface. Data processing component must be also categorized to an APP implementation.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

— Registry implementations manage the relationships between Point ID and components, provide registration of the role of components and semantic of Points by registration method, and provide inquiry of appropriate components and Points by lookup method.

Note that this specification also allows calling APP the system that accesses other components but does not have a query and data method.

This generalization enables open development of facility networking components (i.e., GWs, Storages and APPs) by any vendors. And we would deploy facility networking systems for customer buildings without customized programming, by binding these developed components (Figure 4).

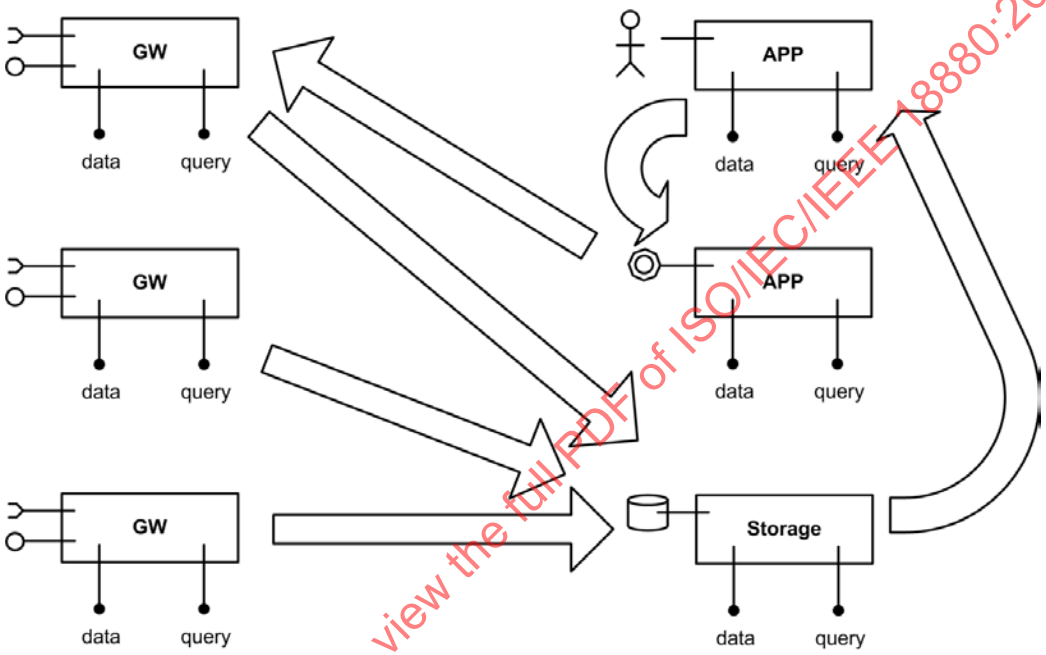

Figure 4—Implementation of a facility networking system

The role of Registry is to increase the autonomy of component-to-component collaboration. It allows autonomous collaboration of components, by sharing the information of component roles in an operational domain (in fact, not only in an operational domain but also with other external domains) (Figure 5).

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

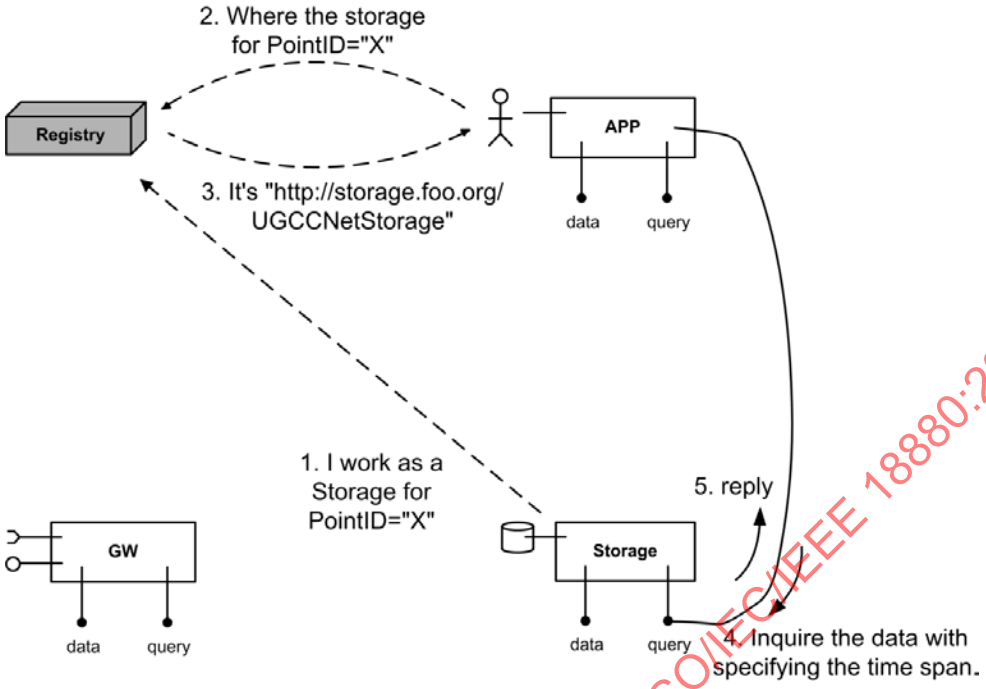

Figure 5—Autonomous collaboration of components

4.5 Point

4.5.1 Introduction

This subclause introduces the concept of Point. A Point shall have an URI-based globally unique identifier. It identifies a dataflow that exchanges data (i.e., sensor readings, actuator commands, and meta-control signals) among components.

4.5.2 Definition

A Point is an elemental message channel for a specific data sequence among Components. A sequence of sensor readings, actuator commands, and others (e.g., virtualized sensor readings, meta-control signals) shall be bound to a Point. We denote a message in a Point (whether it is coming from a sensor, or it is outgoing to an actuator) by value. Any object type is allowed for values in a Point.

Delivery of values among components shall be made by invoking other components' interface. The provided methods are as follows:

— Query: to read objects from specified Points

— Data: to write objects into specified Points

By using these methods, a component can get data of the specified Points from another component, and it can also transfer data to another component with specification of the Points.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

NOTE—This specification extends the traditional concept of Point in facility networking. Traditionally, “point” originally stood for a specific device to enable direct access (by read and write method). This definition is still true when the target component is a gateway. However, in this specification, Storage and APPs have the same interface as GWs, so we extended the definition not only to access them but also to manage the data sequences related to them. In writing a value to a Point, if the component has a GW implementation, the associated physical actuator would work according to the written value. If the component has a Storage implementation, the value would be archived in its disk.

4.5.3 URI-based identification

A Point is associated to a globally unique data sequence. The data shall have been generated from a specific sensor or to a specific actuator in the world. Thus, in order to identify the data sequence globally, each Point should have a globally unique identifier. Note that for private operation, it does not necessarily need to be globally unique. However, it is not recommended.

In UGCCNet, every Point shall have a URI for its identifier. Practically, we will first assign IDs for physical sensors and actuators, and then we will use the IDs for the Point IDs. This operation goes well with the traditional facility networking operation.

Taking URI for identifiers enables global access (if the Point is public) to the Point. Let X(=http://gw.foo.org/sensor1) be a Point ID.

If components do not know the registry server that manages the Point ID (X), they should try to access X directly. Then, the URI can redirect to the registry server. If components already know the registry server for X, Point ID may not need to be reachable. However, in order to obtain operational consistency, the host of the URI should be the host name of the GW (because physical sensors and actuators are attached to the GW). Thus, typical URI format should be:

```
point ID = "http://<GW host name>/<any format to identify the Point in
the GW>"
```

4.5.4 PointSet

This specification also defines PointSet to enable hierarchical management of Points. A PointSet aggregates multiple Points and multiple PointSets. This definition allows the conventional operation of grouping of Points hierarchically. However, the PointSet feature is optional. All the components should allow operation without PointSet. See 8.3.2.5 for formal definition.

5. Common communication protocol

5.1 General

This specification defines two types of communication protocols for components and registry, including the component-to-component communication protocol and component-to-registry communication protocol. The protocol message for component-to-component and component-to-registry communication is intended to use Simple Object Access Protocol (SOAP) (see W3C, SOAP Version 1.2 Part 1: Messaging Framework).

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

5.2 Component-to-component communication protocol

5.2.1 Types of component-to-component communication protocol

This subclause specifies and describes the following three types of sub-protocols for component-to-component communication. Note that instances of components are GWs, Storages, and APPs. As for the accessing methods to a registry, see 5.3.

— FETCH protocol—for data retrieval from a remote component.

— WRITE protocol—for data transfer to a remote component.

— TRAP protocol—for event query registration and event data transfer.

The latter part of this subclause describes these protocols in detail.

5.2.2 FETCH protocol

FETCH is a protocol for data retrieval from a remote component. We denote here the component that inquires data from the remote component by Requester, and the component that replies with the data by Provider. Figure 6 provides a diagram of the interaction.

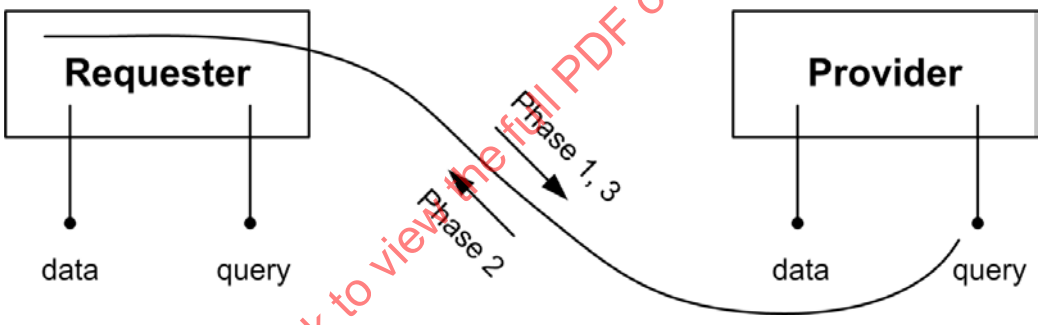

Figure 6—FETCH protocol

Phase 1: The Requester invokes the query method of its Provider. The Requester shall send query information (i.e., the range of interested dataset), and should send the size of acceptable dataset at the RPC-response by the number of value element, at the same time. If the acceptable size is not specified, the server should use 100 for it as the default setting.

Phase 2: The Provider returns a subset of the whole dataset by the RPC-response. If the size of the returning dataset is exceeding the acceptable data size of requester, or if it is consuming too much computing resources at the provider, the provider returns only a subset of the whole dataset. For succeeding data retrieval, the provider also returns a cursor associated to it.

Phase 3: If there is a cursor in the previous response, the Requester invokes the query method at the Provider again (with the given cursor). And, go to Phase 2.

Phase 4: If not, all the dataset has been retrieved and the FETCH procedure shall be completed.

Note that the cursor shall have moderate validity time, and the provider shall return Invalid Cursor Error if it received a query with an expired cursor. We recommend 60 s for cursor validity time. However, if the network provides narrow throughput, the cursor validity should have a longer time because each individual RPC takes more time.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

Note that the Provider returns the information of an error if any was encountered, e.g., access control policy, malformed-XML, or other system error.

5.2.3 WRITE protocol

WRITE is a protocol for data transfer to a remote component. We denote the component that submits data to the remote component by Requester, and the component that receives the data by Target. Figure 7 provides a diagram of the interaction.

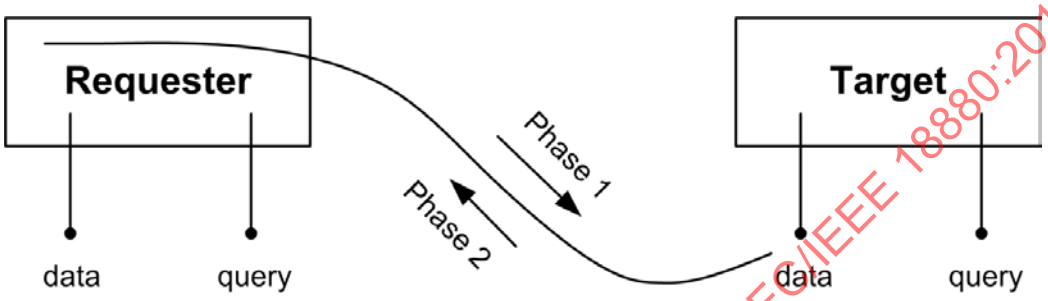

Figure 7—WRITE protocol

Phase 1: Requester invokes the data method of the Target with data contents.

Phase 2: Target replies whether it was successful or ended with failure to the Requester.

5.2.4 TRAP protocol

TRAP is a protocol for event query registration and event data transfer. The components are named in the following manner:

- Requester—the component that sets event-based query to Provider
- Provider—the component that transmits data when it has received query-matching updates
- Callback (Data)—the components that receives data from the Provider
- Callback (Control)—the components that receives control signals from the Provider

This subclause provides the definition of collaboration among these components (Figure 8). Though the roles are explicitly categorized in general, in most of the practical systems, Callback (Data), Callback (Control), and Requester will be the same component (Figure 9).

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

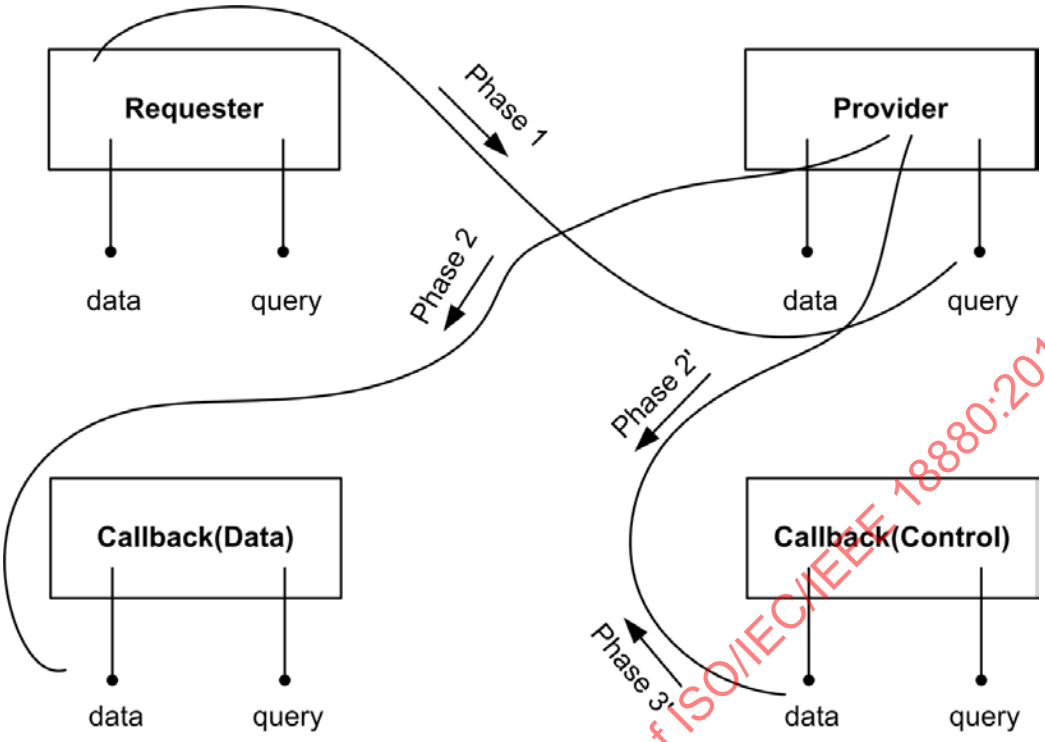

Figure 8—TRAP protocol (general implementation)

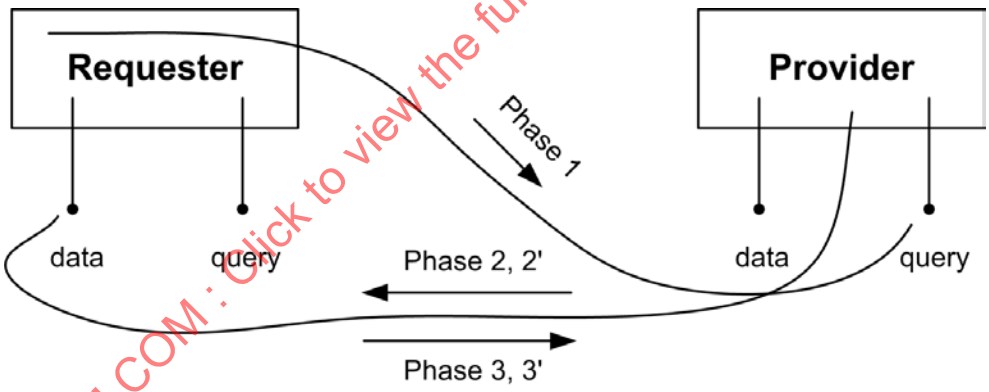

Figure 9—TRAP protocol (practical implementation)

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Phase 1: The Requester invokes the query method at the Provider. Here, the arguments transmitted at the same time with query-expression shall include query validity time [i.e., time to live (TTL)] in seconds, data callback URI, and control-signal callback URI.

Note that this action shall be carried out periodically (e.g., with an interval of one-half or one-third TTL) in order to update TTL at the Provider to continue receiving update notifications. Here, the same id shall be set in the query.

Note that provider shall return an error if it was not able to accept the query.

Phase 2: The Provider posts the update that matches the query-expression to the specified data callback URI by its data method. The Provider sends a query at the same time.

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

Phase 3: The data callback replies whether it was successful or not. If not, it also replies with the error message.

Phase 2: The Provider posts the communication errors (e.g., when it encountered an error at Phase 3) to the specified control callback URI by its data method.

Phase 3': The control callback replies whether it was successful or not. If it was failure, it also replies the error message.

Note that the TTL of the query-expression shall be decreased in the Provider by every elapsed second. If the TTL reaches "0," the Provider declines the query by removing it from the query table entry.

Note that to remove a query explicitly, the Requester shall invoke query method by specifying "0" for TTL.

5.3 Component-to-registry communication protocol

5.3.1 Type of component-to-registry communication protocol

This subclause specifies the following two types of sub-protocols for component-to-registry communication:

— REGISTRATION Protocol—for registration of the role of components and semantics of Points

— LOOKUP Protocol—for searching appropriate components and Points

Note that Registry has two methods (registration and lookup) for its access interface. See 6.4 for details. REGISTRATION protocol uses registration method and LOOKUP protocol uses lookup method.

The latter part of this subclause describes these protocols in detail.

5.3.2 REGISTRATION protocol

REGISTRATION is a protocol that enables a component to register the role of components and semantics of Points. We denote the component that submits registration request to its Registry by "Registrant." Figure 10 provides a diagram of the interaction.

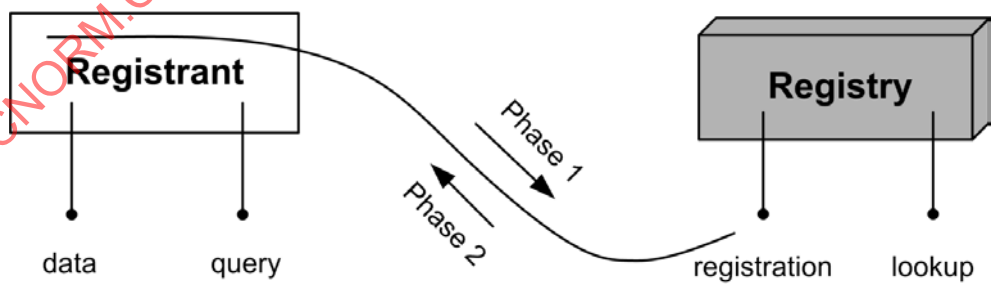

Figure 10—REGISTRATION protocol

Phase 1: The Registrant invokes the registration method of the Registry with the component information (e.g., the name, the access URI of the component, and protocol types this component supports) and its role information (e.g., what Point IDs this component manages).

Copyright © 2014 IEEE. All rights reserved.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

NOTE—Registrant implementation is not always necessary to have the data and query method. Here we specifically cite the registration of Point properties, where registration of semantic information of Points is not necessarily the role of a component. See 6.4.2 for details.

Phase 2: The Registry replies whether it was successful or not. If it was not, it shall return an error message to the Registrant.

5.3.3 LOOKUP protocol

LOOKUP is a protocol for a component to search appropriate access components (for component-to-component communication) and to search Points by semantic query. We here denote the component that searches appropriate components and Points from its Registry by "Requester." Figure 11 provides a diagram of the interaction.

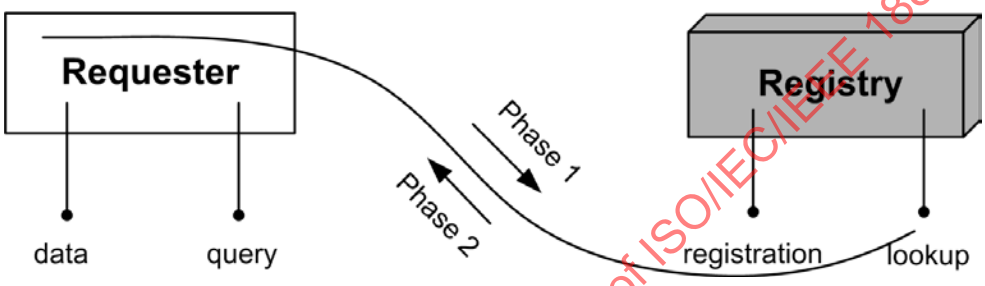

Figure 11—LOOKUP protocol

Phase 1: The Requester invokes the lookup method of the Registry. Here, the lookup expression shall indicate the "type" information of intended objects (i.e., component or Point).

Phase 2: The Registry replies the access URI of the resolved component or matched Points to the Requester.

6. Application programming interfaces

6.1 General

This subclause defines the following two types of application programming interfaces (APIs):

— Component access interface—for component-to-component communication

— Registry access interface—for component-to-registry communication

6.2 Transport data structure

Whether it is component-to-component communication or component-to-registry communication, access methods shall be implemented by RPC. In Figure 12, the Caller is invoking a method of Callee with request-data, and response-data is returning from Callee to Caller. Here, the data structure of request and response is the same; it has a header part and a body part. The header part contains control information such as query-expressions (for component-to-component communication) or lookup-expressions (for component-to-registry communication), acknowledgement, or failure information. The body part contains Point or PointSet objects with data values such as observed sensor readings and actuator commands (for

Copyright © 2014 IEEE. All rights reserved.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

component-to-component communication), or the role of component and semantics information of Point ID (for component-to-registry communication). The control information associated to each Point or specific value shall be included in the body part.

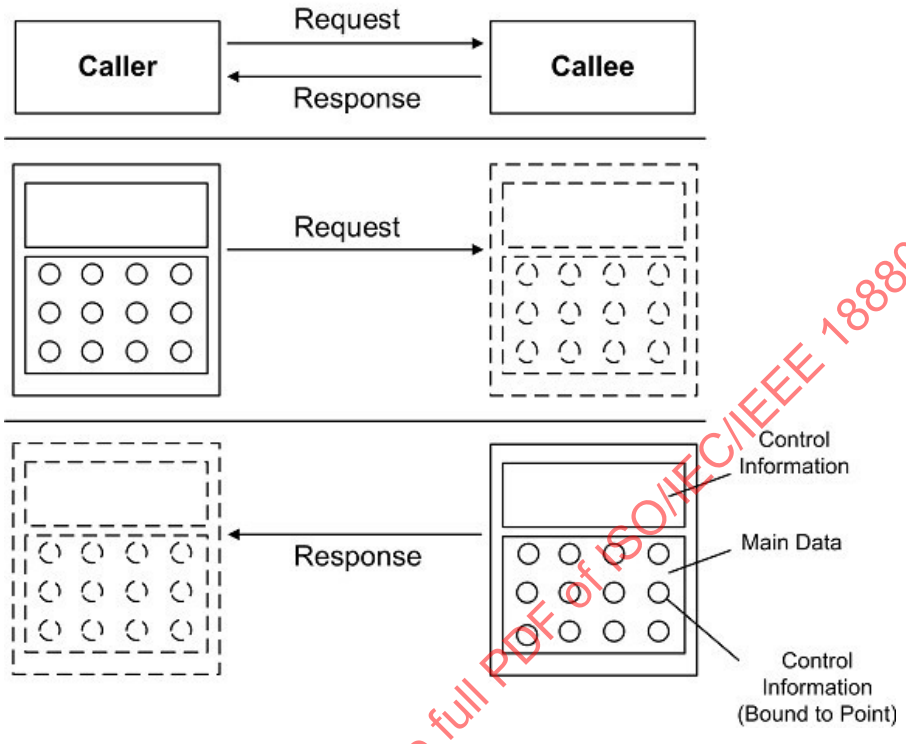

Figure 12—Transport data structure

6.3 Component access interface

6.3.1 Types of methods

A component (i.e., GW, Storage, and APP) implements this interface. This interface defines the query and data method for other components to access. The definition of component access interface in the Web Services Description Language (WSDL) form is provided in Annex D.

— Query: to retrieve or to subscribe data from the component.

— Data: to post data to the component.

The following three sub-protocols defined in 5.2 are expected to be implemented with the query and data method:

— FETCH protocol

— WRITE protocol

— TRAP protocol

For practical implementation of these protocols, see Annex C.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

NOTE—All the components should implement these methods, however, some components (especially APPs) do not always need to implement all the features of them (e.g., some components only implement the FETCH-Requester function).

6.3.2 Query method

6.3.2.1 Introduction

6.3.2.1.1 Format

Transport query (Transport *t*).

6.3.2.1.2 Outline

Query is a method for a Requester to get data from a Provider by specifying query-expression. The query-expression shall be contained in the header part of the Transport argument *t*.

This standard specifies two types of queries: type="storage" and type="stream". Depending on the type of the query the Provider shall behave in the following manner.

- type="storage"—works as FETCH protocol
- type="stream"—works as TRAP protocol

Following are the details of query method for both of the preceding cases.

6.3.2.2 Query (type="storage")

6.3.2.2.1 Request

The query-expression shall be put in the header part of *t*. The body part of *t* shall be ignored.

6.3.2.2.1.1 Attributes of query-expression

- id: For identifying the query (UUID).
- type: “Type” shall always be set as “storage.”
- acceptableSize: Requester’s acceptable number of value elements at an RPC response.
- cursor: For retrieval of the succeeding dataset. Cursor shall not be set at the first query. This cursor shall be given by the provider for the next dataset retrieval.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

6.3.2.2.2 Response

6.3.2.2.2.1 Header

In the header part of the response, the query (and at least the same id attribute) and an OK or Error object shall be contained. If there is a cursor attribute in the query-expression, it means that there are succeeding datasets at the Provider side. If not, it means that all the data has been retrieved from the Provider.

The OK object indicates that the procedure at the Provider has been successfully completed. The Error object indicates that an error has occurred while carrying out the procedure. The error information shall be included in the Error object.

6.3.2.2.2.2 Body

If it was successful (that is, if an OK object was contained in the header), objects of PointSets or Points shall be contained in the body part. If not, the Body object shall be ignored.

6.3.2.3 Query (type="stream")

6.3.2.3.1 Request

The query-expression shall be put in the header part of *t*. The body part of *t* shall be ignored.

6.3.2.3.1.1 Attributes of query-expression

- id: For identifying the query (UUID).
- type: “Type” shall always be set as “stream.”
- ttl: Valid time of the query at the Provider (i.e., time to live) in seconds.
- callbackData: The callback URI of the query-match data.
- callbackControl: The callback URI of the control information.
- acceptableSize: callbackData’s acceptable number of value elements in one transfer (optional).

6.3.2.3.2 Response

6.3.2.3.2.1 Header

In the header part of the response, the query (and at least the same id attribute) and an OK or Error object shall be contained.

The OK object indicates that the procedure at the provider has been successfully completed. The Error object indicates that an error has occurred while carrying out the procedure. The error information shall be included in the Error object.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

6.3.2.3.2.2 Body

The Body object shall be ignored.

6.3.2.3.3 Other behavior

The Provider invokes the data method of the callbackData to transfer the query-match data. Here, the provider adds the query-expression in the header part of the argument *t* so that the callbackData can identify the context of the data transfer.

Note that Requester shall use the same query id when it attempts to update the query TTL at the Provider. (It is TRAP protocol specification.)

6.3.3 Data method

6.3.3.1 Format

Transport data (Transport *t*).

6.3.3.2 Outline

Data is a method for a Requester to transfer PointSet or Point objects to a Target. The data (whether they are PointSet or Point objects) shall be contained in the body part of the Transport argument *t*.

When this method is used by the TRAP protocol for transferring data from a Provider to a callback, the header part of *t* shall contain the consistent query-expression matching that of query method.

6.3.3.3 Request

Data objects (i.e., PointSet or Point) shall be put in the body part of the argument *t*. PointSet and Point objects should not have more than two depth descendants. In the header part, a query-expression and control-signal shall be contained in the context of TRAP protocol.

6.3.3.4 Response

6.3.3.4.1 Header

In the header part of the response, an OK or Error object shall be contained.

The OK object indicates that the procedure at the provider has been successfully completed. The Error object indicates that an error has occurred while carrying out the procedure. The error information shall be included in the Error object.

6.3.3.4.2 Body

The body part shall be ignored.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

6.4 Registry access interface

6.4.1 Types of methods

The Registry shall implement this interface. This interface defines the registration and lookup method for components (i.e., GW, Storage, APPs) to access. Implementation of this interface at components is optional. The definition of registry access interface in WSDL form is provided in Annex E.

— Registration: to register the role of component and semantics of Points.

— Lookup: to find an appropriate component to access, and to find Points by semantic-query.

The following two sub-protocols defined in 5.3 are expected to be implemented with the registration and lookup method.

— REGISTRATION protocol

— LOOKUP protocol

6.4.1.1 Data structure of the registry access interface

```
Transport
    +--Header
    |    +--Lookup
    |         +--Key
    |         +--Point
    +--Body
         +--Component
         |    +--Key
         +--Point
```

Here, the data structure for REGISTRATION and LOOKUP protocol is the same; both have a header part and a body part. The header part contains lookup-expressions, acknowledgement, or failure information. The body part contains information that describes the role of component(s) and the semantics of Point(s). See 8.4 for details.

6.4.2 Registration method

6.4.2.1 Introduction

6.4.2.1.1 Format

Transport registration (Transport *t*).

6.4.2.1.2 Outline

Registration is a method for a Registrant to register the role of itself and semantics of Points to Registry. The REGISTRATION protocol (defined in 5.3.2) uses this method. In this specification, there are two kinds of registration information that Registry should manage, as follows:

— Information that describes the role of component(s)

Copyright © 2014 IEEE. All rights reserved.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

— Information that describes Point(s)

6.4.2.2 Registration of components

6.4.2.2.1 Request

Registration information of components shall be put in the body part of the Transport argument *t*.

6.4.2.2.2 Response

6.4.2.2.2.1 Header

In the header part of the response, an OK or Error object shall be contained.

The OK object indicates that the procedure at the Registry has been successfully completed. The Error object indicates that an error has occurred while carrying out the procedure. The error information shall be included in the Error object.

6.4.2.2.2.2 Body

The body part should be ignored.

6.4.2.3 Registration of Points

Semantics information of Points shall be put in the body part of the argument *t*.

In this standard, we do not specify any property for the Point except Point ID as the identifier. The system operator may design appropriate Point profiles according to the context of different applications and submit their profiles to the Registry by the registry access interface. Thus we keep semantics information of Points open to make system deployment and operation to be flexible.

6.4.3 Lookup method

6.4.3.1 Introduction

6.4.3.1.1 Format

Transport lookup (Transport *t*).

6.4.3.1.2 Outline

Lookup is a method for a component to find appropriate component(s) for a specified Point, and to find matched Point(s) by semantic-query. LOOKUP protocol (defined in 5.3.3) uses this method. This document specifies two types of lookup: type="component" and type="point". Depending on the type of lookup, the Registry shall behave in the following manner:

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

— type="component" finds appropriate component(s) to access. The Registry replies the access URI(s) for the resolved component(s).

— type="point" finds Point(s) by semantic-query. The Registry replies the matched Point(s) profile.

The following subclauses specify the details of two fields.

6.4.3.2 Lookup (type="component")

6.4.3.2.1 Request

The lookup-expression shall be put in the header part of the Transport argument *t*. The body part of *t* shall be ignored.

6.4.3.2.1.1 Attributes of lookup-expression

— id: For identifying the lookup(UUID).

— type: “Type” shall always be set as “component.”

6.4.3.2.2 Response

6.4.3.2.2.1 Header

In the header part of the response, a lookup-expression and an OK or Error object shall be contained.

The OK object indicates that the procedure at the Registry has been successfully completed. The Error object indicates that an error has occurred while carrying out the procedure. The error information shall be included in the Error object.

6.4.3.2.2.2 Body

If it was successful (if an OK object was contained in the header), the access URI(s) for the resolved component(s) shall be contained in the body part. If not, the Body object shall be ignored.

6.4.3.3 Lookup (type="point")

6.4.3.3.1 Request

The lookup-expression shall be put in the header part of *t*. Here, the header contains specified Point properties for semantic-query. The body part of *t* shall be ignored.

6.4.3.3.1.1 Attributes of lookup-expression

— id: For identifying the lookup(UUID).

— type: “Type” shall always be set as “point.”

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

6.4.3.3.2 Response

6.4.3.3.2.1 Header

In the header part of the response, a lookup-expression and an OK or Error object shall be contained.

The OK object indicates that the procedure at the Registry has been successfully completed. The Error object indicates that an error has occurred while carrying out the procedure. The error information shall be included in the Error object.

6.4.3.3.2.2 Body

If it was successful (if an OK object was contained in the header), the lookup-match Point(s) profile shall be listed in the body part. If not, the Body object shall be ignored.

7. Data and query model

7.1 General

This clause defines the data and query model for component-to-component communication. This data model basically takes the conventional tree data structure.

7.2 Point management with PointSet tree

Figure 13 is a typical structure of PointSet data tree. A PointSet aggregates Points and PointSets. This structure allows hierarchical management of Points by PointSet. A Point has a sequence of Value elements. A sensor reading or an actuator command is contained inside a Value element. Every PointSet and Point has a globally unique identifier by id attribute. A Value element itself usually is not bound to any globally unique identifiers, but has a time attribute to identify when the value was taken from a sensor or when the value shall be set to an actuator.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

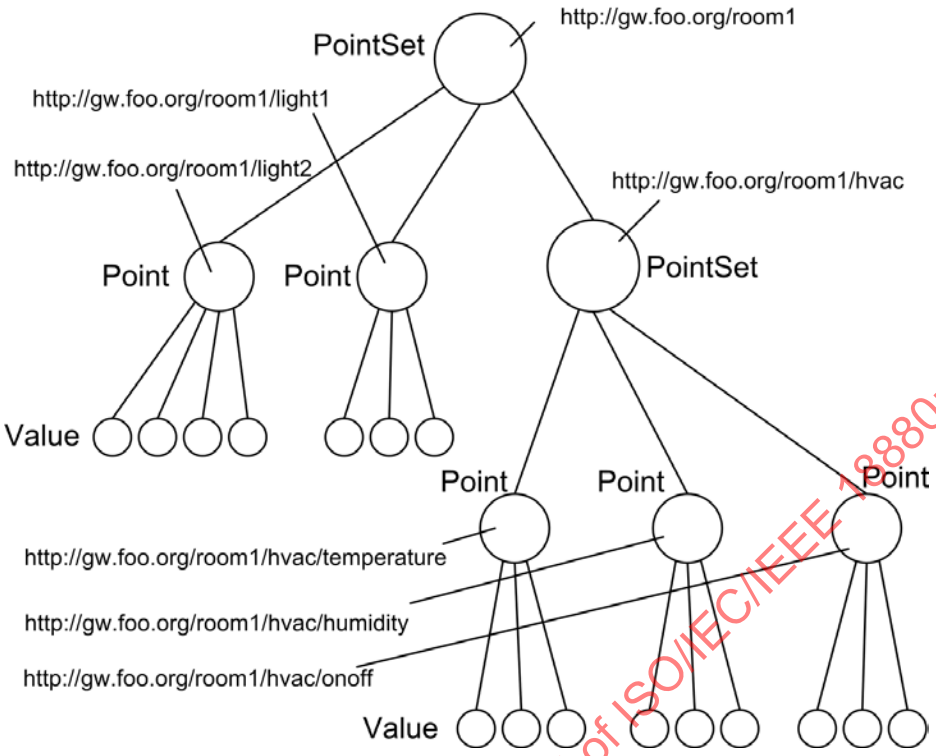

Figure 13—PointSet tree data model

Though the specification defines the concept of PointSet, implementation of the PointSet feature is optional. All the components should allow operation without PointSet.

In this document, we do not specify any attributes for PointSet, Point, and Value except for the identifier and time attribute. The attribute shall be defined in the context of applications of facility networking. Thus, we leave other attributes open in order to make this protocol adaptive to many application scenarios.

7.3 Query model for PointSet tree

A requester of FETCH and TRAP specifies the range of data set by query and key elements. Then, the provider shall provide corresponding PointSet, Point, and Value objects. It provides the specified PointSet and Point objects as the children of the body element in response, and the PointSet and Point objects should not have more than two depth descendants on the PointSet tree.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

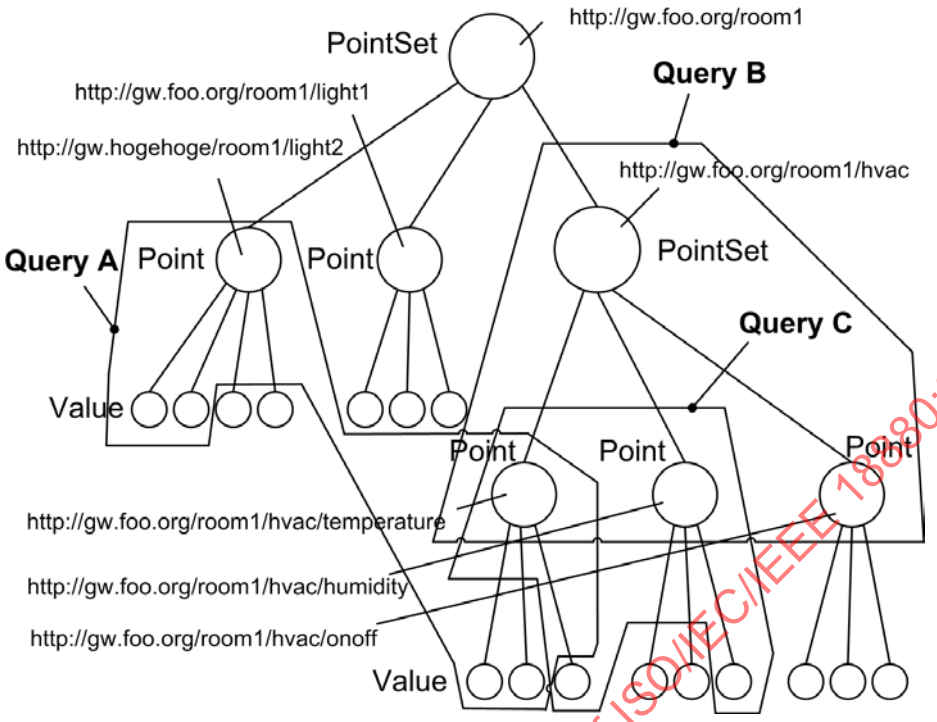

Figure 14—Query for specifying the range of data set

7.3.1 Query A

Figure 14 provides a simple query model for PointSet tree. The following is a description of query for specifying the range of dataset.

Description: get values of "http://gw.foo.org/room1/hvac/light2" or "http://gw.foo.org/room1/hvac/temperature"; these times are between 2009-10-01T00:00:00+09:00 and 2009-10-02T00:00:00+09:00.

```
<query>
 <key id="http://gw.foo.org/room1/light2" attrName="time"
gteq="2009-10-01T00:00:00+09:00"
lteq="2009-10-02T00:00:00+09:00" />
 <key id="http://gw.foo.org/room1/hvac/temperature" attrName="time"
gteq="2009-10-01T00:00:00+09:00"
lteq="2009-10-02T00:00:00+09:00" />
</query>
```

7.3.2 Query B

Description: get Point information at "http://gw.foo.org/room1/hvac"

```
<query>
 <key id="http://gw.foo.org/room1/hvac" />
</query>
```

7.3.3 Query C

Description: get the latest values of "http://gw.foo.org/room1/hvac/temperature" and "http://gw.foo.org/room1/hvac/humidity"

Copyright © 2014 IEEE. All rights reserved.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

```
<query>
 <key id="http://gw.foo.org/room1/hvac/temperature" attrName="time"
select="maximum" />
 <key id="http://gw.foo.org/room1/hvac/humidity" attrName="time"
select="maximum" />
</query>
```

8. Data structure

8.1 General

This clause defines the data structure for component-to-component and component-to-registry communication protocol. All the messages shall be described in XML-format [W3C, Extensible Markup Language (XML) 1.0]. A class in this document corresponds to an XML element. The following set of class definitions provides the scheme of XML (W3C, XML Schema Part 1: Structures, XML Schema Part 2: Datatypes). The XML namespace (W3C, Namespaces in XML 1.0) for this data format is xmlns="http://gutp.jp/fiap/ 2009/11/".

8.2 Naming rules between object-class names and XML-element names

Names are case-sensitive. However, the class name shall start with an upper case, while the XML-element name shall start with a lower case. From the second letter on, the class name and XML-name shall be identical. However, if the first two letters of the class name are upper cases, the first letter of the XML shall be an upper case too.

8.3 Data structure for component-to-component communication protocol

8.3.1 Introduction

This subclause defines the data structure for component-to-component communication protocol. This data structure is expected to be delivered with request and response interaction in the following protocols (defined in 5.2):

— FETCH protocol
— WRITE protocol
— TRAP protocol

8.3.2 List of classes

8.3.2.1 General

This version of the UGCCNet specification defines the following object classes for component-to-component communication protocol:

— Transport
— Header
— Body

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

— PointSet
— Point
— Value
— Query
— Key
— OK
— Error

8.3.2.2 Transport class

8.3.2.2.1 Outline

Transport class enables delivery of both data and control-plane information in one message.

8.3.2.2.2 Member

Transport has a Header object and a Body object. They may be omitted when they do not contain any data.

8.3.2.2.3 Attribute

None.

8.3.2.2.4 Example

```
<transport>
 <header> ... </header>
 <body> ... </body>
</transport>
```

8.3.2.3 Header class

8.3.2.3.1 Outline

Header class provides a container for control-plane information. Control-plane information includes query-expression, acknowledgement, or failure of the access and redirection.

8.3.2.3.2 Member

Query, OK, and Error object.

8.3.2.3.3 Attribute

None.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

8.3.2.3.4 Example 1

```
<header>
 <query ... > ... </query>
</header>
```

8.3.2.3.5 Example 2

```
<header>
 <OK/>
</header>
```

8.3.2.4 Body class

8.3.2.4.1 Outline

The Body object provides a container for the data of Points or PointSets.

Note that the control signals that are tightly attached to Points or Values shall be also dealt with data and expressed with Points or Values.

8.3.2.4.2 Member

Body has PointSet objects and Point objects.

8.3.2.4.3 Attribute

None.

8.3.2.4.4 Example 1

```
<body>
 <pointSet id="http://gw.foo.org/tv1">...</pointSet>
 <pointSet id="http://gw.foo.org/refrigerator1">...</pointSet>
 <pointSet id="http://gw.foo.org/pot1">...</pointSet>
</body>
```

8.3.2.4.5 Example 2

```
<body>
 <point id="http://gw.foo.org/tv1/power">...</point>
 <point id="http://gw.foo.org/tv1/switch">...</point>
 <point id="http://gw.foo.org/pot1/power">...</point>
</body>
```

8.3.2.4.6 Example 3

```
<body>
 <pointSet id="http://gw.foo.org/room1/light">...</pointSet>
 <pointSet id="http://gw.foo.org/room1/hvac">...</pointSet>
 <point id="http://gw.foo.org/tv1/power">...</point>
```


Copyright © 2014 IEEE. All rights reserved.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

```
 <point id="http://gw.foo.org/pot1/power">...</point>
</body>
```

8.3.2.5 PointSet class

8.3.2.5.1 Outline

PointSet aggregates related Point objects and PointSet objects. In the conventional operation, we develop hierarchical data structure to manage related Points by one identifier. PointSet class enables this type of aggregation.

8.3.2.5.2 Member

PointSet has PointSet objects and Point objects.

8.3.2.5.3 Attribute

— id: URI-based identifier for this PointSet.

8.3.2.5.4 Example 1

```
<pointSet id="http://gw.foo.org/room1">
 <pointSet id="http://gw.foo.org/room1/light/">...</pointSet>
 <pointSet id="http://gw.foo.org/room1/hvac">...</pointSet>
</pointSet>
```

8.3.2.5.5 Example 2

```
<pointSet id="http://gw.foo.org/tv1/">
 <point id="http://gw.foo.org/tv1/power">...</point>
 <point id="http://gw.foo.org/tv1/switch">...</point>
 <point id="http://gw.foo.org/tv1/channel">...</point>
</pointSet>
```

8.3.2.5.6 Example 3

```
<pointSet id="http://gw.foo.org/room1">
 <pointSet id="http://gw.foo.org/room1/light">...</pointSet>
 <pointSet id="http://gw.foo.org/room1/hvac">...</pointSet>
 <point id="http://gw.foo.org/room1/temperature">...</point>
 <point id="http://gw.foo.org/room1/humidity">...</point>
</pointSet>
```

8.3.2.6 Point class

8.3.2.6.1 Outline

The Point object is a representation of a Point.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

8.3.2.6.2 Member

A Point object has Value objects.

8.3.2.6.3 Attribute

— id: identifier of the Point (i.e., Point ID)

8.3.2.6.4 Example

```
<point id="http://gw.foo.org/room1/temperature">
 <value time="2009-09-01T00:00:00+09:00">25.5</value>
 <value time="2009-09-01T00:01:00+09:00">25.6</value>
 <value time="2009-09-01T00:02:00+09:00">25.6</value>
 <value time="2009-09-01T00:03:00+09:00">25.7</value>
 ...
</point>
```

8.3.2.7 Value class

8.3.2.7.1 Outline

A Value object contains one specific data value. An input data from a sensor and an outgoing data to an actuator shall be encapsulated by the Value object.

8.3.2.7.2 Member

None.

8.3.2.7.3 Attribute

— time: generated time (INPUT case) or scheduled time (OUTPUT case) in ISO 8601 format with specifying at least second and timezone.

8.3.2.7.4 Example

```
<value time="2009-10-19T00:00:00+09:00">true</value>
<value time="2009-10-19T00:00:00+09:00">false</value>
<value time="2009-10-19T00:00:00+09:00">10</value>
<value time="2009-10-19T00:00:00+09:00">0</value>
<value time="2009-10-19T00:00:00+09:00">3.4</value>
<value time="2009-10-19T00:00:00+09:00">0.5323</value>
<value time="2009-10-19T00:00:00+09:00">HIGH</value>
<value time="2009-10-19T00:00:00+09:00">MID</value>
<value time="2009-10-19T00:00:00+09:00">LOW</value>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

8.3.2.8 Query class

8.3.2.8.1 Outline

The Query object manages query-expressions for storage-based and stream-based queries.

8.3.2.8.2 Member

Query class has Key objects.

8.3.2.8.3 Attribute

- id: the identifier of this query (UUID).
- type: the type of this query := {storage, stream}.
- cursor: the cursor for sequential dataset retrieval (valid when type="storage") acceptableSize: Receiver's maximum acceptable size of value objects at one RPC.
- ttl: validity time of query at the Provider (valid when type="stream").
- callbackData: The URI of data callback in TRAP protocol (valid when type="stream").
- callbackControl: The URI of control-signal callback in TRAP protocol (valid when type="stream").

8.3.2.8.4 Example

```
<query id="6229c37f-970d-9292-83e4-7c0e54733f8a"
       type="storage"
       acceptableSize="20"
       cursor="dab751ed-0133-4ce4-8b7d-ba5c54ce4fb5">
 <key> ... </key>
 <key> ... </key>
</query>

<query id="9eed9de4-1c48-4b08-a41d-dac067fc1c0d"
      type="stream"
      ttl="60"
      callbackData="http://foo.org/axis/services/GUTAPI"
      callbackControl="http://foo.org/axis/services/GUTAPI" >
 <key> ... </key>
 <key> ... </key>
</query>
```

8.3.2.9 Key class

8.3.2.9.1 Outline

Key class allows description of query-expression. A Key object corresponds to a key predicate in 7.3.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

8.3.2.9.2 Member

Key class has Key objects.

8.3.2.9.3 Attribute

- id: the target id of Point or PointSet.
- attrName: the attribute name for the following.
  - eq: the predicate becomes true if the key attribute value is equal to the specified value, otherwise becomes false.
  - neq: the predicate becomes true if the key attribute value is not equal to the specified value, otherwise becomes false.
  - lt: the predicate becomes true if the key attribute value is less than the specified value.
  - gt: the predicate becomes true if the key attribute value is greater than the specified value.
  - lteq: the predicate becomes true if the key attribute value is less than or equal to the specified value.
  - gteq: the predicate becomes true if the key attribute value is greater than or equal to the specified value.
- select: for selection of {maximum, minimum} attribute value.
- trap: for event detection := {changed} (valid if the query type="stream").

Note that the time format of eq, neq, lt, gt, lteq, and gteq shall follow ISO 8601 format when specifying at least second and timezone.

8.3.2.9.4 Example

```
<key id="http://gw.foo.org/room1/temperature"
 attrName="time" select="maximum />

<key id="http://gw.foo.org/room1/temperature"
attrName="time"
lteq="2009-10-01T00:00:00+08:00"
gteq="2009-09-01T00:00:00+08:00" />

<key id="http://gw.foo.org/room1/temperature"
attrName="time" trap="changed" />

<key id="http://gw.foo.org/room1/temperature"
attrName="value" trap="changed" />
```

8.3.2.10 OK class

8.3.2.10.1 Outline

The OK object indicates that a request has been successfully accepted.

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

8.3.2.10.2 Member

None.

8.3.2.10.3 Attribute

None.

8.3.2.10.4 Example

```
<OK/>
```

8.3.2.11 Error class

8.3.2.11.1 Outline

Error class contains error messages.

8.3.2.11.2 Member

None.

8.3.2.11.3 Attribute

— type: category of error

The value of the type attribute for component-to-component communication is shown in Annex F.

8.3.2.11.4 Example

```
<error type="INVALID_REQUEST">Malformed IEEE1888/XML Error - query
element should be specified in FETCH request</error>
<error type="POINT_NOT_FOUND">Point id=”...” is not managed in this
storage.</error>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

8.4 Data structure for component-to-registry communication protocol

8.4.1 Introduction

This subclause defines data structure for component-to-registry communication protocol. This data structure is expected to be delivered with request and response interaction in the following protocols (defined in 5.3):

— REGISTRATION Protocol

— LOOKUP Protocol

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

8.4.2 List of classes

8.4.2.1 General

This version of the UGCCNet specification defines the following object classes for component-to-registry communication protocol:

- Transport
- Header
- Body
- Component
- Point
- Lookup
- Key
- OK
- Error

8.4.2.2 Transport class

8.4.2.2.1 Outline

Transport class enables delivery of both data and control-plane information in one message.

8.4.2.2.2 Member

Transport has a Header object and a Body object. They may be omitted when they do not contain any data.

8.4.2.2.3 Attribute

None.

8.4.2.2.4 Example 1

```
<transport>
 <header> ... </header>
 <body> ... </body>
</transport>
```

8.4.2.2.5 Example 2

About the registration of Point properties, the system operators design Point profiles and submit it to Registry with namespace awareness, e.g.,

```
<transport xmlns:s="...">
 <body> ... </body>
</transport>
```


Copyright © 2014 IEEE. All rights reserved.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

8.4.2.3 Header class

8.4.2.3.1 Outline

Header class provides a container for control-plane information. Control-plane information includes lookup-expression, acknowledgement, or failure of the registration and lookup.

8.4.2.3.2 Member

Lookup, OK, and Error object

8.4.2.3.3 Attribute

None.

8.4.2.3.4 Example 1

```
<header>
 <lookup ... > ... </lookup>
</header>
```

8.4.2.3.5 Example 2

```
<header>
 <OK/>
</header>
```

8.4.2.4 Body class

8.4.2.4.1 Outline

The Body object provides a container for the resolved component (i.e., GW, Storage, APPs) and Point properties.

8.4.2.4.2 Member

Body has Component objects and Point objects (e.g., Point objects associated with GW, Point objects whose Point IDs are managed by Storage, the APP-available, or APP-providing Point IDs).

8.4.2.4.3 Attribute

None.

8.4.2.4.4 Example 1

```
<body>
 <component ...> ... </component>
</body>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

8.4.2.4.5 Example 2

```
<body>
 <point id="..." .../>
 <point id="..." .../>
 <point id="..." .../>
</body>
```

8.4.2.5 Component class

8.4.2.5.1 Outline

Component is the generalization object for GW, Storage, and APPs.

8.4.2.5.2 Member

Component has Key objects.

8.4.2.5.3 Attribute

- name: the name of the specified component (i.e., GW, Storage, or APP).
- uri: the access URI of the specified component.
- priority: priority of access for the redundant dataset (optional).
- support: which protocol type(s) the specified component supports (i.e., FETCH, WRITE, TRAP).
- expires: the effective expiration time of the registration in seconds (optional).

Note that the “priority” attribute should be provided if there exist more than two candidate components to access (e.g., main-backup operation to increase availability). If the priority value gets large, the component gets more priority to be selected.

Note that the “expires” attribute should be supplied to the Registry to indicate the lifetime of the registration, and it shall be decreased in Registry by every elapsed second. If the “expires” reaches “0,” Registry declines the registration by removing it from the Registry.

Note that to remove a component registration explicitly, Registrant shall invoke the registration method of Registry by specifying “0” for “expires.”

Note that if this attribute is not provided by Registrant, Registry determines how long the registration is valid.

8.4.2.5.4 Example 1

A typical GW would be described as follows:

```
<component name="myGW" uri="http://fiap-gw.gutp.ic.i.u-
tokyo.ac.jp/axis/services/FIAPBACnetWSGW" support="FETCH|TRAP">
<key id="http://fiap-gw.gutp.ic.i.u-
tokyo.ac.jp/EngBldg2/10F/102A2/DB1"stream="in" limit="1"/>
  <key id="http://fiap-gw.gutp.ic.i.u-okyo.ac.jp/EngBldg2/10F/102A2/DB2"
```


Copyright © 2014 IEEE. All rights reserved.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

```
  stream="in" limit="1" />
  <key id="http://fiap-gw.gutp.ic.i.u-okyo.ac.jp/EngBldg2/10F/102A2/RH1"
  stream="in" limit="1" />
  <key id="http://fiap-gw.gutp.ic.i.u-okyo.ac.jp/EngBldg2/10F/102A2/RH2"
  stream="in" limit="1" />
  ...
</component>
```

Note that limit="1" means that myGW only has 1 data cache.

8.4.2.5.5 Example 2

A typical Storage would be described as follows:

```
<component name="myStorage" uri="http://fiap-storage.gutp.ic.i.u-
tokyo.ac.jp/axis/services/FIAPStorage" support="FETCH|WRITE">
<key id="http://fiap-gw.gutp.ic.i.u-tokyo.ac.jp/EngBldg2/10F/102A2/DB1"
/>
<key id="http://fiap-gw.gutp.ic.i.u-tokyo.ac.jp/EngBldg2/10F/102A2/DB2"
/>
<key id="http://fiap-gw.gutp.ic.i.u-tokyo.ac.jp/EngBldg2/10F/102A2/RH1"
/>
<key id="http://fiap-gw.gutp.ic.i.u-tokyo.ac.jp/EngBldg2/10F/102A2/RH2"
/>
 ...
</component>
```

8.4.2.6 Point class

8.4.2.6.1 Outline

The Point object is a representation of a “Point” but without a value.

8.4.2.6.2 Member

None.

Note that different from the data structure of component-to-component communication protocol, Point object here does not have any object member for itself.

8.4.2.6.3 Attribute

— id: identifier of the Point (i.e., Point ID)

Note that other attributes (e.g., Type, Writable, Location…) remain open for flexible design of Point properties.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

8.4.2.6.4 Example

```
<point id="X" s:type="BINARY_INPUT" s:writable="false"
s:location="Building2F221MeetingRoom1"/>
<point id="Y" s:type="BINARY_INPUT" s:writable="false"
s:location="Building2F221MeetingRoom1"/>
<point id="Z" s:type="ANALOG_INPUT" s:writable="false"
s:location="Building2F221MeetingRoom1"/>
<point id="W" s:type="MULTI_STATE_INPUT" s:writable="false"
s:location="Building2F221MeetingRoom1"/>
```

8.4.2.7 Lookup class

8.4.2.7.1 Outline

The Lookup object manages lookup-expressions.

8.4.2.7.2 Member

Lookup class has Key and Point objects.

8.4.2.7.3 Attribute

- id: the identifier of this lookup (UUID).
- type: the type of this lookup := {component, point}.

8.4.2.7.4 Example 1: Lookup of components

```
<lookup id="6e5a0e85-b4a0-485f-be54-a758115317e1" type="component">
 <key id="http://fiap-gw.gutp.ic.i.u-tokyo.ac.jp/EngBldg2/10F/102A2/DB1"
.../>
</lookup>
```

8.4.2.7.5 Example 2: Lookup of Points

```
<lookup id="3f2504e0-4f89-11d3-9a0c-0305e82c3301" type="point">
 <point s:type="BINARY_INPUT" s:location="Building2F221MeetingRoom1"
.../>
</lookup>
```

Note that s is the namespace prefix for the target namespace of the Point profile, e.g., xmlns:s="http://application-dependent.org/AttributeSet/".

8.4.2.8 Key class

8.4.2.8.1 Outline

Key class allows description of lookup-expression and dataset profile for Points (multiple Keys in parallel).

Copyright © 2014 IEEE. All rights reserved.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

8.4.2.8.2 Member

None.

8.4.2.8.3 Attribute

- id: identifier of the Point (i.e., Point ID)
- attrName: the searching attributes of objects
- stream: data flow comes into and out {in, out}of the component
- limit: data caching maximum size
- eq: this predicate becomes true if the key attribute value is equal to the specified value, otherwise it becomes false
- neq: this predicate becomes true if the key attribute value is not equal to the specified value, otherwise it becomes false
- lt: this predicate becomes true if the key attribute value is less than the specified value
- gt: this predicate becomes true if the key attribute value is greater than the specified value
- lteq: this predicate becomes true if the key attribute value is less than or equal to the specified value
- gteq: this predicate becomes true if the key attribute value is greater than or equal to the specified value

Note that the time format of eq, neq, lt, gt, lteq, and gteq shall follow ISO 8601 format when specifying at least second and timezone.

8.4.2.8.4 Example

If a component has data of Point id="X" from time="2010-01-01T00:00:00+09:00," this part would be:

```
<key id="X" attrName="time" gteq="2010-01-01T00:00:00+09:00"/>
```

If a component can provide input data stream (i.e., the component has sensor X), this part would be:

```
<key id="X" attrName="time" stream="in"/>
```

If a component can provide output data stream (i.e., the component has actuator X), this part would be:

```
<key id="X" attrName="time" stream="out"/>
```

If a component can provide input and output data stream, this part would be:

```
<key id="X" attrName="time" stream="in|out"/>
```

8.4.2.9 OK class

8.4.2.9.1 Outline

The OK object indicates that a request has been successfully accepted.

Copyright © 2014 IEEE. All rights reserved.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

8.4.2.9.2 Member

None.

8.4.2.9.3 Attribute

None.

8.4.2.9.4 Example

```
<OK/>
```

8.4.2.10 Error class

8.4.2.10.1 Outline

Error class contains error messages.

8.4.2.10.2 Member

None.

8.4.2.10.3 Attribute

— type: category of error.

The value of the type attribute for component-to-registry communication is shown in Annex F.

8.4.2.10.4 Example

```
<error type="INVALID_REQUEST">Malformed IEEE1888/XML Error – lookup
element should be specified.</error>
<error type="SERVER_ERROR">The server encountered an out of memory
error.</error>
```

9. Protocol binding

Protocol binding may be achieved using almost all of the existing communication protocols for data transmission, such as SMTP, SIP, FTP, and HTTP. However, taking into account the universality of application and implementation, this document introduces SOAP, HTTP, and SIP as underlying binding protocols for the implementation of the UGCCNet communication protocol, especially SIP for operational and manageability aspects of networking facilities.[8]

[8] Specification for UGCCNet over SIP is the future work item of IEEE P1888 Working Group.

Copyright © 2014 IEEE. All rights reserved.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

10. Security considerations

UGCCNet protocol is basically open. It assumes multi-domain operation and public access from other domain's system components. In this context, security requirements to the system would be listed as follows:

- To avoid unintended data disclosure to the public
- To avoid unauthorized access to writable resources
- Availability and confidentiality of remote communication host
- Integrity and confidentiality of data
- To avoid unintended access or operational conflicts

To get confidentiality of remote communication host, we would be able to use VPN, SSL, SSH, and other related technologies. HTTPS, or SIP and its security extension, would help in getting integrity and confidentiality of data.

Access control and access confliction management shall be the other important but different types of security issues that should be discussed independently. Generally, access control is used to allow only specific users to access both readable and writable resources, which would certainly help to avoid unauthorized access from or unintended data disclosure to the public (sometimes anonymous) users. In order to manage this, the system would need to introduce the concept of users to identify who is accessing the resources. We assume URI-based identification for user authentication just as Point ID takes URI for its identifier. Authentication of these users and components (probably by taking advantage of the existing authentication platforms) would certainly need to be considered.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

Annex A

(informative)

Typical sequence of UGCCNet communication

Figure A.1 shows a typical sequence of UGCCNet communication.

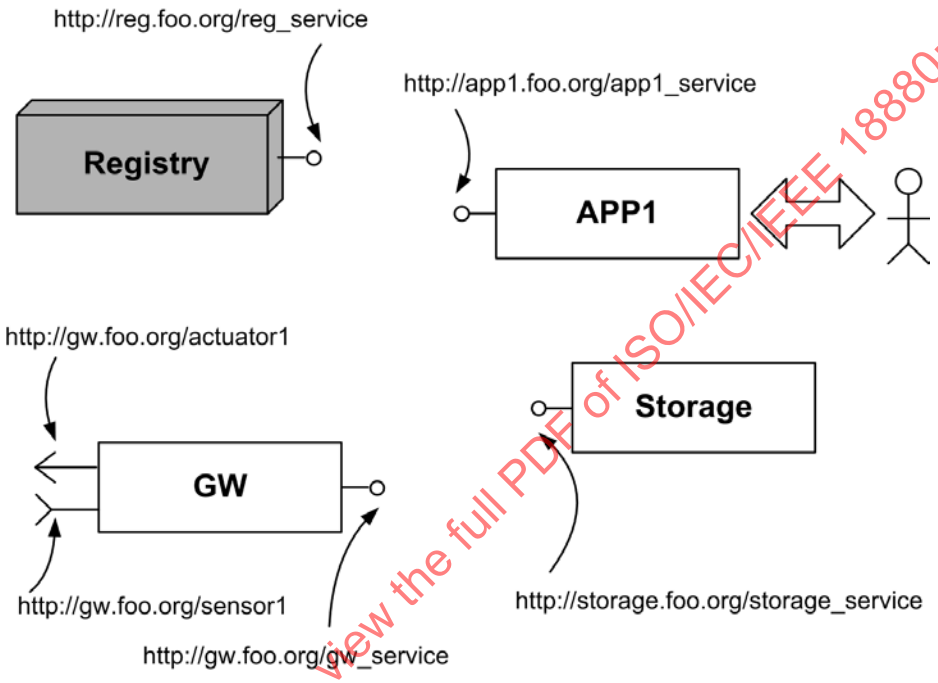

Figure A.1—Configuration detail of Figure 2

Case A: Registration of components:

- http://reg.foo.org/reg_service <--(request) -- registration("Description of GW")
- http://reg.foo.org/reg_service --(response)--> "<OK />"
- http://reg.foo.org/reg_service <--(request) -- registration("Description of Storage")
- http://reg.foo.org/reg_service --(response)--> "<OK />"
- http://reg.foo.org/reg_service <--(request) -- registration("Description of APP")
- http://reg.foo.org/reg_service --(response)--> "<OK />"

Case B: Searching the Points by semantic-query:

- http://reg.foo.org/reg_service <--(request) -- lookup("Search query for Points through specified Point properties")
- http://reg.foo.org/reg_service --(response)--> lookup<the lookup-match Point(s) profile>

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

Case C: Searching a storage component for point ID="http://gw.foo.org/sensor1":

— http://reg.foo.org/reg_service <--(request) -- lookup("Search query for Storage that manages point ID=\"http://gw.foo.org/sensor1\.")

— http://reg.foo.org/reg_service --(response)--> It is http://storage.foo.org/storage_service

Case D: Transmission of data from GW to Storage:

— http://storage.foo.org/storage_service <--(request)-- data("<point id=\"http://gw.foo.org/sensor1\" > <value time=\"2009-09-01T00:00:00+09:00\">25.5</value></point>")

— http://storage.foo.org/storage_service --(response)--> "<OK />"

Case E: Sequential retrieval of data from Storage by APP:

— http://storage.foo.org/storage_service <--(request) -query("<query>Values from time=\"2009-09-01T00:00:00+09:00\" until time=\"2009-10-01T00:00:00+09:00\" for point ID=\"http://gw.foo.org/sensor1\"</query>")

— http://storage.foo.org/storage_service --(response)--> "<query cursor=\"001\">"++"<point id=\"http://gw.foo.org/ sensor1\"><value>...</value>...</point>"

— http://storage.foo.org/storage_service <--(request) -- query("<query cursor=\"001\">Values from time=\"2009-09-01T00:00:00+09:00\" until time=\"2009-10-01T00:00:00+09:00\" for point ID=\"http://gw.foo.org/sensor1\"</query>")

— http://storage.foo.org/storage_service --(response)--> "<query cursor=\"001\">"++"<point id=\"http://gw.foo.org/ sensor1\"><value>...</value>...</point>"

— http://storage.foo.org/storage_service <--(request) -- query("<query cursor=\"001\">Values from time=\"2009-09-01T00:00:00+09:00\" until time=\"2009-10-01T00:00:00+09:00\" for point ID=\"http://gw.foo.org/sensor1\"</query>")

— http://storage.foo.org/storage_service --(response)--> "<query />"++"<point id=\"http://gw.foo.org/ sensor1\"><value>...</value>...</point>"

Case F: Searching a GW component for point ID="http://gw.foo.org/sensor1":

— http://reg.foo.org/reg_service <--(request) -- lookup("Search query for GW that manages point ID=\"http://gw.foo.org/sensor1\.")

— http://reg.foo.org/reg_service --(response)--> It is "http://gw.foo.org/gw_service"

Case G: Retrieval of Data from GW by APP:

— http://gw.foo.org/gw_service <--(request) -- query("<query>Values select time gives maximum for point ID=\"http://gw.foo.org/sensor1\"</query>")

— http://gw.foo.org/gw_service --(response)--> "<query />"+"<point id=\"http://gw.foo.org/sensor1\"><value>...</value>...</point>"

Case H: Notification of Data from GW to APP:

a) APP puts event query to GW periodically

— http://gw.foo.org/gw_service <--(request) -- query("<query>post data to \"http://app.foo.org/app_service\" if point id=\"http://gw.foo.org/sensor1\" has updated.< /query>")

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

— http://gw.foo.org/gw_service --(response)--> "<OK />"

b) GW submits updates to APP

— http://app.foo.org/app_service <-- (request) -- data("<query>" + "<point id=\"http://gw.foo.org/sensor1\"><value time="2009-09-02T01:00:00+09:00">30.2</value></point>")

— http://app.foo.org/app_service --(response)--> "<OK/>"

Case I: Writing of Data from APP to GW:

— http://gw.foo.org/gw_service <-- (request) -- data("<point id=\"http://gw.foo.org/actuator1\"><value>true</value></point>")

— http://gw.foo.org/gw_service -- (response)-->"<OK/>"

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved

Annex B

(informative)

Typical facility networking system implementation

Figure B.1 shows a typical implementation of a facility networking system. {A, B, ..., G} are components, and {a, b, ..., h} are Points.

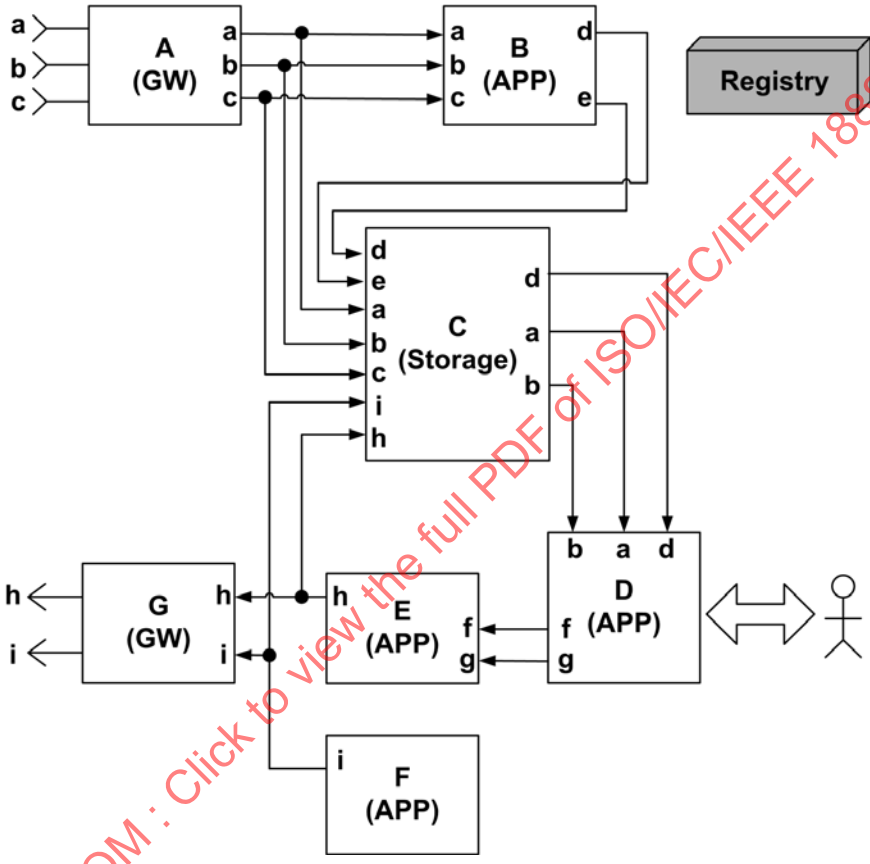

Figure B.1—A typical facility networking system implementation

Each component has following roles:

- A works as a GW for {a, b, c} physical sensors.
- B works as an APP (data processing component) that reads {a, b, c} and generates {d, e}.
- C works as a Storage for {a, b, c, d, e, h, i}.
- D works as an APP (provides user interface) that reads {a, b, d} and generates {f, g}.
- E works as an APP (data switch component) that reads {f, g} and generates {h}.
- F works as an APP (timer) that generates {i}.
- G works as a GW for {h, i} physical actuators.

The roles' information is stored in the Registry, and components refer to the registry to find its collaborative partner to actually send and receive values.

Copyright © 2014 IEEE. All rights reserved.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

© IEEE 2015 – All rights reserved

Annex C

(informative)

Tutorial of query and data method

C.1 FETCH protocol (i.e., query type="storage")

C.1.1 First query method invocation

```
<transport>
 <header>
  <query id="6229c37f-970d-9292-83e4-7c0e54733f8a"
         type="storage"
         acceptableSize="20">
   ... SEARCH_KEY ...
  </query>
 </header>
</transport>
```

C.1.2 Response

```
<transport>
 <header>
  <query id="6229c37f-970d-9292-83e4-7c0e54733f8a"
         type="storage"
         acceptableSize="20"
         cursor="dab751ed-0133-4ce4-8b7d-ba5c54ce4fb5"> <!-- cursor is
given, so there are succeeding dataset. -->
 ... SEARCH_KEY ...
   </query>
   <OK />
 </header>
 <body>
  <point id="...">
   <value time="...">...</value>
   <value time="...">...</value>
   <value time="...">...</value>

   ...
  </point>
 </body>
</transport>
```

C.1.3 Second query method invocation

```
<transport>
 <header>
  <query id="6229c37f-970d-9292-83e4-7c0e54733f8a"
       type="storage"
       acceptableSize="20">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 18880:2015

Copyright © 2014 IEEE. All rights reserved.

© IEEE 2015 – All rights reserved