IEC 61970-405

Edition 1.0 2007-08

INTERNATIONAL STANDARD

Energy management system application program interface (EMS-API) – Part 405: Generic Eventing and Subscription (GES)

IEC 61970-405:2007(E)

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

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

- Catalogue of IEC publications: www.iec.ch/searchpub

The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub

Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv

If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

IEC 61970-405

Edition 1.0 2007-08

INTERNATIONAL STANDARD

Energy management system application program interface (EMS-API) – Part 405: Generic Eventing and Subscription (GES)

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE U

ICS 33.200

ISBN 2-8318-9252-X

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

ENERGY MANAGEMENT SYSTEM APPLICATION PROGRAM INTERFACE (EMS-API) –

Part 405: Generic Eventing and Subscription (GES)

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 61970-405 has been prepared by IEC Technical Committee 57: Power systems management and associated information exchange.

The text of this standard is based on the following documents:

FDIS	Report on voting
57/888/FDIS	57/907/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

A list of all parts of the IEC 61970 series, under the general title *Energy Management System Application Program Interface (EMS-API)*, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the maintenance result date indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

A bilingual version of this publication may be issued at a later date.

INTRODUCTION

This part of IEC 61970 is part of the IEC 61970 series that defines Application Program Interfaces (APIs) for an Energy Management System (EMS). The IEC 61970-4XX and IEC 61970-5XX series documents comprise Component Interface Specifications (CISs). The IEC 61970-4XX series CIS are specified as Platform Independent Models (PIMs), which means they are independent of the underlying technology used to implement them. PIM specifications are also referred to as Level 1 specifications. The IEC 61970-5XX series CIS, on the other hand, are specified as Platform Specific Models (PSMs). PSM specifications are also referred to as Level 2 specifications.

IEC 61970-4XX CISs specify the functional requirements for interfaces that a component (or application) should implement to exchange information with other components (or applications) and/or to access publicly available data in a standard way. The component interfaces describe the specific event types and message contents that can be used by applications for this purpose.

IEC 61970-405 specifies an interface for the efficient transfer of event messages and alarm acknowledge messages in a distributed environment. Small numbers of messages are transferred with short delay but also large amounts are transferred in short time but with possibly longer delay. This is a typical requirement for a SCADA system that acts as a real time data provider to other sub-systems. Other systems than SCADA may also benefit from the characteristics of Generic Eventing and Subscription (GES) interface. When short delay times as well as bulk message transfer is required, GES is a good fit.

The component interface specifications refer to entity objects for the power system domain that is defined in the IEC 61970-3XX series, including IEC 61970-301.

ENERGY MANAGEMENT SYSTEM APPLICATION PROGRAM INTERFACE (EMS-API) –

Part 405: Generic Eventing and Subscription (GES)

1 Scope

The IEC 61970-405 Generic Eventing and Subscription (GES) specification specifies a generalized interface for efficient exchange of messages. The specification takes into account the latencies caused by a Local Area Network (LAN) providing efficient data exchange also over Local Area Networks. The Generic Eventing and Subscription (GES) API is expected to provide one of the primary means for accomplishing application integration. Beyond the scope of the GES API, other APIs address the high performance, real-time interactive needs of an application within a running system as well as request/reply oriented generic data access.

IEC 61970-405 is derived from the Object Management Group (OMG) Data Acquisition from Industrial Systems section Alarms and Events (DAIS A&E) specification. OMG DAIS A&E relies on the OMG Data Access Facility (DAF) and OPC Alarms and Events (A&E) specifications. OMG DAIS A&E is a Platform Specific Model (PSM) with CORBA as the platform and OPC A&E is a PSM with Microsoft COM as the platform. Implementers wanting an introduction to OMG DAIS A&E and OPC A&E shall read these documents.

The GES interface is intended to interoperate with other IEC 61970 based interfaces. Hence it is possible to use information retrieved from other interface to access the same information using this interface, for example:

- object identifiers,
- attribute names or identifiers,
- class names or identifiers.

The way data is organized in a server implementing the GES interface can be seen by using the browse interfaces for data and meta data. It is also possible to use the data access interface directly without using the browse interfaces if the client has an *a priori* knowledge of object, class and attribute identifiers. Object identifiers may be retrieved using data from other interfaces, for example a CIMXML file or the IEC 61970-404 interface. Information on what classes and attributes are available will be described in IEC 61970-45X documents.

IEC 61970-405 describes the functionality in a technology independent way, it is a Platform Independent Specification (PIM). Hence, it explains the functionality to a level that can be used to create PSMs or be an introduction to existing PSMs, i.e. DAIS A&E and OPC A&E. Implementers wanting an introduction to OMG DAIS A&E and OPC A&E should read these documents.

IEC 61970-405 consists of two parts:

- SCADA alarms and events that is the Platform Independent Specification (PIM) derived from DAIS A&E and OPC A&E. This part is called “Generic Eventing and Subscription Alarms and Events” (GES A&E).
- Generic messaging that is a generalization of the SCADA alarms and events. This part is just called “Generic Eventing and Subscription” (GES).

IEC 61970-1 provides the EMS-API reference model upon which this standard is based. In that reference model, the terminology used in this part of IEC 61970 is introduced and the role of the CIS is explained.

IEC 61970-401 provides an overview and framework for the CIS (IEC 61970-4XX) standards.IEC 61970-402 provides the base services to be used in conjunction with other IEC 61970-4XX documents. This specification extends the Common Services to provide an event subscription oriented mechanism for applications to exchange CIM data.

The mapping of IEC 61970-405 to implementation specific technologies or Platform Specific Models (PSMs) is further described in a separate series of documents, i.e. the future IEC 61970-5XX. For actual implementations, the future IEC 61970-5XX, OMG DAIS A&E, OMG DAF or OPC A&E are used.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61970-1:2005, *Energy management system application program interface (EMS-API) – Part 1: Guidelines and general requirements*

IEC/TS 61970-2, *Energy management system application program interface (EMS-API) – Part 2: Glossary*

IEC 61970-301:2005, *Energy management system application program interface (EMS-API) – Part 301: Common Information Model (CIM) base*

IEC 61970-401, *Energy management system application program interface (EMS-API) – Part 401: Component Interface Specification (CIS) Framework*

IEC 61970-402, *Energy management system application program interface (EMS-API) – Part 402: Component Interface Specification (CIS) – Common Services*

Data Acquisition from Industrial Systems section Alarms and Events (DAIS A&E), OMG Adopted Specification Version1.1, formal/2005-06-01 June 2005 (Referred herein as 'OMG DAIS A&E')

Utility Management System (UMS) Data Access Facility (DAF), OMG Adopted Specification, Version 2.0.1, formal/05-06-03, July 2005 (Referred to herein as 'OMG DAF')

OPC Alarms and Events Specification, Version 1.10, OPC Foundation, October 2002 (Referred to herein as 'OPC A&E')

3 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC/TS 61970-2 apply.

NOTE Refer to International Electrotechnical Vocabulary, IEC 60050, for general glossary definitions.

4 Generic Eventing and Subscription (Normative)

4.1 Overview

4.1.1 General

Figure 1 illustrates the interaction between a Generic Eventing and Subscription Alarms and Events (GES A&E) client and server. A subscription means that the server has no *a priori* knowledge of its clients. Once a connection is established, a server calls the clients back when data becomes available or updated.

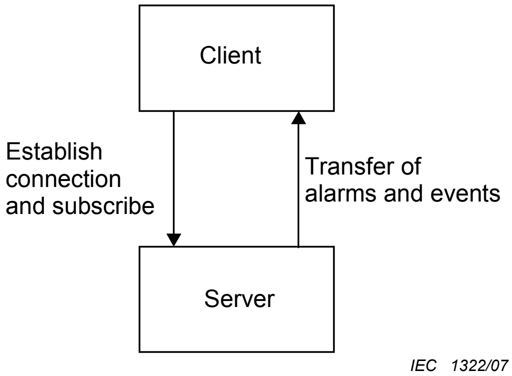

Figure 1 – Data subscription

This Clause discusses Generic Eventing and Subscription (GES) that is a generalization of the Generic Eventing and Subscription Alarms and Events (GES A&E). GES A&E is described in Clause 5. The scope of GES A&E is intended for SCADA data exchange, as described in Clause 5. GES A&E Simple Events can be applied to a much broader scope and is the basis for GES.

4.1.2 Suitability of the GES to the integration of a variety of application categories

The use of GES is not limited to SCADA oriented data. Application categories exchanging non SCADA data that GES can be applied to include:

- Outage Management Systems
- Network Applications
- Generation Control
- Geographic Information Systems
- Energy Management Systems
- Asset and Work Management Systems
- Other application categories used in the operation of a power system as listed in IEC 61970-1 (2005) Annex B.

4.1.3 Suitability of the GES to the integration beyond the control center

Though the target of this IEC standard is the control center technical domain, GES encompasses a general set of concepts that can be applied to many types of systems. Examples of these systems include:

- Customer Information Systems
- Substation Automation Systems. For integration with Substation Automation Systems, it shall be noted that several specifications related to communication already exists, for example the IEC 61850 series for substation communication and IEC 60870-5 for RTU communication. The intention of this part of IEC 61970 is to describe a service interface that may encapsulate such communication solutions.
- Other types of technically oriented operational business systems.

In recognition that the integration between applications in two or more of these systems is often necessary, the intent of this part of IEC 61970 is to meet general GES requirements to the extent that they are common to different types of systems while effectively addressing control center needs.

4.1.4 Suitability of the GES to the integration of tightly coupled and loosely coupled applications (informative)

Over the last couple of years, the software community has come to realize that different software architectures are required when integrating tightly coupled applications (for example transmission network applications deployed in a utility control center) versus loosely coupled applications (e.g. distribution network application applications deployed as Web Services). In the case of tightly coupled applications, the integrating architecture might be based on CORBA IIOP, Java RMI, Microsoft DCOM or even a centralized database. In the case of loosely coupled integration, however, it is generally accepted that an XML-based messaging architecture is preferred. One goal of the GES is to allow a component to be developed without prior knowledge of the deployment environment. Thus, for example, components with a GES interface may be tied together via a CORBA remote procedure call based or an XML message bus based infrastructure.

In accordance with IEC 61970-1, the GES is a component interface and does not include access to middleware specific functionality such as messaging Quality of Service (QOS). Even though the GES does not provide a message broker-oriented interface, the GES can be used as an interface for a component adapter that in turn accesses a message broker-based interface such as JMS or OAMAS for example.

GES specializations are possible for the following interface technologies, some of which are language-specific and some of which are middleware-specific:

- C++ language.
- C language.
- CORBA.
- COM.
- Java.
- XML. The XML specialization will provide interoperability between independently developed components with a GES interface when XML based messaging is used as the integration technology. The exact format of this XML will be determined by the use of a specification such as the W3C's XML Protocol.

Figure 2 below illustrates how a component with a language or interface technology-specific specialization of the GES Interface can be adapted to run over a middleware-specific implementation. Thus, in cases where an application component has been developed using a particular language-specific specialization such as CORBA, the use of a component adapter would allow components to communicate using a particular middleware such as XML messaging. Thus, the GES is independent of what transport technology is used for example IIOP or XML Protocol.

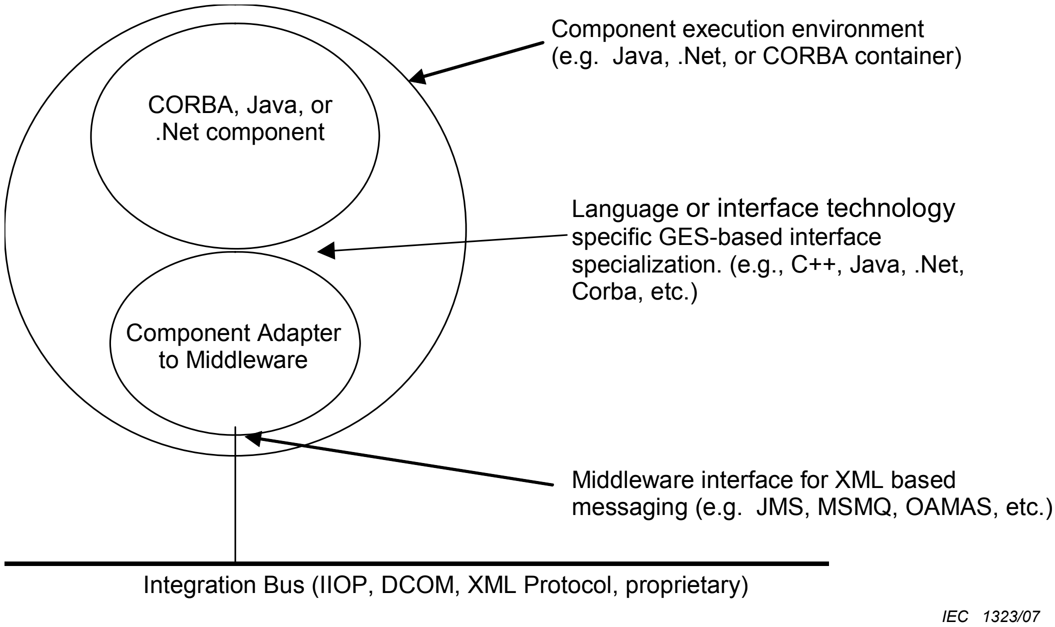

Figure 2 – Component architecture

Deployment environment independence is a key facet of component interoperability as well as a way of lowering the cost of component development. For example, this specification provides a common way of subscribing to and filtering of small object (also called business object) events. COM does not provide any standard way of subscribing to events related to a particular small object (such as a particular breaker). Consequently, when running over a Microsoft-based infrastructure, it is often up to the system integrator to supply this functionality. Enabling subscribing component developers to add small object event filtering functionality “out of the box” (as is done in off-the-shelf OPC clients) lowers the time and effort required to integrate components.

Enabling small object filtering in the component does not in any way break the contract between components, component adapters, and component execution systems. The use of small object- based filtering does not define a middleware interface. Proof of this can be seen in that OPC and DAIS both provide small object-based event filtering. DAIS/OPC are not used as middleware interfaces. Rather they are used at the component interface. Middleware API’s, such as JMS and CORBA Notification, include transport semantics such as QOS (e.g. delivery characteristics such as reliability, priority, and expiration time) as well as administrative functionality (e.g. event channel load balancing). QOS and administrative functionality is specific to the middleware used. On the other hand, small object filtering is common to any middleware used.

The use of publish/subscribe functionality exposed at a component interface means that event filtering code does not need to be rewritten for each different implementation scenario. Event filtering is done by a component and not by a component adapter; consequently, power system specific (CIM-based) event filtering tools can be used instead of generic event filtering tools. CIM-based event filtering provides for more familiar as well as more powerful and easy to use configuration management.

Furthermore, since the GES API can be deployed independent of any application category specific information model, GES component interfaces do not need to be recompiled when an information model changes. Thus, application category information model changes can be handled via declarative action rather than programmatic action.

4.2 Using GES A&E Simple Eventing for Generic Messaging

4.2.1 General

GES can be used to construct a document exchange system. While the High Speed Data Access (HSDA) interface as described in IEC 61970-404 supports subscribing to measurement, parameter and other simple data, GES A&E interface as described in Clause 5 allows components to publish and subscribe to non-measurement documents (e.g. work orders) as opposed to collections of measurement points. The eventing interface allows components to publish and subscribe to documents such as work orders without having to subscribe to each individual work order property.

More precisely, the use of "Simple Events" in the GES A&E interfaces is not limited to SCADA data. GES A&E Simple Events may contain a document-oriented payload that can be sent to or received by an arbitrary control center application. In this way, GES A&E Simple Events are useful for accessing the entire Common Information Model (CIM). The data transmitted through GES interfaces is independent of the GES Interface Definition Language (IDL). That is, the GES can be used to expose and transfer any information exchange model.

The schema for GES A&E Simple Events is generic. GES A&E Simple Events have the following properties:

- Source – a reference to the object that generated the event. This can be either an item such as a particular measurement or more generally the name of CIM class as well as an application category.
- Source pathname – the fully qualified IECTC57PhysicalModel pathname for the source.
- Time stamp – the time that the event occurred.
- Message – an text string describing the event.
- Main category – for Simple Events, this field contains the resource ID of Simple Event.
- Category – the resource ID for the event type. Every category corresponds to a specific event schema.
- Category name – the name of the event type.
- Severity – a hint from the server about the level of severity for this event.
- Properties – a list of mapped and/or indexed properties.

For example, the Source field could contain a reference to the Work Management System, the Message field could contain "New Work Order", the Category could contain the resource ID for the "Transformer Installation Request" event type, the Category name field could contain "Transformer Installation Request", and the properties would be that data associated with a Transformer Installation Request. As can be seen by this example, Simple Events are general purpose in nature – that is, they can be used by any application category to exchange information.

4.2.2 GES messages

The Common Information Model described in IEC 61970-301 defines a normalized data model for use in integration of control center applications. Future IEC 61970-450 to IEC 61970-499 Component Interface Specifications will provide denormalized views of IEC 61970-301 data. For example, while IEC 61970-301 objects such as transformers and loads are distinct classes, IEC 61970-450 to IEC 61970-499 Component Interface Specifications will define document style interface classes that may amalgamate a few properties from multiple normalized classes such as transformers and loads. IEC 61970-450 to IEC 61970-499 CIS documents (event types) will provide application specific ways of grouping or viewing a collection of properties.

4.2.3 Eventing and Subscription topic trees (informative)

Events can contain documents that are delivered asynchronously. Messages can convey documents, but the use of the term “message” also implies a broker-oriented transport mechanism. Thus, the term event is more generic than message.

GES Eventing is similar to the Java Messaging Service (JMS) Interface with one important distinction; GES Eventing does not include access to messaging-specific functionality such as message broker service QOS. However, since GES Eventing is a component interface that is independent of what middleware is deployed between components, the lack of a messaging specific interface is appropriate.

The JMS Publish/Subscribe model defines how components use JMS to publish and subscribe to messages related to a well-known node in a content-based hierarchy. GES Eventing calls these nodes Areas and JMS calls these nodes topics. By relying on the area/topic as an intermediary, event/message publishers are kept independent of subscribers and vice versa. Many messaging service providers, such as MQSeries, group topics into hierarchies and provide various options for subscribing to parts of the hierarchy. GES Area strings consist of an array of area names (fully qualified pathnames) created as a result of navigating a hierarchical area map. Neither GES Eventing nor JMS places any restriction on what an area/topic object represents. It might be a leaf in a topic hierarchy or it might be a larger part of the hierarchy (for subscribing to a general class of information).

Frequently, users like to see data organized in accordance with a well-known topic hierarchy. GES does not standardize the presentation of well-known (e.g. CIM based) hierarchical views. IECTC57 Namespaces as described in IEC 61970-402 Common Services specification is used with GES in the same way that OPC Batch is used with OPC Data Access and OPC Alarms and Events by providing a means by which servers can present well-known hierarchical views. The use of the IECTC57 Namespaces together with the IArea/ICategory/IType (refer to Figure 7) interfaces provide the base for CIM-oriented subscription. Thus, GES with Common Services provide a standard mechanism for describing how CIM objects are ordered in a hierarchy as well as a way for components to use those hierarchies when setting up event (document) subscriptions.

4.2.4 Subscription filtering

GES subscription filtering provides the ability to specify property values that are used to qualify a subscription. In SQL terms, this addition to the GES Filtering is equivalent to adding a “where” clause. For example, a subscriber can indicate that it wishes to receive New Transformer Installation events as well as indicate that it only wants to receive New Transformer Installation events pertaining to transformers produced by a given manufacturer.

A GES Subscription object (refer to Figure 7) maintains a filter specification set up by a client. The filter is used to specify what notifications shall be sent to the client. A server may support various filter functions and a client can ask the Subscription Home object what filter functions are supported. The subscription home is also used to create any number of subscription manger objects. Each subscription manger is associated with a client that has implemented callback object so that the server can send alarm and event notifications to the client.

4.2.5 GES Publish/Subscribe wildcards

4.2.5.1 General

GES Eventing does not support “wild card” functionality in topic subscription strings. For example, GES Eventing allows components to subscribe to events on the basis of event Category (e.g. “Transformer Installation Request”) or on the basis of topics (e.g. all work orders for breakers), but there is no facility to choose all areas at a given level. This specification extends GES Eventing with support for wildcards in topic based filtering. The GES Events interfaces are not changed in order to add support for wild cards in area paths.

This specification requires a well-known URI for the wildcard. The URI for the wildcard is: http://omg.org/schema/GES#WILDCARD

4.2.5.2 Example topic hierarchies (informative)

To illustrate how subscription topic trees are used, consider a topic tree that consists of a CIM containment model hierarchy as shown in Figure 3. An example area string created by navigating the above tree might be "IECTC57PhysicalModel/NorthArea/AirportSubstation". In this case, a subscriber could receive all events that pertain to this topic – including breaker, transformer, and load-related events. However, if one wants to subscribe to breaker related events from all substations, then the GES requires that the user specifically includes the name of each substation node one may be interested in. This specification includes the added capability to create an area string (array of area names) without having to manually navigate to every node on is interested in. Such a topic string may appear as "IECTC57 PhysicalModel/*/*/Breakers".

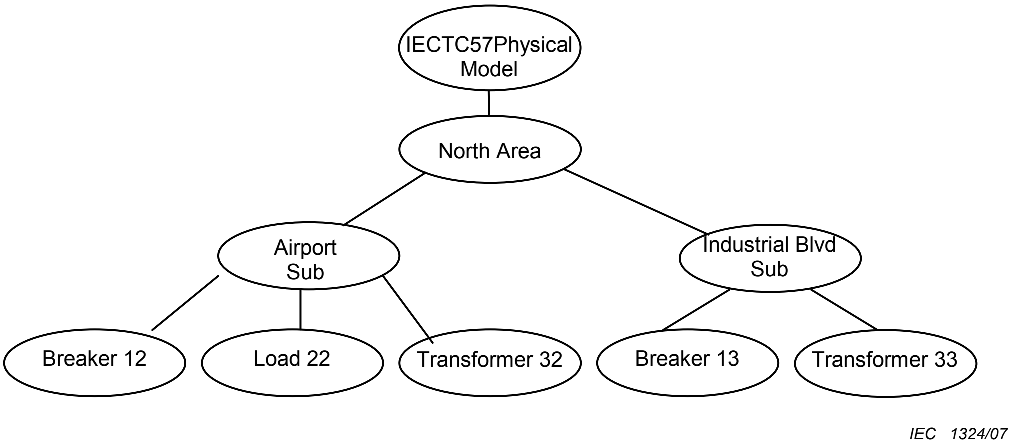

Figure 3 – Populated namespace example for IECTC57PhysicalModel (informative)

4.2.6 Browsing IECTC57 namespaces with GES

4.2.6.1 General

IEC 61970-402 Common Services defines three trees or sub-trees used for topic based subscriptions:

- IECTC57PhysicalModel
- IECTC57ClassModel
- IECTC57ISModel

IECTC57PhysicalModel contains subscription topic nodes representing the containment hierarchy of instances of CIM classes, for example control areas, substations, bays, breakers, etc. To browse this hierarchy, GES uses the IArea browse interface (refer to Figure 7).

IECTC57ClassModel contains subscription topic nodes representing the class hierarchy of CIM classes, for example the Power System Resource class, AC Conducting Equipment class, Switch class, Breaker class, etc. To browse this hierarchy, GES uses the IType browse interface (refer to Figure 7).

IECTC57ISModel contains subscription topic nodes representing the hierarchy of events types. To browse this hierarchy, GES uses the ICategory interface (refer to Figure 7).

Using OPC with GES, clients browse all subscription topic nodes with Area browsing by putting them into the same tree. The single OPC namespace tree may be divided into the three sub trees: IECTC57PhysicalModel, IECTC57ClassModel, and IECTC57ISModel. As data describing the properties is different depending on the sub-tree in question, the GES approach is to have sub-tree specific browsers returning data specific for the sub-trees, i.e. IArea and IType and ICategory (refer to Figure 7) for the IECTC57PhysicalModel, IECTC57ClassModel, and IECTC57ISModel topic trees respectively. Therefore, GES browsing does not required three sub trees. Using GES, IECTC57PhysicalModel, IECTC57ClassModel, and IECTC57 ISModel are the roots of independent topic trees.

4.2.6.2 Mapping GES browsing to OPC browsing

4.2.6.2.1 General

In order to compensate for the lack of native support for Resource ID's and Type in OPC, special properties shall be introduced into a GES compliant OPC namespace.

4.2.6.2.2 Custom property required for Resource ID

OPC differs from DAIS in that only names can be used to subscribe to events. In order to facilitate a consistent mapping from OPC to DAIS, every node in an OPC implementation of the IECTC57 Namespaces shall have a custom property defined named “Resource ID”. This property contains the value of the Resource ID for that object. The OPC Custom Property Index for this property shall be 10 000. The format of this property shall be an ASCII string representation of hex digits: {xxxxxxxx-xxxxxxxx-xxxxxxxx-xxxxxxxx}.

4.2.6.2.3 Custom property required for Resource Type

OPC differs from GES in that only names can be used to subscribe to events. In order to facilitate a consistent mapping from OPC to GES, every node in an OPC implementation of the IECTC57 Namespaces shall have a custom property defined named “Resource ID”. This property contains the value of the Resource ID for that object. The OPC Custom Property Index for this property shall be 10 001. The value of this custom property for items in the IECTC57PhysicalModel shall be the same as GES Type in the IECTC57ClassModel.

4.2.6.2.4 Custom property required for Type Resource ID

OPC differs from GES in that only names can be used to subscribe to events. In order to facilitate a consistent mapping from OPC to GES, every node in an OPC implementation of the IECTC57PhysicalModel Namespace shall have a custom property defined named “Type Resource ID”. This property contains the value of the Resource ID for the type of that object. The OPC Custom Property Index for this property shall be 10 002. The format of this property shall be an ASCII string representation of hex digits: {xxxxxxxx-xxxxxxxx-xxxxxxxx-xxxxxxxx}.

4.2.6.2.5 Custom property required for Generic Data Access Type

OPC differs from GES in that only names can be used to subscribe to events. In order to facilitate a consistent mapping from OPC to GES, every attribute node in an OPC implementation of the IECTC57 Namespaces shall have a custom property defined named “Generic Data Access Type”. This property identifies the GDA Simple Value data type of the attribute. The OPC Custom Property Index for this property shall be 10 003.

4.2.6.2.6 Custom property required for Modelled

OPC differs from GES in that only names can be used to subscribe to events. In order to facilitate a consistent mapping from OPC to GES, every attribute node in an OPC implementation of the IECTC57ISModel Namespace shall have a custom property defined named “Modelled”. This property indicates if the attributes can be mapped to a resource ID using the ResourceIDService. For attributes that can be mapped, the value of this property is the ASCII string “Mapped”. For attributes that cannot be mapped, the valued of this property is the ASCII string “Indexed”. The OPC Custom Property Index for this property shall be 10 004.

5 Generic Event Subscription Alarms and Events

5.1 Background (informative)

For historical reasons, control systems for different industrial processes have evolved along different lines. Control systems for power systems have evolved on a UNIX base and control systems for most other industrial processes have evolved on a Windows base. For Windows based control systems, OPC has become the dominating standard. For UNIX based systems, the DAIS API defined in Common Object Request Broker Architecture (CORBA) Interface Definition Language (IDL) has been developed. DAIS is based on OPC to benefit from the success of OPC and enable easy bridging to OPC. Whith this intent, Object Management Group (OMG) started in 1997 to develop a CORBA based interface with the same functionality as OPC. GES A&E has the functionality from OMG DAIS A&E and OPC A&E described in a technology neutral way, hence GES A&E is intended to be a Platform Independent Model (PIM).

5.2 SCADA use case (informative)

Alarms are generated with the intent to gain attention to fault conditions that need operator intervention. Hence alarms are recorded and presented to operators. The presentation usually involves acoustic annunciation and some highlighting, for example red colour and/or blink etc. Events are more general and may be generated due to various reasons and are distributed to clients that have interests in the specific information topics. GES A&E can be used to monitor and control a power system consisting of the following major parts:

- Process instrumentation making sensor data and actuation capabilities available.
- Remote terminal units (RTUs) or substation control systems reading sensor data and controlling actuators.
- Process communication units connecting to RTUs or substation control systems.
- SCADA subsystem making processed sensor data and control capabilities available to operators, applications or other systems.
- An Energy Management System (EMS) using the SCADA subsystem for extended processing and control.

In a SCADA/EMS, the SCADA subsystem typically takes care of the presentation of alarms and events related to field devices. In such a system, the data flows as shown in Figure 4.

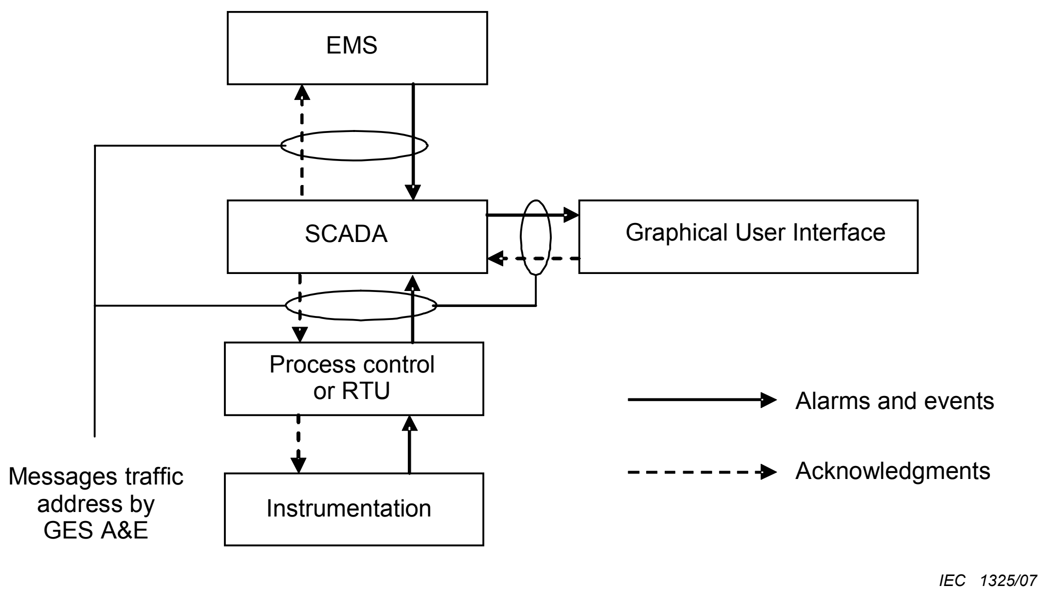

Figure 4 – Typical SCADA flows of DAIS alarms and events

Alarms and Event servers may be organized in a hierarchy where simple servers generate alarms that are collected by servers with more advanced functions, for example filtering and storage of alarms.

The DAIS Alarms and Events interface can be used for several different purposes:

- Presentation of alarms or events from a process in event lists.
- Printing of alarms and events on paper.
- Archiving of alarms and events.
- Collection of alarms and events from simple sources to more advanced alarms and events management servers.
- Acknowledgement of alarms.

Two main actors appear in the use cases

- A maintenance engineer that configure alarms and event subscriptions.
- An operator that calls up alarm lists for presentation, reviews alarms printed on paper or acknowledge alarms. An operator may also filter the alarms that are presented in lists. It is supposed that a GUI that aids specification of filter parameters exists in this case.

5.3 Data model (normative)

The GES A&E data model describes how data seen through the GES A&E interface appears to be organized within a server. A server implementation may organize data differently, but a client using GES A&E will see a data model as shown in Figure 5.

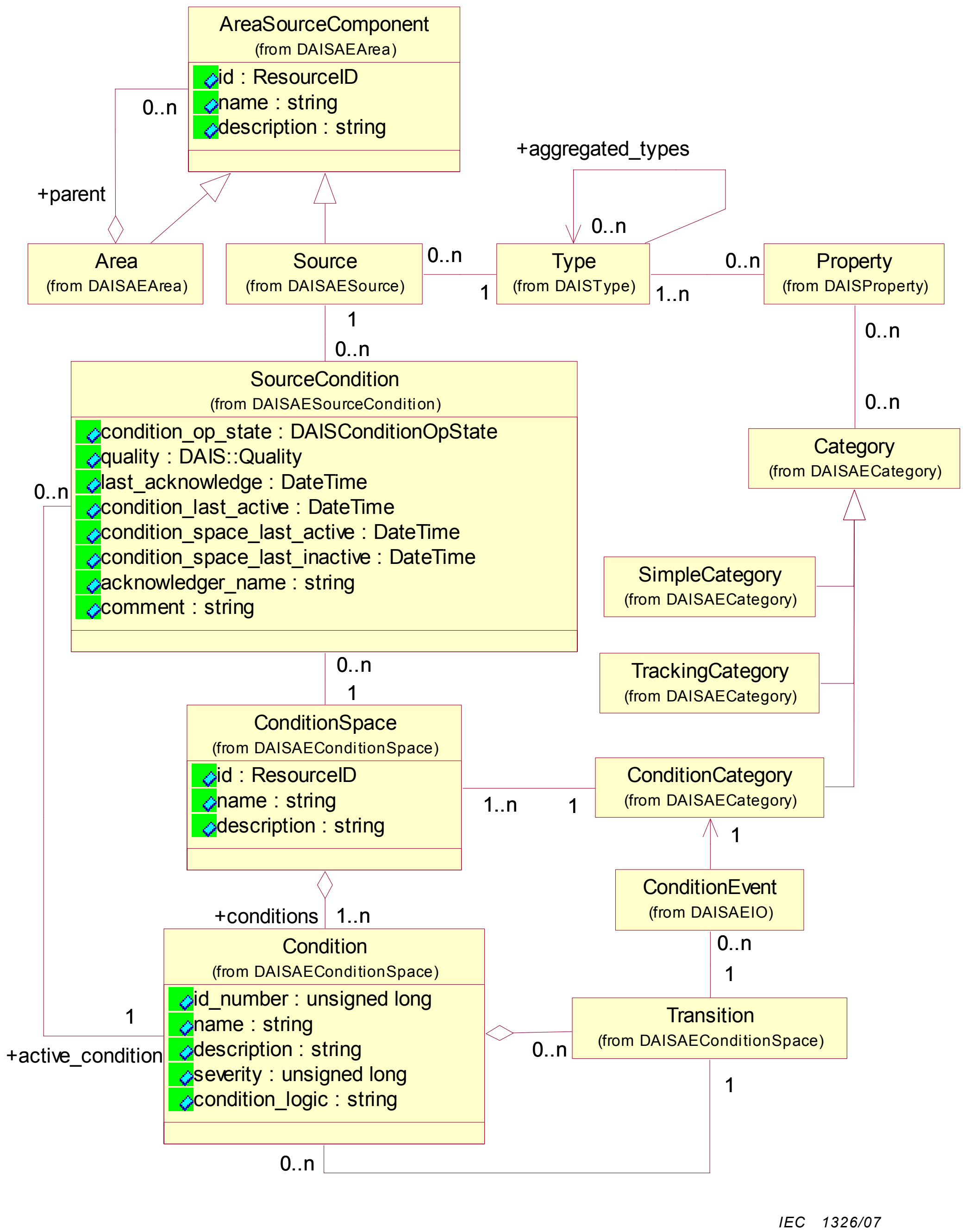

Figure 5 – GES A&E data model

Major objects that appear in alarms and events server are:

- Areas that organize Sources in a hierarchical structure.
- Sources that are the objects that are alarmed.
- Categories that is categorization of the alarm processing.
- ConditionSpaces that defines alarm processing types.
- Conditions that define alarm states.
- Type define the type of a source.

- Property define the properties that are defined for a Type and that may be included in Categories.
- Subscriptions and Filters.

A Source refers to the originator of an event, for example:

- Breaker position, analog measurement, a tank, a generator etc.
- An application program, for example a Customer Information System.

Areas are used to create hierarchical organizations of Sources. Areas are typically used to support operator authority. An Area can contain other Areas or Sources. The client can filter event subscriptions by specifying the areas to limit the event notifications received.

Categories are used to describe the type of alarm processing. The following basic categories are defined

- Simple, which is used for events without alarm.
- Tracking, which is used to track operator actions, also without alarm.
- Condition, which is used for alarms.

All three can be further specialized into sub-categories.

A ConditionSpace defines an alarm processing type, for example a tank level with five limit values, a breaker with the positions open, closed and intermediate.

A Condition defines a state that belongs to a ConditionSpace. A state is described by a condition when it is active. It also describe the alarming that is made when active, for example the severity. Each Condition is described by rules describing the value range, for example

- “Tripped” when breaker position = open
- High Alarm condition where: value > high alarm limit.
- “Disk Full” when the Customer Information System has run out of storage space.

A SourceCondition describes the currently active Condition for a Source. As a Source may have several ConditionSpaces defined for it, each ConditionSpace will have one Source Condition for the Source. The SourceCondition has the following data describing the alarm state:

- Enabled, indicating if the ConditionSpace is enabled for the Source.
- The currently active Condition, if enabled.
- Active, which tells if the currently active Condition indicates a fault.
- Alarm, which tells if any unacknowledged alarms exist.

A Type defines the type of Source, for example measurement, breaker, tank, generator etc.

A subscription allows a client to receive events asynchronously. Filters are used to limit what events and alarms a client receives.

5.4 Messages (normative)

A Source may cause the following main message types:

- SimpleEvent, which belongs to the SimpleCategory.
- TrackingEvent, which belongs to the TrackingCategory.
- ConditionEvent, which belongs to the ConditionCategory.

Figure 6 show the message types.

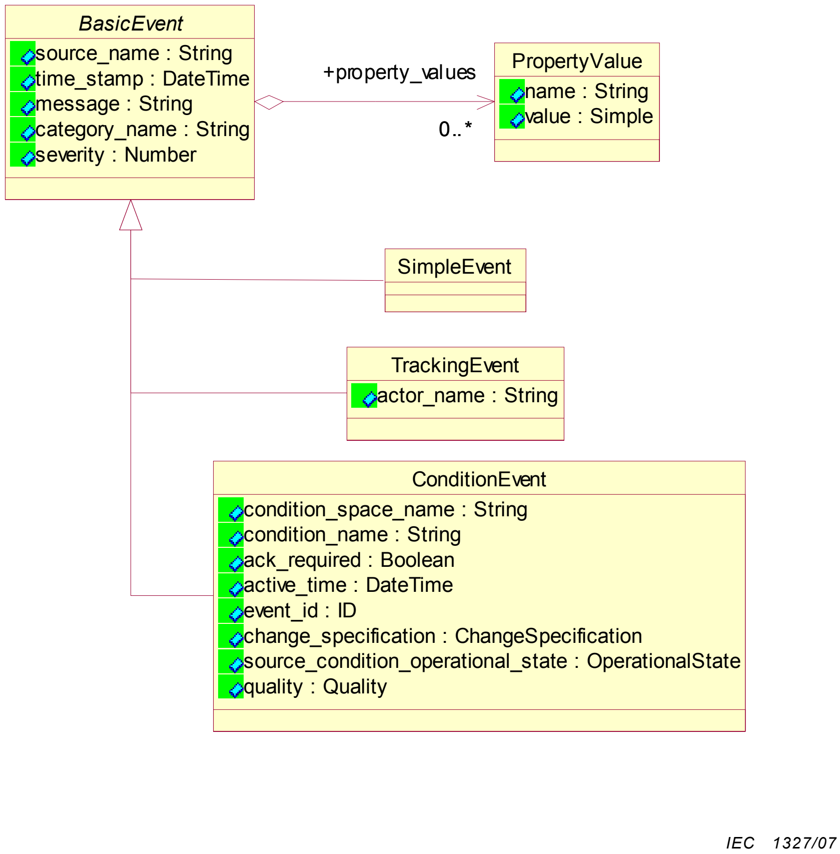

Figure 6 – Event messages

SimpleEvents contain the following data:

- The identity of the Source object.
- A time stamp indicating when the event was created.
- A message text.
- The identity of the Category.
- A severity for the event.
- A custom list of property values, i.e. name value pairs.

TrackingEvents are caused by an actor or operator. Typical actions are data entries, commands or acknowledgements. A tracking event is the same as a SimpleEvent with the following additional data:

- The identity of the actor or operator.

ConditionEvents are alarms that contains the following information in addition to the SimpleEvent:

- The identity of the ConditionSpace.
- The identity of the current Condition, for example "High alarm", "Disk Full".
- Unacknowledged alarm, i.e. the current Condition requires an acknowledgement.

- The time when the current Condition was activated.
- A change specification that indicates what has changed.
- The current alarm state for the Source and the ConditionSpace, i.e. alarming is enabled, unacknowledged alarm exists or a fault exists (a Condition indicating a fault is active).
- Acknowledge identification to be used in a later acknowledgement by an operator.
- The quality of the value causing the alarm.

When an operator acknowledges an alarm, this is reported by a TrackingEvent message containing the operator name.

5.5 Interface (normative)

5.5.1 Objects and interfaces

The object types (e.g. Server, Client, Session etc.) described in this Subclause shall all exist in an implementation if not otherwise noted. It is allowed for an implementation to create more object types than those described below, for example instead of letting an object type implement multiple interfaces a specific object type may implement each interface as in OMG DAIS A&E.

The GES A&E interfaces, objects and their relations are shown in Figure 7.

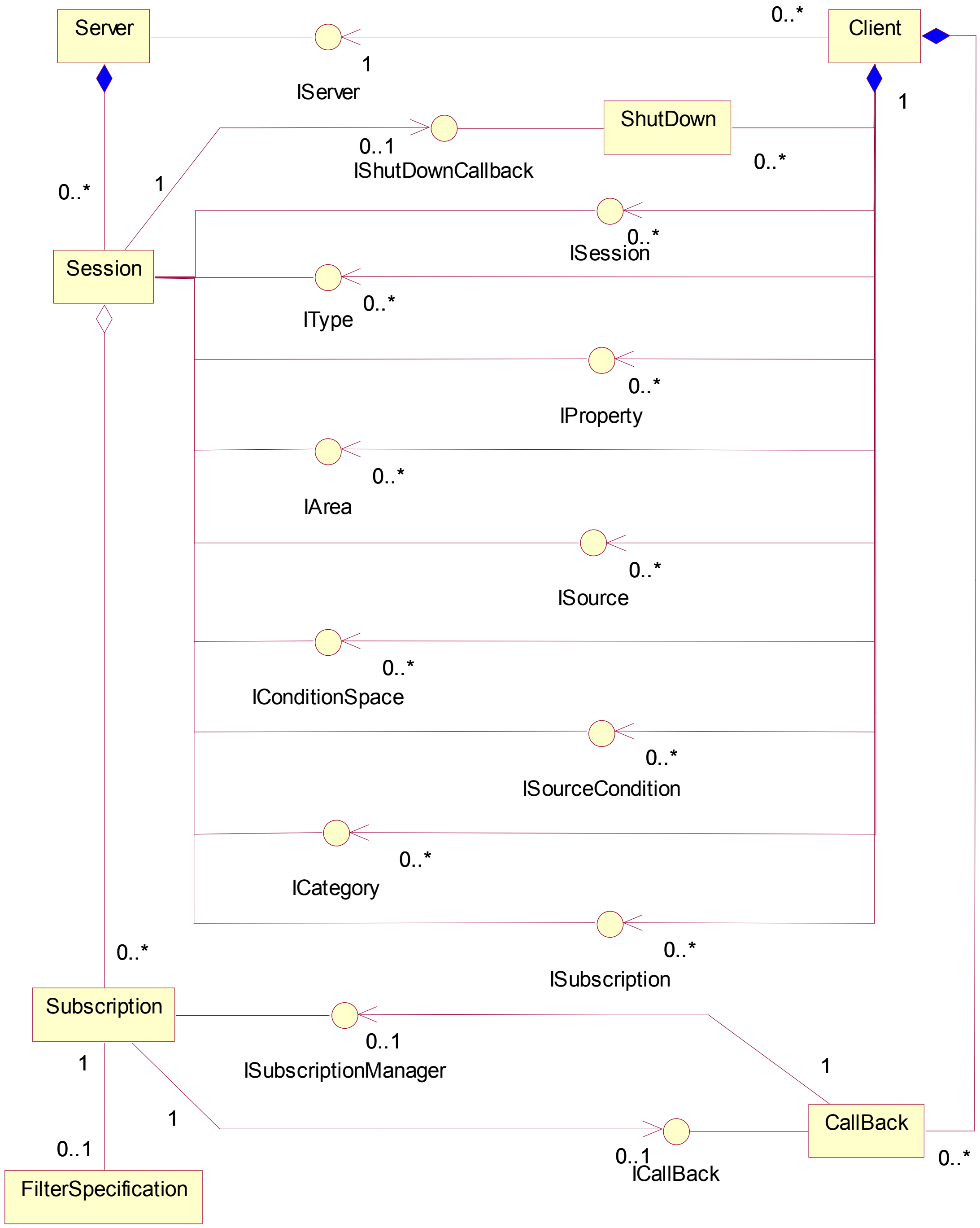

Figure 7 – GES A&E objects and interfaces

A GES A&E Server is an object that may have any number of Clients using it. It has a number of Session objects. Also note that the Server and Session objects may be combined into one object as in OPC A&E.

The browse interfaces (i.e. Type, Property, Area, Source, ConditionSpace, SourceCondition and Category) directly corresponds to the objects in the data model shown in Figure 5. The Session object has a number of interfaces:

- ISession that is used to manage the session.
- IArea browser that is used to get the Area objects.
- ISource browser that is used to get the Source objects.
- IConditionSpace browser that is used to get the ConditionSpace objects and their Condition objects.
- ISourceCondition browser that is used to get the SourceCondition objects.
- ICategory browser that is used to get the Category and sub-Category objects.
- IType browser that is used to find meta-data about the data objects (i.e., classes) implemented by a HSDA server.
- IProperty browser that is used to find meta-data about the object data (i.e., properties) implemented by a HSDA server.
- ISubscription interface that is used to create and Subscription objects.

The Session object has a number of Subscription objects used by Clients to manage the Subscriptions through the ISubscriptionManager interface. Each Subscription is described by a FilterSpecification. The CallBack is an object that implements the IAECallBack interface and is implemented by the client. The CallBack object is used by the server to deliver alarms or events according to the Subscription FilterSpecification. The following FilterSpecification parameters exist and specify what shall be included in a subscription:

- EventFormat, which specifies the message types described in 5.4.
- Categories.
- An upper and lower limit for the Severity.
- Sources.
- Areas.
- Types.

Leaving any of these parameters unspecified means all objects for that parameter shall be included in the subscription.

A typical interaction sequence between a GES A&E Server and a Client is shown in Figure 8.

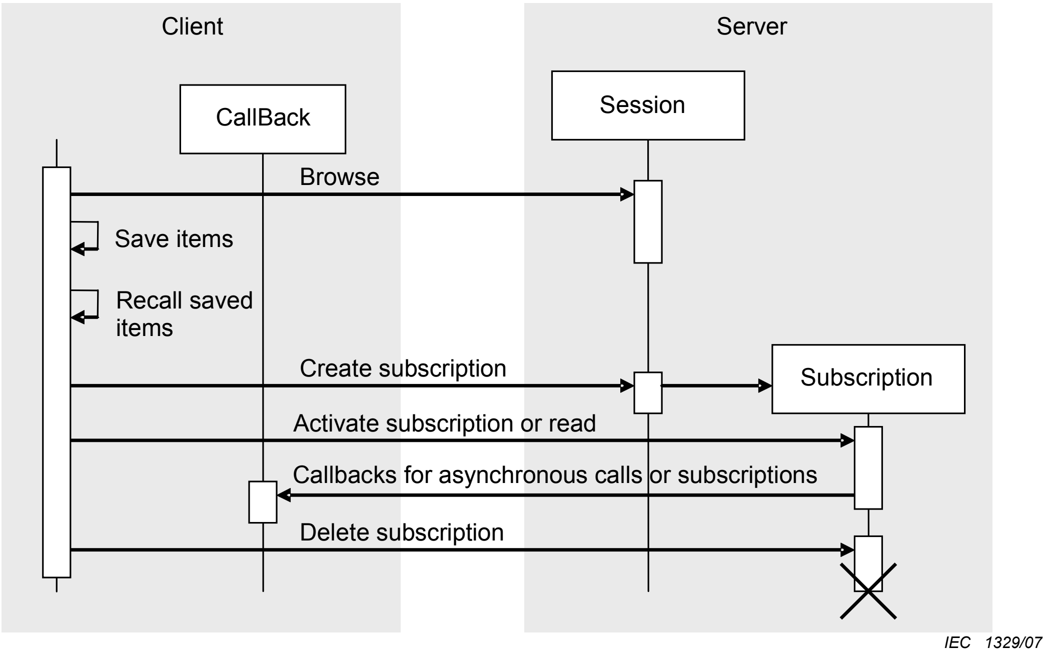

Figure 8 – Typical interaction between the DAIS A&E objects

Figure 8 also show the data alarm and event messages described in 5.4.

A Client will typically start to browse the server to find out what data is available. The browse APIs (Area, Source, ConditionSpace and SourceCondition) are used for this. The retrieved information is used to specify a number of filters. A client will likely have a GUI that helps in creating filter specifications.

At a later time, for example, at alarm or event list call up, the saved filter specifications will be recalled and used to set up a subscription. Exactly how a filter specification is selected is implementation dependent.

5.5.2 Server and Session interfaces

- The IServer interface has the following attributes and methods:create_alarms_ and_events_session()
- create_alarms_and_events_session_for_view(). This method can be used if a server supports multiple hierarchies. Each hierarchy then corresponds to a view.
- find_views() that returns the views that are supported by the server.
- a read only attribute Server status that tells the status of the Server, for example, vendor info, time when started, current time etc.
- a read only attribute that tells the functions supported by the interface.

The IDASession interface has the following attributes and methods:

- a read only status attribute that tells the status of the Session, for example, name, time when started, current time and number of groups.
- an attribute holding the optional ShutDown object.

5.5.3 Browse interfaces

The INode, IType, IArea, ISource and ICategory interfaces have the following common methods:

- find(), which returns more information about one object specified by it's id.
- find_each(), which returns more information about a number of objects specified by their id's.
- find_by_parent(), which returns all children for a parent with a specified id.
- find_by_type(), which recursively returns all children with a given Type.id for a parent with a specified id.
- get_pathnames(), which translates a number of pathnames to the corresponding id's.
- get_ids(), which translates a number of id's to pathnames.

IArea interface has the additional methods:

- enable_condition(), which is used to enable reporting of alarms for all Sources that belong to the Area.
- disable_condition(), which is used to disable reporting of alarms for all Sources that belong to the Area.

ISource interface has the additional methods:

- enable_condition(), which is used to enable reporting of alarms for the specified Sources.
- disable_condition(), which is used to disable reporting of alarms for the specified Sources.
- translate_to_item_ids(), which is used to find the identities of the DAIS DA Nodes that correspond to the specified Sources.

The ICategory interface has the additional methods:

- get_event_properties(), which is used to get the custom properties that a server may add to the SimpleEvent message.

The IConditionSpace interface has the following methods:

- find(), which returns more information about one ConditionSpace specified by it's id.
- find_each(), which returns more information about a number of ConditionSpaces specified by their id's.
- find_by_category(), which returns all ConditionSpaces for the specified Category.
- find_by_source(), which returns all ConditionSpaces for the specified Source.
- get_pathnames(), which translates a number of pathnames to the corresponding id's.
- get_ids(), which translates a number of id's to pathnames.

The ISourceCondition interface has the following methods:

- find(), which returns more information about one SourceCondition specified by it's id.
- find_each(), which returns more information about a number of SourceCondition specified by their id's.
- ack_condition(), which is used to acknowledge a number of alarms where each alarm is identified by the Acknowledge identification from the alarm message.

The ISubscription interface has the following attributes and methods:

- query_available_filters(), which returns the filtering capabilities implemented by the server.
- create_subscription(), which creates a Subscription manager object.

The ISubscriptionManager interface has the following attributes and methods:

- a callback attribute that holds the object provided by the client that created the Subscription object. The client shall update this attribute with the callback object.

- set_filter(), which provides a FilterSpecification to the subscription.
- get_filter(), which gets the current FilterSpecification from the subscription.
- select_returned_properties(), which is used to specify the PropertyValues that shall be included in event messages.
- get_returned_properties(), which is used to get what PropertyValues that are currently included in event messages.
- refresh(), which ask the server to report all currently active or unacknowledged Conditions that matches the FilterSpecification.
- async_read_history() reports the specified number of alarms and events from a given start time.
- cancel(), which is used to abort an on going reporting due to a refresh() or async_read_history().
- get_state(), which is used to get the following subscription information: if it is active, maximum wait time before a message buffer is sent, maximum size of the message buffer before it's messages are sent and min time before sending messages.
- set_state(), which is used to set the above data.
- clone(), which is used to clone the subscription.
- destroy(), which is used to destroy the subscription and it's resources.

5.5.4 Client interfaces

To support subscriptions a Client must implement a CallBack object interface ICallBack with the following methods:

- on_event(), which is used by the server to report alarms and event messages (refer to 5.4) matching an active subscription FilterSpecification. A refresh() call by a client will also result in reporting of messages.
- on_read_complete(), which is used to report alarms and events requested by the client by calling async_read_history().

5.5.5 Mapping DAIS A&E to IEC 61970-3XX series

The GES A&E interfaces may convey data originating from an IEC 61970-301 compliant data source. For an IEC 61970-301 compliant server supporting GES, the interface shall be used as follows:

- The IArea browse interface expose a hierarchical organisation of Sources as shown in 4.2. There is no model in IEC 61970 corresponding to the Area data.
- The ISource browse interface exposes Source objects that can have alarm Conditions. Source objects can be any Type. Such object types are defined in IEC 61970-301 or extensions based on IEC 61970-301.
- The IConditionSpace, ISourceCondition, ICategory, ISubscription and ISubscription interfaces exposes data that is specific to the alarm and event processing and has no corresponding data defied in IEC 61970.
- The IType browse interface exposes the classes (meta data) defined in IEC 61970-301, for example Station, Bay, Measurement etc. The classes are exposed in a flat structure, i.e. the inheritance between classes is not exposed as a hierarchical structure. This is the ClassView as defined in IEC 61970-402.
- The Property browse interface exposes the properties (meta data) defined for a class in IEC 61970-301.

Annex A
(informative)

Proxy Event Subscription sequence

Figures A.1 to A.4 illustrate the delegation of subscriptions and callbacks through a proxy GES server.

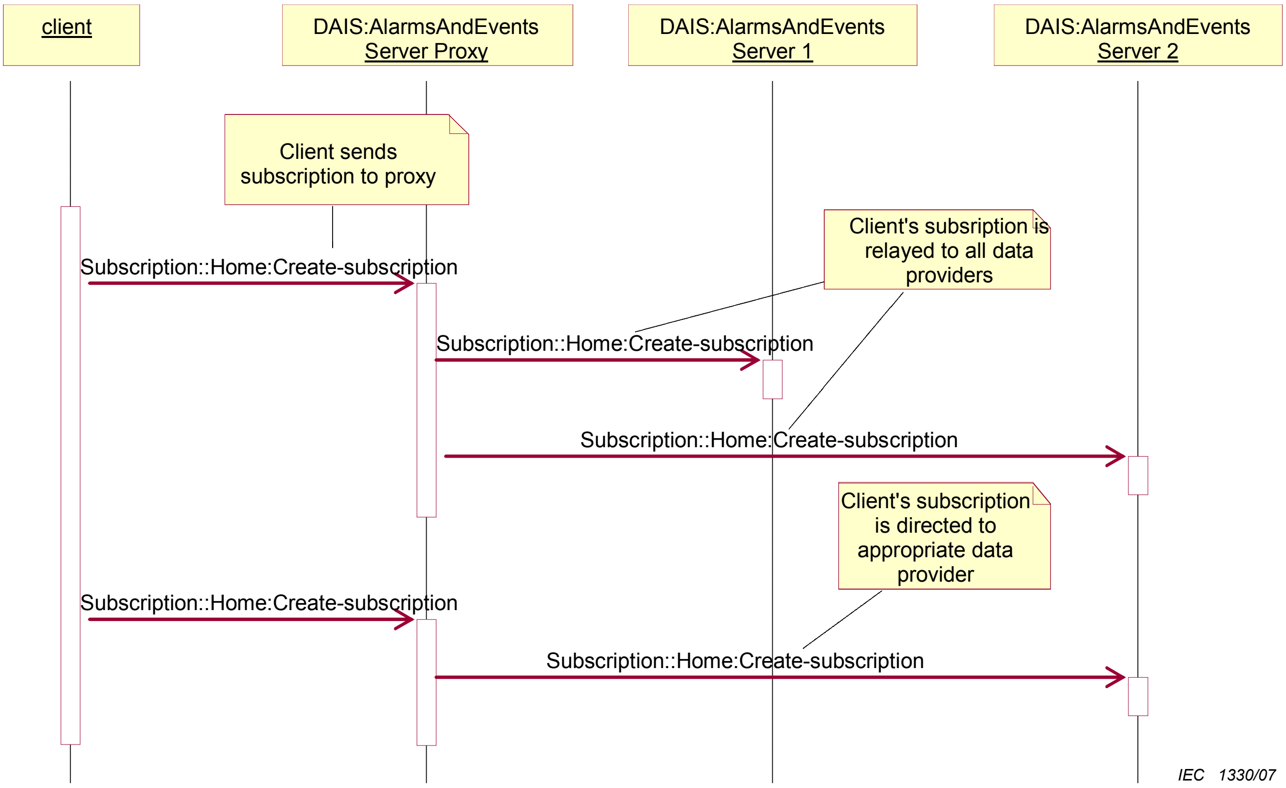

Figure A.1 – Proxy subscription and callback sequence with multiple servers

Figure A.1 illustrates two ways to delegate subscriptions through a proxy GES server to two servers. In the first case, the proxy has no knowledge of what server is providing what data. In this case, all subscription messages are simply passed on to all servers. From the clients point of view, there is only one server, thus the client is unaware of server locations. This helps decouple clients and servers, since client configuration does not need to change when a server is relocated. In the second case, the proxy has knowledge of what server is providing what data. In this case, subscriptions are only sent to the appropriate server. This more advanced configuration minimizes network traffic as well as provides foundation functionality required to construct a system with which a client to be notified when data that is subscribes to becomes unavailable.

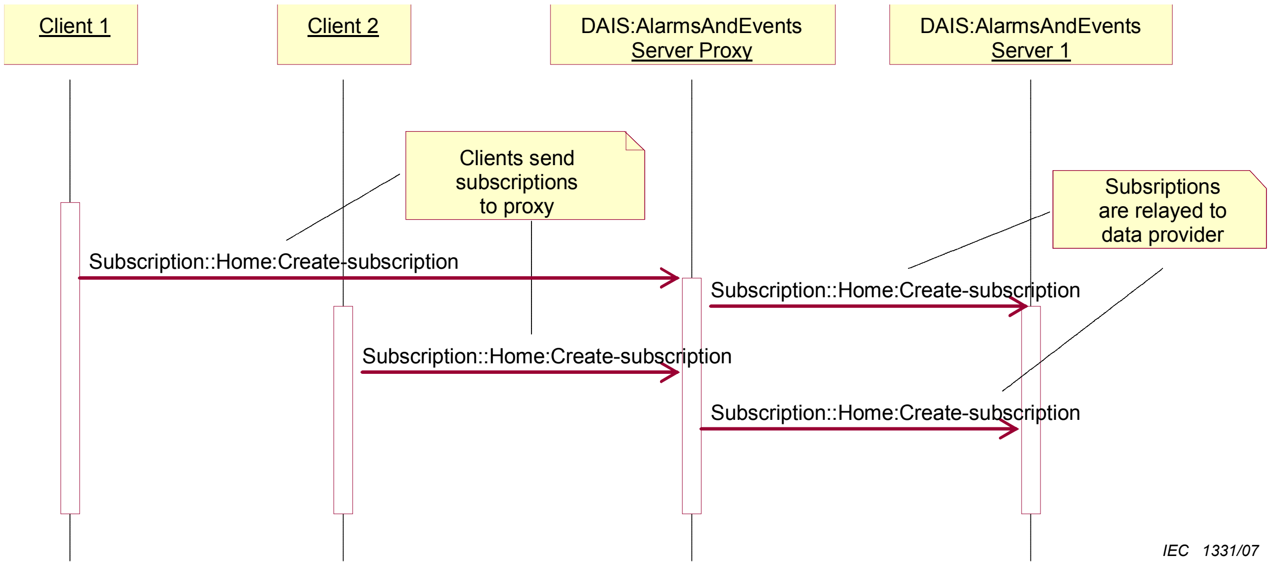

Figure A.2 – Proxy subscription sequence with multiple clients

Figure A.2 illustrates delegation of subscriptions through a proxy GES server from two clients. In this case, the server has no knowledge of what client is subscribing to what data. From the server's point of view there is only one client, thus the server is unaware of client locations. This helps decouple clients and servers, since server configuration does not keep track of multiple clients.

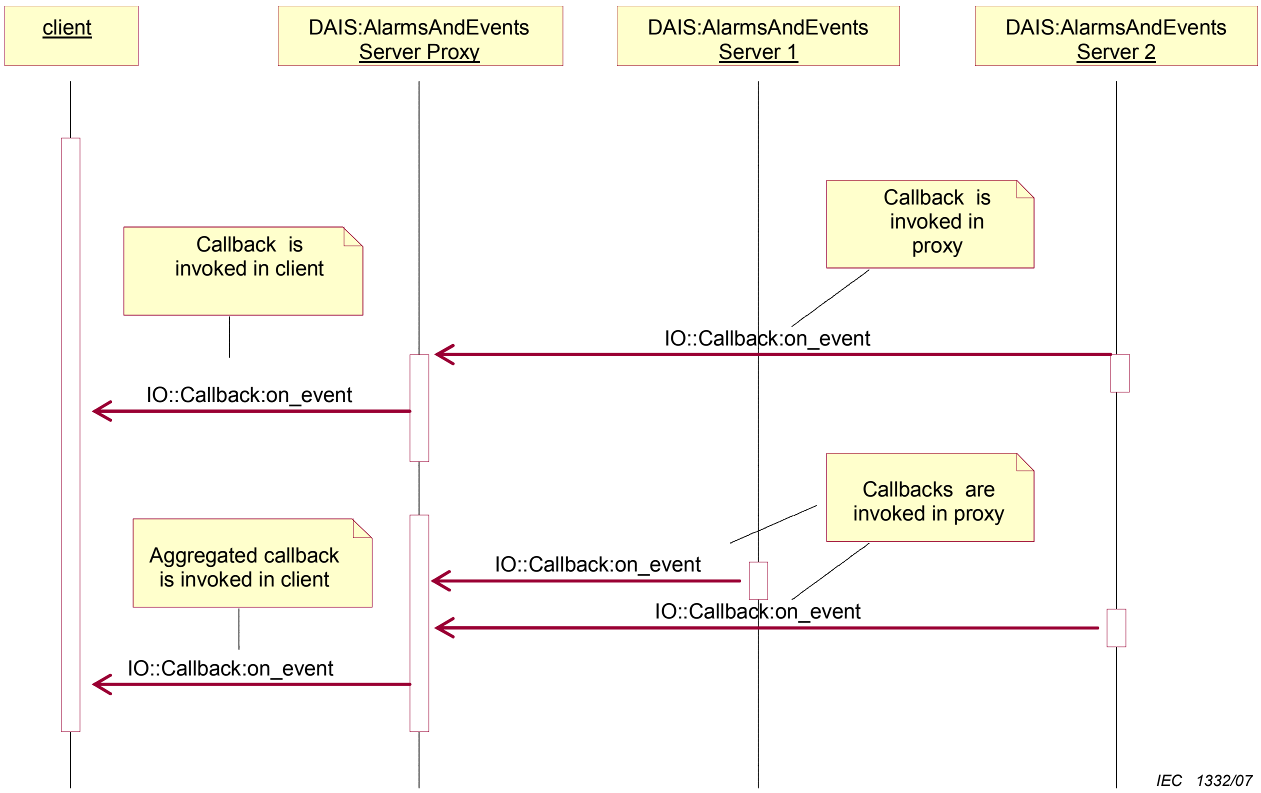

Figure A.3 – Proxy callback sequence with multiple servers

A proxy GES provider must implement the calling side of IO::Callback::on_event. Figure A.3 illustrates two ways to relay on_event messages through a proxy GES server from two servers. In the first case, the proxy simply passes on to all messages as it receives them. In the second case, the proxy aggregates on_event messages so that the client receives messages according to the update rate set during the subscription process.

In order to combine event sources, the proxy must connect to all of the ultimate GES providers and receive their on_event messages. It must then provide an event service to the clients. Each event from each GES provider is passed on to the clients.

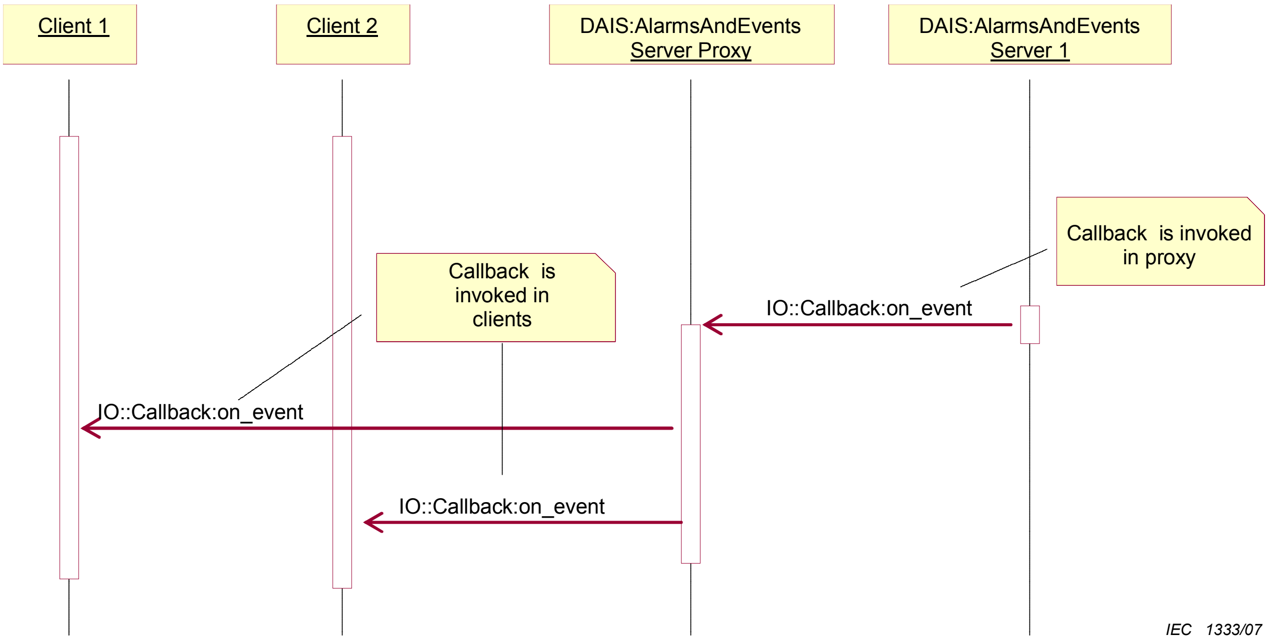

Figure A.4 – Proxy callback sequence with multiple clients

Figure A.4 illustrates delegation of on_event messages through a proxy GES server to two clients. In this case, the proxy has knowledge of what client has subscribed to what data and on_event messages are only sent to the appropriate client. This more advanced configuration can substantially decrease server loading and facilitates distribution of system processing.

Bibliography

IEC 61850-7-2, *Communication networks and systems in substations – Part 7-2: Basic communication structure for substation and feeder equipment – Abstract communication service interface (ACSI)*

IEC 61968-1, *Application integration at electric utilities – System interfaces for distribution management – Part 1: Interface architecture and general requirements*

IEC 61968-3, *Application integration at electric utilities – System interfaces for distribution management – Part 3: Interface for network operations*

Control Center Application Program Interface (CCAPI) Project: API Standard Proposal Requirements for Generic Interface Definition (GID), EPRI, Palo Alto, CA: 2001. 1001975.

OMG, *Utility Management System Data Access Facility, document formal/2001-06-01*

OPC Foundation, *OPC Data Access Custom Interface Specification*

OPC Foundation, *OPC Historical Data Access Custom Interface Specification*

OPC Foundation, *OPC XML Specification*

OPC Foundation, *OPC Batch Custom Interface Specification*

Sun Microsystems, *Java 2 Enterprise Edition, J2EE Connector Architecture Specification*

Sun Microsystems, *Java Messaging Service Specification*

Open Applications Group, *Open Applications Middleware API Specification*

Resource Description Framework (RDF) Model and Syntax Specification, W3C *Recommendation, 22* February 1999 http://www.w3.org/TR/REC-rdf-syntax, Ora Lassila, Ralph R. Swick.

Resource Description Framework (RDF) Schema Specification, *W3C Proposed Recommendation, 03* March 1999 http://www.w3.org/TR/PR-rdf-schema, Dan Brickley, R.V. Guha, Netscape

Uniform Resource Identifiers (URI): Generic Syntax; Berners-Lee, Fielding, Masinter, Internet Draft Standard August, 1998; RFC2396.

Namespaces in XML; Bray, Hollander, Layman eds., W3C Recommendation; http://www.w3.org/TR/1999/REC-xml-names-19990114

____

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
P.O. Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch