INTERNATIONAL STANDARD

IEC 61968-1

First edition
2003-10

Application integration at electric utilities – System interfaces for distribution management –

Part 1: Interface architecture and general requirements

Reference number
IEC 61968-1:2003(E)

Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site (www.iec.ch)**

- **Catalogue of IEC publications**

  The on-line catalogue on the IEC web site (www.iec.ch/searchpub) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

  This summary of recently issued publications (www.iec.ch/online_news/ justpub) is also available by email. Please contact the Customer Service Centre (see below) for further information.

- **Customer Service Centre**

  If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

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

INTERNATIONAL STANDARD

IEC 61968-1

First edition
2003-10

Application integration at electric utilities – System interfaces for distribution management –

Part 1: Interface architecture and general requirements

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE XB

For price, see current catalogue

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

APPLICATION INTEGRATION AT ELECTRIC UTILITIES – SYSTEM INTERFACES FOR DISTRIBUTION MANAGEMENT –

Part 1: Interface architecture and general requirements

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 61968-1 has been prepared by IEC technical committee 57: Power system control and associated communications.

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

FDIS	Report on voting
57/650/FDIS	57/668/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.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

IEC 61968 consists of the following parts under the general title *Application integration at electric utilities – System interfaces for distribution management*:

Part 1: Interface architecture and general requirements

Part 2: Glossary[1]

Part 3: Interface standard for network operations[1]

Part 4: Interface standard for records and asset management[1]

The committee has decided that the contents of this publication will remain unchanged until 2005. At this date, the publication will be

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

[1] Under consideration.

INTRODUCTION

The IEC 61968 series is intended to facilitate inter-application integration, as opposed to intra-application integration, of the various distributed software application systems supporting the management of utility electrical distribution networks. Intra-application integration is aimed at programs in the same application system, usually communicating with each other using middleware that is embedded in their underlying runtime environment, and tends to be optimized for close, real-time, synchronous connections and interactive request/reply or conversation communication models. IEC 61968, by contrast, is intended to support the inter-application integration of a utility enterprise that needs to connect disparate applications that are already built or new (legacy or purchased applications), each supported by dissimilar runtime environments. Therefore, IEC 61968 is relevant to loosely coupled applications with more heterogeneity in languages, operating systems, protocols and management tools. IEC 61968 is intended to support applications that need to exchange data on an event driven basis. IEC 61968 is intended to be implemented with middleware services that broker messages among applications, and will complement, but not replace utility data warehouses, database gateways, and operational stores.

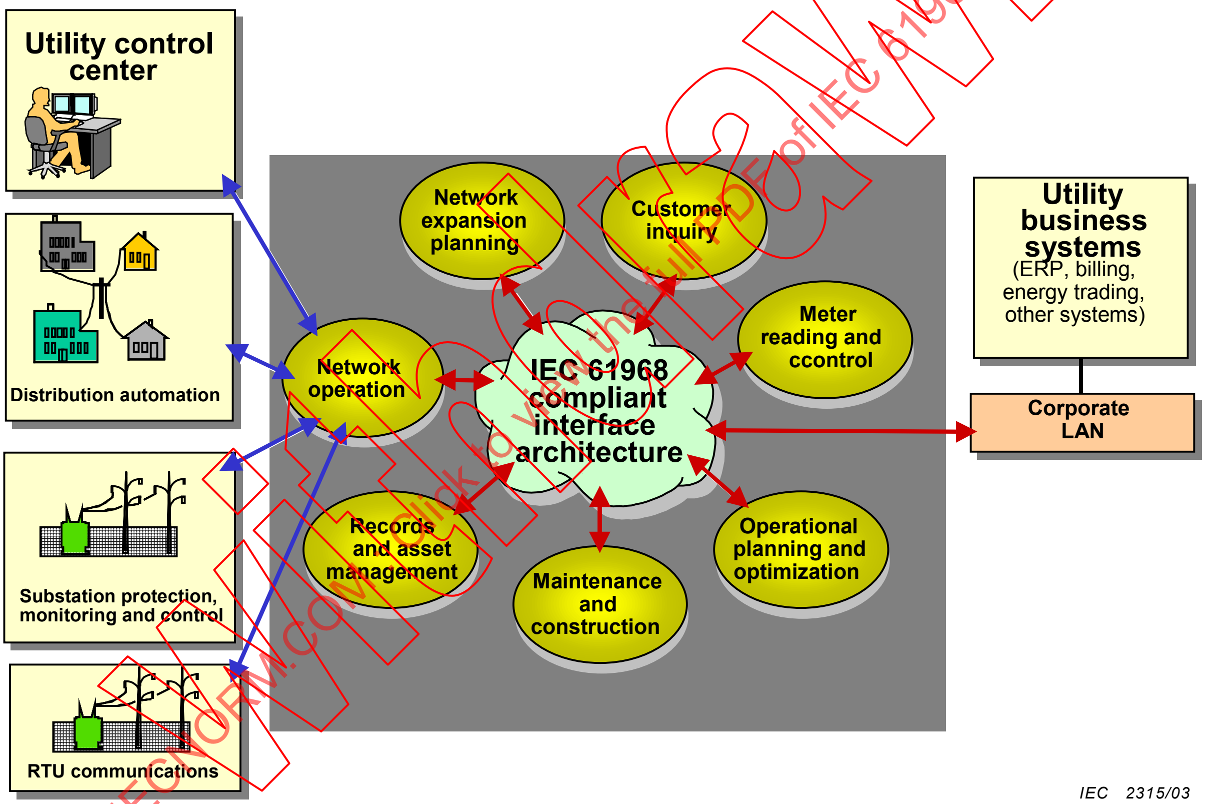


Figure 1 – Distribution management system with IEC 61968 compliant interface architecture

Figure 1 clarifies the scope of IEC 61968-1 graphically in terms of business functions and shows a Distribution Management System with IEC 61968 compliant interface architecture.

APPLICATION INTEGRATION AT ELECTRIC UTILITIES – SYSTEM INTERFACES FOR DISTRIBUTION MANAGEMENT –

Part 1: Interface architecture and general requirements

1 Scope

This part of IEC 61968 is the first in a series that, taken as a whole, defines interfaces for the major elements of an interface architecture for Distribution Management Systems (DMS). This part of IEC 61968 identifies and establishes requirements for standard interfaces based on an Interface Reference Model (IRM). Subsequent parts of this standard are based on each interface identified in the IRM. This set of standards is limited to the definition of interfaces and is implementation independent. They provide for interoperability among different computer systems, platforms, and languages. Methods and technologies used to implement functionality conforming to these interfaces are considered outside of the scope of these standards; only the interface itself is specified in the IEC 61968 series.

As used in the IEC 61968 series, a DMS consists of various distributed application components for the utility to manage electrical distribution networks. These capabilities include monitoring and control of equipment for power delivery, management processes to ensure system reliability, voltage management, demand-side management, outage management, work management, automated mapping and facilities management. The IRM is specified in Clause 4.

2 General

2.1 Overview of the IEC 61968 series

As used in IEC 61968, a DMS (Distribution Management System) consists of various distributed application components for the utility to manage electrical distribution networks. These capabilities include monitoring and control of equipment for power delivery, management processes to ensure system reliability, voltage management, demand-side management, outage management, work management, automated mapping and facilities management. Standards interfaces are to be defined for each class of applications identified in the Interface Reference Model (IRM), which is described in Clause 4.

IEC 61968 recommends that system interfaces of a compliant utility inter-application infrastructure be defined using Unified Modelling Language (UML).

The eXtensible Markup Language (XML) is a data format for structured document interchange particularly on the Internet. One of its primary uses is information exchange between different and potentially incompatible computer systems. XML is thus well-suited to the domain of system interfaces for distribution management.

Where applicable, future parts of the IEC 61968 series will define the information required for 'message payloads'. Message Payloads will be formatted using XML with the intent that these payloads can be loaded on to messages of various messaging transports, for example OAG, SOAP (Simple Object Access Protocol), etc. The XML encoding rules will be covered in a future part of the IEC 61968 series.

Communication between application components of the IRM requires compatibility on two levels:

- Message formats and protocols.
- Message contents must be mutually understood, including application-level issues of message layout and semantics.

Clause 5 defines abstract middleware services required to support communication between the applications defined in the IRM. These services are intended to be deployed, with little additional software required, by mapping them to commonly available services from various messaging technologies including middleware such as message brokers, Message Oriented Middleware (MOM), Message-Queuing Middleware (MQM), and Object Request Brokers (ORBs). This clause is organized as follows:

- Subclause 5.1 identifies general requirements of the applications identified in the IRM.
- Subclause 5.2 describes how standard information exchange services may either be invoked directly from an application (native mode) or that software may be used to map (adapt) an application to the information exchange services.
- Subclause 5.3 identifies standard services required for applications to exchange information with other applications.
- Subclause 5.4 describes how information exchange services may either be supported directly by middleware or that software may be required to map (adapt) the utility's middleware services to the standard information exchange services.
- Subclauses 5.5 to 5.7 describe environmental requirements for information exchange.

2.2 An example using the IEC 61968 series

An example of a typical utility's implementation of the IEC 61968 series is provided in Figure 2. In this example, the utility has used interface adapters as a means of integrating many of its legacy systems with other application systems that are IEC 61968 compliant. Note those legacy systems and IEC 61968 series compliant systems both continue to use proprietary integration techniques among their internal applications, only information that needs to be exchanged among applications at the utility enterprise level is expected to use IEC 61968 series middleware services.

For the purposes of this example, the utility's Outage Management System (OMS) is assumed to already have the capability of issuing controls to and gathering device states from the Distribution Automation System (DAS). As it is working acceptably for the utility, this interface does not need to be changed. However, because other applications need to be notified when distribution devices change state, the DAS publishes state changes through middleware services. Another benefit of publishing events is that they can be recorded by an event history application in a data store; this data can then be used in the generation of various types of reports. As much of the information exchanged among these systems is useful for management decision support, a data warehouse application has also been connected to the IEC 61968 middleware services so that it may receive published information.

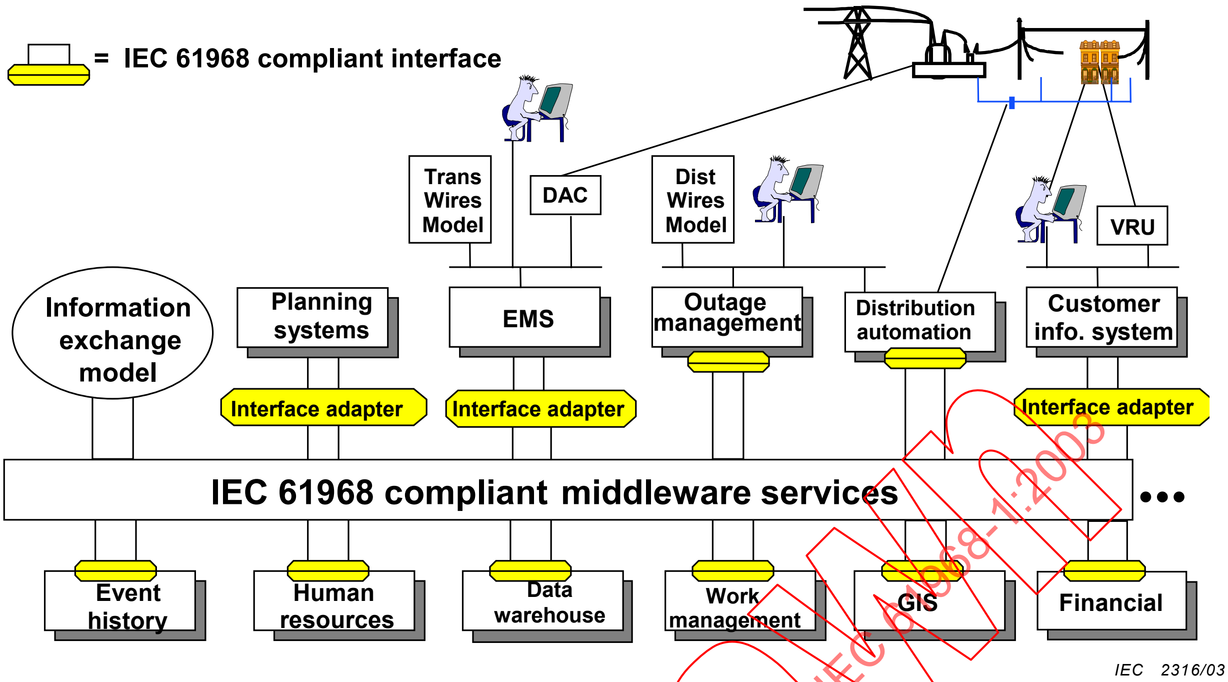


Figure 2 – Example utility implementation of the IEC 61968 series

2.3 Overview of IEC 61968-1

The organization of IEC 61968-1 is described in Table 1.

Table 1 – Document overview for IEC 61968-1

Clause	Title	Purpose
1	Scope	Scope of IEC 61968, Part 1.
2	General	Overview and examples.
3	Interface reference model	The domain relevant to the IEC 61968 series is described. For each relevant business function, a list of abstract components is provided, which is described by the functions performed by the component. future parts of the IEC 61968 series will define interfaces for these abstract components.
4	Interface architecture	The interface reference model for utility inter-application integration is provided along with the rationale for its structure.
5	Interface profile	Utility inter-application integration environmental requirements are described. Abstract message passing services are defined that must be available for applications to communicate information to other applications, including publish and subscribe services.
6	Information exchange model	Metadata is used to describe event types that are published by applications. Applications subscribing to receive all messages for a certain event type recognize the fields of a particular event message once they have looked up the metadata for the event type in the information exchange model. While many event types are described in the IEC 61968 series, metadata is the means by which vendors and utilities can add new event types without violating this standard.
7	Component reporting and error handling	Requirements for audit trails and error message handling authentication necessary to support utility inter-application integration are described.
8	Security and authentication	Requirements for security and authentication necessary to support utility inter-application integration are described.
9	Maintenance aspects	General maintenance requirements are specified.
Annex A	Distribution management domain	An overview of business functions required for electric utility distribution management is described.

Annex B	IEC 61968 series Development process	The methodology used to determine interface architecture requirements for utility inter-application integration is described.
Annex C	Inter-application integration performance considerations	Some typical performance requirements necessary to support utility inter-application integration are described. These requirements are of a general nature as specific implementation requirements will vary by utility.
Annex D	Views of data in a conventional electric utility	This annex describes some of the underlying principles of defining the reference data dictionary of a future part of the IEC 61968 series.
Annex E	Business functions	This annex describes the typical data producer and consumer subsystems for each DMS business function.

3 Interface reference model

3.1 Domain

Within this part of IEC 61968, the distribution management domain covers all aspects of management of utility electrical distribution networks. A distribution utility will have some or all of the responsibility for monitoring and control of equipment for power delivery, management processes to ensure system reliability, voltage management, demand-side management, outage management, work management, automated mapping and facilities management.

The distribution management domain may be organised as two inter-related types of business, electricity supply and electricity distribution. Electricity supply is concerned with the purchase of electrical energy from bulk producers for sale to individual consumers. Electricity distribution covers the management of the physical distribution network that connects the producers and consumers. In some countries, the responsibility of organisations may be legally restricted and certain sections of the IEC 61968 series will be inapplicable.

A utility domain includes the software systems, equipment, staff and consumers of a single utility organisation, which could be a company or a department. It is expected that within each utility domain, the systems, equipment, staff and consumers can be uniquely identified. When information is exchanged between two utility domains, then identifiers may need to be extended with the identity of the utility organisation in order to guarantee global uniqueness.

3.2 Business functions

Various departments within a utility co-operate to perform the operation and management of a power distribution network; this activity is termed distribution management. Other departments within the organisation may support the distribution management function without having direct responsibility for the distribution network. This segmentation by business function[2] is provided in the Interface Reference Model (IRM), which is described in detail in 3.3.

The use of a business-related model should ensure independence from vendor-produced system solutions. It is an important test of the viability of this standard that the IRM be recognisable to utility staff as a description of their own distribution network operation and management.

Major utility business functions, which provide the top level categories of the IRM, are shown in Figure 3 below.

2 The work of the CIRED working group on distribution automation, published in 1996, is fully acknowledged in the segmentation.

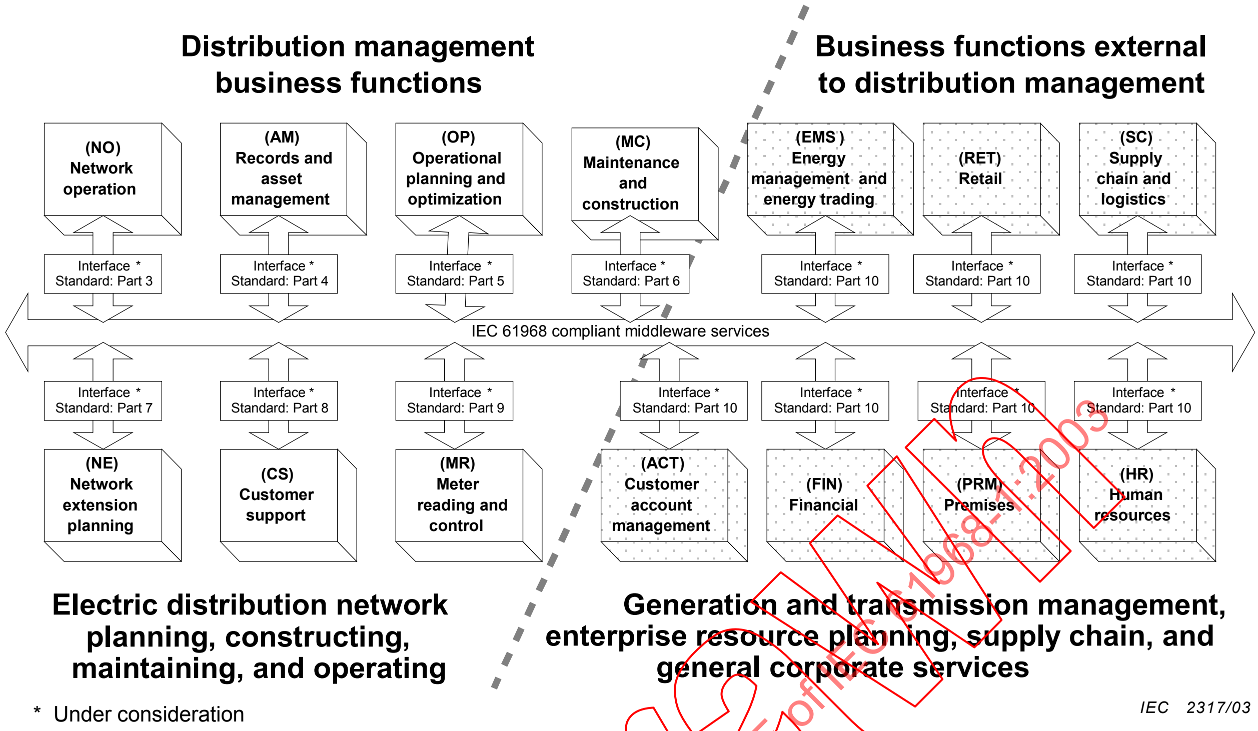


Figure 3 – Typical applications mapped to interface reference model

3.3 Interface reference model

It is not the intention of this standard to define the applications and systems that vendors should produce. It is expected that a concrete (physical) application will provide the functionality of one or more abstract (logical) components as listed in this standard. These abstract components are grouped by the business functions of the interface reference model.

In this standard, the term abstract component is used to refer to that portion of a software system that supports one or more of the interfaces defined in future parts of the IEC 61968 series. It does not necessarily mean that compliant software is delivered as separate modules.

In this subclause, the definitions of business functions defined in Subclause 3.2 are further extended into:

- Sub-business functions (second column of Table 2).
- Abstract components (third column of Table 2).

NOTE Some abstract components may be used by several different business functions. For example, a component like power flow can be used for network operation, short term operational planning and optimisation, and long term network extension planning. Much of the information exchanged for power flow purposes in each of these areas will therefore use many of the same information exchange message types (see Clause 5).

Applications from different vendors package the functionality of these abstract components in different ways. To use the IEC 61968 services, each application must support one or more of the interfaces for the abstract components.

This part of IEC 61968 describes infrastructure services common to all abstract components whilst future parts of the IEC 61968 series will define the details of the information exchanged for specific types of abstract component.

IEC 61968 series defines that:

a) An inter-application infrastructure is compliant if it supplies services defined in this part of IEC 61968 to support at least two applications with interfaces compliant to sections of future parts of the IEC 61968 series.

b) An application interface is compliant if it supports the interface standards defined in future parts of the IEC 61968 series for the relevant abstract components defined in the interface reference model.

c) An application is only required to support interface standards of the applicable components listed in column 3 of Table 2. It is not required to support interfaces required by other abstract components (column 3 of Table 2) of the same business sub-function (column 2 of Table 2) or within the same business function (column 1 of Table 2). While this standard primarily defines information exchanged among components in different business functions, it will occasionally also defines information exchanged among components within a single business function when a strong market need for this capability has been realised.

Table 2 – Interface reference model

Business functions	Business sub-functions	Abstract components
Network operation (NO) (Refer to future IEC 61968-3)	Network operation monitoring (NMON)	Substation state supervision
		Network state supervision
		Switching action supervision
		Management of data acquired from SCADA and metering systems
		Management of data acquired through operation (field crews, customers, scheduled and unscheduled outages)
		Alarm supervision
		Operator and event logs
		Weather monitoring (lightning detection)
	Network control (CTL)	User access control
		Automatic controls: Protection (fault clearance) Sectionalising Local voltage/reactive power control
		Assisted control: Remote switch control Load shedding Voltage reduction broadcast Local control through field crews
		Safety document management
		Safety checking and interlocks
		Major incident co-ordination

Business functions	Business sub-functions	Abstract components
	Fault Management (FLT)	Trouble call handling and coherency analysis (LV network)
		Protective relays analysis
		Fault location by analysis of fault detectors and/or trouble call localisation
		Supply restoration assessment
		Customer incident information
	Operation feedback analysis (OFA)	Mal-operation analysis
		Network fault analysis
		Quality index analysis
		Device operation history
		Post-disturbance review
	Operation statistics and reporting (OST)	Maintenance information
		Information for planning
		Information for management control
	Network calculations – real-time (CLC)	Load estimation
		Energy trading analysis
		Load flow/voltage profile
		Fault current analysis
		Adaptive relay settings
	Dispatcher training (TRN)	SCADA simulation
Records and asset management (AM) (Refer to future IEC 61968-4)	Substation and network inventory (EINV)	Equipment characteristics
		Connectivity model
		Substation display
		Telecontrol database
	Geographical inventory (GINV)	Network displays
		Cartographic maps
	Asset investment planning (AIP)	Maintenance strategy
		Life-cycle planning
		Reliability centred analysis
		Engineering and design standards
		Performance measurements
		Risk management
		Environmental management
		Decision support
		Budget allocation
		Maintain work triggers
		Asset maintenance groups (lists)
		Asset failure history
		Asset financial performance
		Thermal ratings of network equipment and lines

Business functions	Business sub-functions	Abstract components
Operational planning and optimisation (OP) (Refer to future IEC 61968-5)	Network operation simulation (SIM)	Load forecast
		Power flows computation
		Contingency analysis
		Short circuit analysis
		Optimal power flow
		Supply restoration assessment
		Switching simulation
		Incident simulation
		Weather forecast analysis
		Fire risk analysis
		Thermal ratings of network equipment and lines
	Switch action scheduling/operation work scheduling (SSC)	Release/clearance remote switch command scheduling
		Field crew loading analysis and work order scheduling
		Customer outage analysis and information
	Power import scheduling and optimisation (IMP)	
Maintenance and construction (MC) (Refer to future IEC 61968-6)	Maintenance and inspection (MAI)	Maintenance program management
		Maintain work triggers
		Asset maintenance groups (lists)
		Manage inspection readings
		Asset maintenance history
		Asset failure history
		Work order status tracking
		Work order closing
		Financial control
	Construction and design (CON)	Work initiation
		Work design
		Work cost estimation
		Work flow management
		Work order status tracking
		Work order closing
		Financial control
	Work scheduling (SCHD)	Work task planning
		Crew management
		Vehicle management
		Equipment management
		Material coordination
		Permit management

Business functions	Business sub-functions	Abstract components
	Field recording and design (FRD)	Field design
		Field inspection results
		Crew time entry
		Actual materials
	Work dispatch (DSP)	Field status tracking
		Real-time communication
		Weather monitoring
Network extension planning (NE) (Refer to future IEC 61968-7)	Network calculations (NCLC)	Load forecast
		Power flows
		Contingency analysis
		Short-circuit analysis
		Optimal power flow
		Energy loss calculations
		Feeder voltage profiles
	Construction supervision (CSP)	Construction costing
		Work management
	Project definition (PRJ)	Capital approval
	Compliance management (CMPL)	Safety compliance
		Technical compliance
		Regulatory compliance
Customer Support (CS) (Refer to future IEC 61968-8)	Customer service (CSRV)	Service requests
		Construction billing inquiry
		Work status
		Self service inquiry (Web, VRU (Voice Response Unit)…)
		Customer connection
		Turn on, turn off
		Service level agreements
	Trouble call management (TCM)	Outage calls
		Power quality
		Planned outage notifications
		Media communication
		Performance indices
		Restoration projection/confirmation
		Outage history
Meter reading and control (MR) (Refer to future IEC 61968-9)	Meter reading (RMR)	Load characteristics
		Consumption meters
		Quality factors
	Load control (LDC)	Meter parameter telesetting
		Dynamic tariff application
		Power modulation

Business functions	Business sub-functions	Abstract components
External to DMS (EXT) (Refer to future IEC 61968-10)	Energy management and energy trading (EMS)	Transmission
		Generation
		Energy trading
	Retail (RET)	Marketing and selling
		Settlements
		Customer registration
		Product line diversification
		Portfolio management
	Supply chain and logistics (SC)	Procurement
		Contract management
		Warehouse logistics
		Materials management
	Customer account management (ACT)	Credit status
		Outage history
		Credit and collections
		Billing and payment
		Customer profiling
	Financial (FIN)	Activity based management
		Accounts payable
		Accounts receivable
		Forecasting
		Budgeting
		General ledger
		Regulatory accounting
		Tax accounting
		Treasury
		Decision support
		Performance metrics
		Strategic planning
		Business development
		Budgeting
		Regulatory relations
	Premises (PRM)	Address
		Source substation
		Meter information
		Right of ways, easements, grants
		Real estate management

Business functions	Business sub-functions	Abstract components
	Human resources (HR)	Health/safety reporting
		Payroll
		Safety administration
		Training
		Qualification tracking
		Hours on shift information
		Benefits administration
		Employee performance, review, and compensation
		Recruiting

4 Interface architecture

4.1 General

This part of IEC 61968 describes utility inter-application infrastructure requirements necessary to integrate components distributed throughout the enterprise. The services and functionality described is independent of the underlying component-based infrastructure. In the following requirements, an “event” is a unit of information exchange which is issued asynchronously by its source (“push”). A “component” is a module of application software which is a component of the integration bus as either a publisher or subscriber (receiver) of an information exchange.

The business process begins by identifying the information to be exchanged and the components involved. This typically involves one publisher that has the information and initiates the exchange, and zero or more subscribers that will receive the information.

The IEC 61968 series requires that a compliant utility inter-application infrastructure:

a) Shall allow components to exchange information of arbitrary complexity.

b) Shall be able to be implemented using various forms of distributed component technology (for example, CORBA (Common Object Request Broker Architecture), DCOM (Distributed Component Object Model), message brokers, message oriented middleware, relational databases, object-oriented databases, or others). (See Clause 5).

c) Shall provide an information exchange model facility (see Clause 6) that users employ to describe the information to be exchanged. This facility presents the user with the models of events and the components to which they relate, and allows the new exchange to be added to the old, so that a comprehensive corporate exchange model, tailored to a utility’s specific needs, can be built rather than a collection of independent models.

d) Shall allow a publisher and/or subscriber component to be deployed by system administrators independently of other components as long as interfaces remain the same.

e) Shall ensure that, once a given type of event is published, additional subscribing components can be configured to receive the event without having to make any changes or additions in the publisher component.

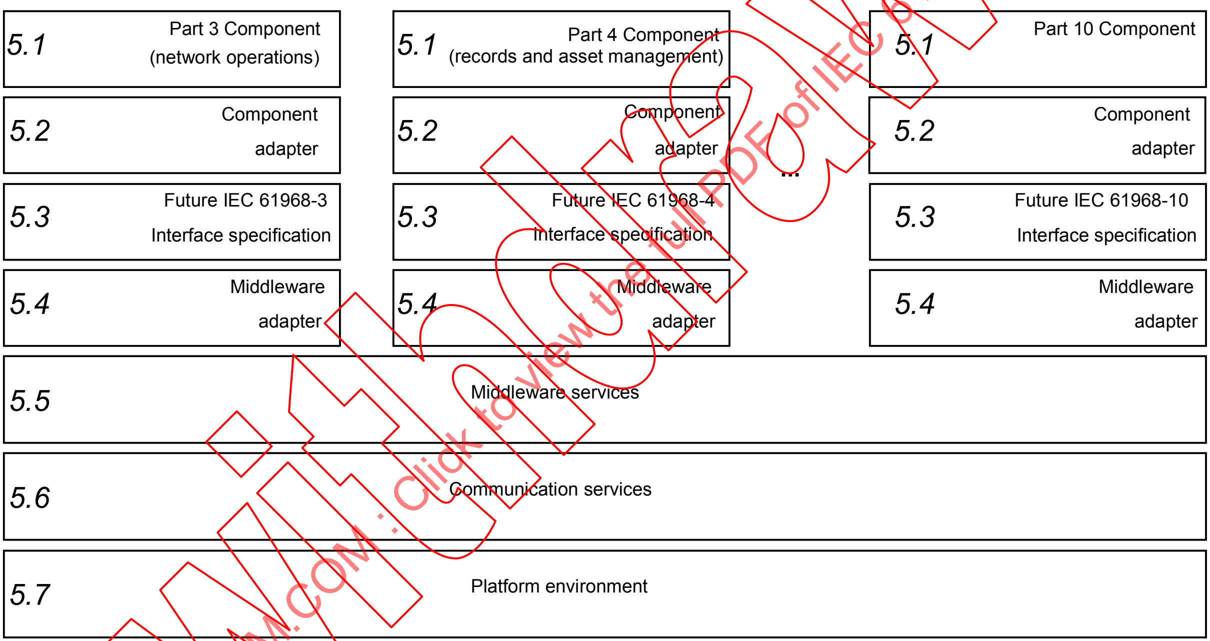
4.2 Requirements analysis methodology

To help solve the problem of effectively sharing information across electric utility departments and systems, a common modelling notation or language is needed. A modelling language extends natural language by adding formal constructs to aid in communication by reducing ambiguity. By using a common modelling language across the utility, utilities can better define what information needs to be shared across departments.

This modelling language should be rich enough to detail the requirements, have a graphically oriented (visual diagrams) to make it easy to use, be widely accepted, and supported by reasonably priced tools. Refer to Annex B for further information regarding this methodology that has been used for the development of the IEC 61968 series. The use cases used for the development of the interface reference model will be compiled in a future IEC technical report.

5 Interface profile

Clause 5 is organised according to the interface profile, given in Figure 4.



IEC 2318/03

Figure 4 – Overview of the interface profile and corresponding subclause numbers

The requirements for all the individual parts in this interface profile are explained in the following subclauses.

5.1 Components

Information exchange among components can either be a piece of data or the result of an execution of functionality[3] and for this purpose is called a services exchange. For example, a component can be a classic, procedural application (also referred to as a legacy application) or a fully object-oriented application build around the latest technology. Also, components can be distributed across the network (LAN, Intranet, private corporate WAN or even the public Internet), enabling flexible deployment of DMS applications in the utility-wide ICT-architecture. The scope of a component is unlimited: it can perform any function that is required for distribution management. Typical categories of functions are showed in the interface reference model in Clause 3.

3 Meaning that this function can be invoked remotely.

A component can either be *profile-compliant*, meaning that it knows, understands and satisfies Services requirements or *non-profile-compliant*. A non-profile-compliant component must be made compliant before it can fulfil its role on the services (see 5.2).[4]

For components, the IEC 61968 series requires that applications shall:

a) Implement at least one of the interfaces as specified in the relevant series of documents from future IEC 61968-3 onwards.
b) Implement producer and consumer services exchanges in at least one interaction type: publish/subscribe, publish/reply or conversation (or request/reply if without session context).
c) Register and unregister as a producer, a consumer or both for at least one interaction type.
d) Provide for appropriate error handling and catching and recovery for service exchanges which cannot be produced or consumed. Each interface specification in the series from future IEC 61968-3 onwards contains a single services exchange for each interaction type. This specific IEC services exchange, for this purpose only, should be produced and consumed by each component on the services
e) Be able to register in at least one specific "context" (for example real-time, test, study, version 1, version 2) at the same time such that services exchanges produced within a context will, by default, be delivered only to components within the same context.
f) Be able to override the default context for a service exchange.
g) When producing a service exchange, identify the generic type of the service exchange in order to verify consistency between the component and the information exchange model.
h) When producing an service exchange, create and package new instances of the service exchange (using the information exchange model to identify items of data that should be filled in).
i) When consuming a service exchange, identify the generic type of service exchange in order to verify consistency between the Component and the information exchange model.
j) When consuming a services exchange, parse and unpack delivered instances of the services exchange (using the information exchange model to identify items of data that should be interrogated).
k) Be able to work within transaction contexts in which one or more consumed services exchanges are either committed in full or rolled back (reverted) in full.

NOTE As an example, in CORBA, a component is equivalent to an object implementation. In DCOM, a component is equivalent to a client object or a server object.

5.2 Component adapters

A component adapter in the context of the IEC 61968 series is profile-compliant software that enables a non-compliant software application to use the services. As such, the component adapter only goes as far as necessary to make the component conformant to one or more specific interface specifications in the series from future IEC 61968-3 onwards.

[4] For example, each vendor of current DMS applications may have its own application architecture, its own API and its own mechanism of interfacing the application with other products of the same vendor. Such existing applications may very well have an important role as a client of the Services. But the industry cannot expect that a vendor rebuild all its existing applications to new versions that are profile-compliant. Even new applications may not always be profile-compliant, but instead use the established vendor-specific architecture and application interface. Therefore, non-profile-compliant components probably will be in the majority during the early stages of the IEC 61968 series. When the IEC 61986 series becomes more widely accepted, profile-compliant components will become more widely available.

NOTE 1 This implies that:

- For components that already are profile-compliant, the component adapter is not necessary.
- When a non-compliant component is used in the services-environment, at least one component adapter is present for that component to make it profile-compliant. It can also be the case that more than one component adapter is used to make a single component compliant with the services (for example one component adapter for each IEC 61968 series interface specification).
- For those components that are non-compliant, each component adapter is custom-made for that specific component because it depends heavily on the architecture and implementation of the component. A component also runs in a specific hardware/operating system (HW/OS) environment. Therefore the triple set component, (set of) component adapter(s) and HW/OS are fully dependent on each other.

How the component adapter makes a non-profile-compliant component compliant to the services, depends on the component and the role it performs. A complication is that a component that was not coded to be profile-compliant cannot be made profile-compliant directly, and that each component is different.

NOTE 2 Examples of how the component adapter accesses non-profile-compliant components to make them profile-compliant are:

- accessing the component via its own specific application program interface;
- accessing the component via its data stores (flat files, databases, whatever);
- accessing the component via screen scraping (emulating a terminal and accessing fixed positions on the emulated screen);
- accessing the component via batch control or another external application run control method.

For component adapters, the IEC 61968 series requires that they shall meet the requirements specified in Clause 5.1 for a profile-compliant component.

NOTE 3 A component adapter does not exist for a profile-compliant component.

5.3 Interface specification

The IEC 61968 series interface specification requirements consists of three parts: component-specific specifications, requirements that refer to services specific for the distribution management domain and requirements that refer to services which are common in a distributed computing environment based on components. Individual IEC 61968 series interface specifications for functional areas (see the interface reference model in Clause 3) are available in following parts of the IEC 61968 series (future IEC 61968-3 and further).

For all three parts in an IEC 61968 series interface specification, it shall:

a) be declarative, containing pre- and postconditions, attributes, methods and parameters as needed for all the service exchanges that are part of the specific interface specification;

b) be programming-language neutral;

c) emphasize the separation of interface and implementation;

d) be middleware-independent.

Requirements for component-specific interface specifications have exclusive usage of IEC 61968 series interface services for those requirements that can be supported by those services.

NOTE This means that for requirements not covered, additional services may be specified (and probably need to be programmed or mapped in the middleware adapter or middleware services).

Required services for the distribution management domain shall be as follows:

e) Identifier creation and aliasing service. This is a set of services for creating and maintaining unique identifiers for business objects for which information is transferred between components. There may be multiple types of identifiers based on specific rules for each type of business object. It is expected that within each utility domain (i.e. company or department), the systems, equipment, staff and consumers can be uniquely identified. When information is exchanged between two utility domains, then identifiers may need extending with the identity of the utility organisation in order to guarantee global uniqueness.

f) Persistent exchange service with checkpoint facilities, allowing components that register at different times to synchronize status with other components. This service supports entering and purging exchanges, marking (a group of) exchanges and reading (a group of) historical exchanges.

g) System administration: this service interface allows administration and monitoring of exchanges and components on the services. Component failure and load balancing are also part of this service.

h) Configuration: this service provides an interface for components to obtain their configuration from a persistent data store at start up or while running after a component has been partially configured locally.

i) Filtering: this service allows for the definition and applying of filters based on exchange types and contents.

Required common distributed computing services in distribution management shall be:

j) Component life cycle service: these services allow the starting, stopping, and control of components to be executed on the services.

k) Naming service: this service provides a component naming service that supports a hierarchical structure and allows a component to locate other components using a human readable name. The naming service supports use of existing utility names, as well as creation, removal and aliasing of names.

l) Time service: this service provides a way for distributed components to all have the same time with a configurable accuracy.

m) Concurrency control service: this service facilitates management of shared, similar items that are distributed on the services, for example when multiple components take care of different parts (exchanges, exchange types) of the same business object in real life.

n) Security services: these services allow an application to set and verify the privilege level of components and users with which exchanges are being performed, as well as encryption and decryption of individual exchanges. This service also supplies host authentication i.e. Authentication of node(s) that attach to the services.

o) Transactional service: this service allows an application to declare the beginning and end of a multi-step transaction that either succeeds or fails as an atomic unit.

p) Component interaction services. These services allow for reliable message transfer with a selectable quality of service. The component interaction services allow for life-cycle management of interaction services (create, delete, copy and move) and querying of established interaction (mainly valid for publish and subscribe interactions).

q) There is the publish and subscribe messaging service, which allows for synchronous and asynchronous message transfer between de-coupled (anonymous) component instances.

r) There is the request/reply messaging service, which allows for reliable synchronous message transfer between coupled, identified component instances.

s) There is the publish/reply messaging service, which allows for a de-coupled initiation of a message transfer (publish), which is then finished by a coupled transfer (reply).

5.4 Middleware adapter

A middleware adapter in IEC 61968 is profile-compliant software that augments existing middleware services so that the utility's inter-application infrastructure supports required services. As such, the middleware adapter only goes as far as necessary to make the used set of middleware services conformant to the requirements of one or more of the interface specifications in the series from future IEC 61986-3 onwards. In this context, the middleware services represent not one single interface, but represents a set of interfaces to a set of corresponding services for components.

For example, each vendor's component may use any middleware internally (or no middleware at all) that is appropriate for the needs of the specific business function. Thus it cannot be assumed that two arbitrary components will always use the same implementation of middleware services that are used by the utility. Thus a middleware adapter is needed that is able to act as a middleware "gateway" for IEC 61968 series exchanges produced by one component over the implemented middleware services into the upper layers of the other component(s) (which may be based on other middleware).

The future parts of the IEC 61968 series (from future IEC 61968-3 onwards) define the required services (see previous Subclause) that must be present in the total architecture implementation that components can depend on. However, different middleware services implementations will provide different levels of services and different operating environments may provide some properties implicitly and require others to be added by the middleware adapter. If the middleware services implementation does not provide a feature, the middleware adapter can provide it. It is possible for an object implementation to have access to a service whether or not it is implemented in the ORB core. If it does not, the middleware adapter must implement it on top of the implemented middleware services.

NOTE This implies that:

- For a middleware service implementation that provides the service, the middleware adapter is required to provide a mapping to it.
- When a non-compliant middleware services implementation is used in an IEC 61968 series environment, at least one middleware adapter is present for that middleware services implementation to make it IEC 61968 series compliant. It can also be the case that more than one middleware adapter is used to make a single middleware services implementation compliant with the services (for example one middleware adapter for each required IEC 61968 series interface service).
- For those middleware services that are non-compliant, each middleware adapter is custom-made for that specific middleware services implementation because it depends heavily on the architecture and implementation of the middleware services implementation. It also runs in a specific, possibly distributed hardware/operating system (HW/OS) environment. Therefore the triple set middleware services implementation, (set of) middleware adapter(s) and HW/OS are fully dependent on each other.
- The middleware adapter (in theory) is reusable for multiple IEC 61968 series interface services running over the same middleware services implementation in the same computing environment.

The IEC 61968 series requires that middleware adapters shall provide the full set of service requirements specified in the specific IEC 61968 series interface specifications. They may do that via simple mapping of services to middleware services implementation, or via additional software components dedicated to provide one or more IEC 61968 series interface services

5.5 Middleware services

Information exchanged among components can be performed within the same process, across processes on the same machine (local) and across machines (remote). Object request brokers usually support different communication patterns, for example synchronous and asynchronous interaction. Subscription refers to the ability to read or modify objects at cyclic or event driven times. Messaging covers more the features of current messaging middleware, such as store-and-forward, persistence of messages and guaranteed delivery.

The middleware services shall provide a set of APIs so that the previous layers in the interface profile among others can:

a) locate transparently across the network, and interact with other applications or services;
b) are independent from communication profile services;
c) be reliable and available;
d) scale up in capacity without losing functionality;
e) provide the ability to support business-to-business (B2B) transactions where needed.

As an example, in CORBA the basic object adapter supplies some of the basic middleware services for life cycle and registration.

5.6 Communication services

Integrating two components requires a connection between them. As there is more than one kind of network, different resources use different protocols, such as IIOP and HTTP. To connect multiple components, an integration system must reconcile network and protocol differences transparently to the components.

IEC 61968 requires that the communication service:

a) shall guarantee delivery of network messages to their network destination if that is active;

b) shall provide guaranteed delivery, ensuring that network messages are delivered exactly once, regardless of network failures or changes;

c) shall provide guaranteed ordering, preserving the sending sequence of the source when delivering messages, regardless of network failures or changes;

d) shall guarantee that if a network message cannot be delivered to a network destination, the network source will receive a message indicating the non-delivery;

e) shall provide a selectable quality of service for prioritization of network messages or delivery via specific network paths;

f) shall provide dynamic adaptation to the speed of processing network messages by the network destination to allow slow destinations to work on the services.

5.7 Platform environment

Services are based on hardware and software standard platforms. Different hardware and operating system platforms from different vendors have to be dealt with. This means that it cannot be expected that a component running in a dedicated hardware environment (processor, operating system, language and compilers) be able to also run on another hardware environment without modifications.

The hardware environment (processor, I/O, operating system, GUI (Geographical User Interface), compilers and tools) of the IEC 61968 series requires that:

a) it shall support multiple local processes running concurrently and it does not matter if this is achieved on a single processor or multi-processor hardware;

b) it shall support inter-process communication between concurrent processes;

c) all other specifics of the hardware environment shall be shielded by the other layers in the interface profile.

6 Information exchange model

6.1 General requirements

This document defines requirements of an Interface Reference Model (IRM) for distribution management where components distributed over the communication network exchange information using IEC 61698 series services. Only functionality and services required to support information exchange are enumerated in this clause; the manner in which this functionality is implemented is beyond the scope of this standard.

The IEC 61968 series requires the following from a compliant utility inter-application infrastructure:

a) It shall have one logical IEC 61968 series Information Exchange Model (IEM) and its implementation may be physically distributed. This facility allows information exchanged among components to be declared in a publicly accessible manner.

NOTE Other non-IEC 61968 series information exchange models may exist within a single utility. For example, information exchanged for a general business application may be separately designed and maintained from the IEM for distribution management. The IEC 61968 series requirements do not affect these IEMs.

b) The IEM shall maintain descriptions of the contents, syntax and semantics (i.e. meaning) of the information exchanged between components. Such descriptions are commonly referred to as metadata (or a data dictionary).

c) The IEM shall be accessible in machine-readable and platform independent form.

d) Information is exchanged between components via one or more events whose types are defined in the IEM.

e) The IEM shall be capable of containing:

- Names of primitive data types and their mapping to standard data types such as float, integer.
- Named business object types such as breaker, outage schedule and network diagram.
- Name and data type of the attributes of the business objects such as ‘inService’, ‘voltage’.
- A name of relationships between business objects such as ‘owns’, ‘connectedTo’.
- Named event types which act on objects for example object attribute update, object creation, object deletion.

f) The IEM may be capable of containing named datasets (i.e. sets of business object types, object attributes, event types or object instances).

g) The IEM shall support services such that (note that the registration services are specified in subclause 5.1, requirement c)):

- A component can register its name and what event types it may publish.
- A component can register its name and what event types it may subscribe to.
- A component can register the context (real-time, study, test) for which it publishes/ subscribes.

6.2 IEM management related services

The IEC 61968 series requires that a compliant utility inter-application infrastructure shall provide the following IEM life cycle services:

a) IEM data definition and maintenance. This service shall allow dynamic changes in the information model governing exchanges. If the existing model is amended and component interfaces have not been modified, no re-coding or recompilation of components shall be required. New versions of components can use the specific information which has changed or was added without affecting the operation of the remaining components.

b) Validation of the information exchange model, for example enforcing uniqueness of names, version control.

c) Synchronisation of components to use the same version of the IEM when it is updated.

d) The IEM management system shall include a method of generating human readable reports.

e) IEM data discovery. This facility shall make the IEM available to the components in machine-readable forms.

f) The IEM management system may provide facilities for the dynamic creation, modification and deletion by components of named data sets.

7 Component reporting and error handling

7.1 General

The IEC 61968 series requires the following from a compliant utility inter-application infrastructure:

a) It shall provide a generic event history facility as a component. This allows all or selected information exchanges to be saved in a permanent store.

b) The event history's schema shall be based on the metadata provided by the information exchange model (refer to Clause 6).

c) The event history component shall record the time at which the publishing component issued each event.

d) It shall be capable of supporting event information model versions and component versions. (This allows a complete audit trail to be preserved which is capable of supporting rigorous reconstruction of history, if that should become a requirement.)

e) It shall provide an inter-application supervisor component that analyses the state of any application component interface connected to the utility services. It may be enabled and disabled, and is capable of providing performance monitoring capabilities. Those elements will help to provide statistics in order to identify bottlenecks or areas subject to improvement in the future. The information is required to help the administrators configure information exchanged among components and to ensure availability.

f) A component shall be able to send or request information without knowing where the receiving component is physically located or if it is currently connected. The receiver may be unreachable because of a network problem, or be naturally disconnected as in the case of mobile users who only connect periodically.

NOTE 1 Components may be unavailable because they have failed or because they only run during certain hours; when the network becomes available or the receiving application is ready to process requests, the waiting information must be delivered.

NOTE 2 A journalizing service may be available and is used for visualisation of computer and communication related (i.e., non-power system) events occurring on the system. The journalising service should be implemented as a specialisation of IEC 61968 persistent exchange service.

7.2 Error message handling

As a general rule, upper layers of architecture contain operations at higher levels of abstraction. At these levels, less detailed information is sufficient because less detail is present concerning the operation that failed. The principle is that error information should match the level of abstraction of the layer in which it is being examined.

The information contained in the error report shall contain sufficient detail to be useful in coping with the error.

NOTE 1 There are different types of errors: warnings, non-fatal errors, and fatal errors.

- Warnings: information messages; for example, message queue buffer is nearly full.
- Non-fatal errors: recoverable erroneous condition that does not require re-initialisation; for example, data integrity failure.
- Fatal errors: erroneous condition that requires re-initialisation of one or more components and/or services. Unaffected components and services continue to operate in a restrictive configuration until recovery is complete.

NOTE 2 An exception specification is part of a function signature; for example, in C++ it consists of the keyword throw, written after the parameter list and is followed by a parenthesised list of types. The exception specifications are not a statement of what should happen, but a statement of what might happen and a guarantee of what will not happen.

8 Security and authentication

8.1 General

Security concerns arise at any exposed (via communication or other) interfaces within a system. A secure system enforces at a minimum, authentication at all such exposed interfaces. As a consequence of both deregulation and the growth and utilisation of the web, it will be necessary to ensure that appropriate security measures are taken. For these reasons, the standards will be drawn from both IEC and non-IEC sources.

A user, either a human being or component, interacts with a component. The interface between the user and the component represents an exposed component interface through which major security breaches could occur within the system. For human users, it is the responsibility of the requesting component to authenticate that the user has the authority to:

- Use the business function.
- Use the Services on an individual service basis. Although such a restriction will aid in security, the rights to issue service requests of a remote component shall be enforced by the requested remote component service.[5]

Once the user has been authenticated, it is the responsibility of the component to perform a determination of the user's authentication versus the security parameter values required by the remote component, which the user is attempting to access.

Requirement

a) An IEC 61968 compliant system shall implement security requirements as determined by the utilities security policy.

This implies that a utility will have conducted an appraisal of security requirements for all application components that form an IEC 61968 series compliant system as a precursor to forming a utility policy. This would normally be achieved by analysing security threats and their consequences on the business of the utility.

b) The security model must support several choices of implementation and the utility must be free to pick the security model most effective for its needs.

c) An IEC 61968 series compliant system shall support the security requirements of compliant application components (see Subclause 5.3 requirement p)). The security shall work in co-operation with application security and should not have to replace it.

NOTE A feature of IEC 61968 series compliant systems is that an application is able to access the data 'owned' by another application. Both parties have a responsibility to define minimum security levels for use of their data. This may vary according to where the data is to be transported, i.e. inside or outside a firewall.

8.2 Security threats

Security threats include:

- **Authorisation violation** – an authorised peer attempts to perform actions/functions for which the peer is not authorised. The appropriate security mechanism to counter this threat is the use of peer authentication coupled with application level access-control.
- **Eavesdropping** – the communication packets are being monitored by a system intruder. This threat impacts the confidentiality of sensitive information. The appropriate security mechanism to counter this threat is the encryption of sensitive information.

5 The requesting component service restriction is optional and does not increase the robustness of the overall security integrity of the system. However, such restrictions may be useful as a migration path towards system security for systems where the remote applications do not support the security services as specified in this document.

- **Information leakage** – disclosure of information to an unauthorised entity. This threat impacts the confidentiality of sensitive information. However, the security outlined in this document does not counter the use of network traffic as a means of conveying information. The appropriate security mechanism to counter this threat is the use of peer authentication coupled with application level access-control.
- **Intercept/alter –** the communication packets are intercepted by an intruder. The information in the packets is then modified and forwarded to the original destination application. This threat poses data integrity issues. The appropriate security mechanism, to counter this threat, is encryption.
- **Masquerade** – this threat is typically referred to as spoofing. An intruder attempts to gain system access by attempting to pretend to be a different entity. This threat poses a severe control and data confidentiality risk. The appropriate security mechanism, to counter this threat, is the use of strong-authentication.
- **Replay** – a communication packet that has been obtained through eavesdropping is retransmitted onto the network at a later time. If the captured packet contains control commands, this threat can have severe consequences. This threat can be countered through appropriate encryption coupled with dynamic encryption key management. The key management mechanisms are a local issue.

8.3 Security functions

Requirement

a) The agreed utility policy for application components forming an IEC 61968 series compliant system shall determine which security and authentication[6] features are required. These features should include the following as appropriate:

- Restriction of access to unauthorised users.
- Automated journalising of all communication system changes.
- Automated management of user authorisations.
- Automated management of communication address[7] allocations.
- Record level data locking facilities.
- Support for encryption of names and data values.
- Automated virus and worm detection and elimination facilities.
- Protection against threats which may deny service to authorised users[8].

The above features will ensure data integrity and adequate immunity of the communication interface to unauthorised access to data and control functions. The middleware mechanisms take responsibility for security and encryption. This includes guaranteed delivery, identification, authentication, access control, and encryption, where required.

Encryption is provided either by the message transport or by added value message handlers invoked during call processing.

---

6 Communication security includes all measures and controls that ensure confidentiality, integrity, and availability of the information processed and stored by an application. Confidentiality provides assurance that information is not disclosed to unauthorised persons, entities or processes.

7 Addressing provides the communication means to identify the source and destination (recipients) of all information transfers.

8 Denial of service refers to any action that disables normal operation of any portion of the communication system. A user with the capability to modify information or exhaust system resources could interfere with the legitimate use of the system. For example, service denial may result from a node or application, accidentally or intentionally, overloading a communication network or interfering with application processing in such a way that a legitimate user is blocked from sending information for some significant time period.

b) Due attention shall be given to the level of security and features which may have been applied elsewhere to certain data, i.e. an IEC 61968 series compliant system shall not be the "weak link in a chain".

8.4 Management of integrity and security

An IEC 61968 series compliant system shall provide both integrity- and security-oriented services.

8.4.1 Levels of immunity

Integrity is the immunity of the communication network to data transfer errors resulting from accidental or intentional interference. Three distinct levels of error immunity shall be required.

- **High:** transfer of data with negligible probability of undetected error; for example, control commands and system critical parameters.
- **Medium:** transfer of inherently redundant information; for example, power system measurements and plain text.
- **Low:** transfer of routinely updated information where occasional errors are merely a nuisance; for example, voice traffic.

8.4.2 Levels of security

Three levels are defined for immunity of communication resources to accidental or intentional unauthorised access. Three levels of security shall be required:

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

The highest level of security may include provision for proof of data origin to a receiver and proof of data delivery to a transmitter.

8.5 Security agent

The security agent is a service that is responsible for the enforcement of authentication, encryption, access control, maintenance of security configuration, and the maintenance of security management parameters. In general, there is a single instance of the security agent within a server. However:

a) there shall be no behavioural differences between a single instance and multiple instance implementation, as the agent(s) shall behave in accordance with association specific attributes.

b) The security agent shall authenticate the establishment of an association through a local mechanism. However, once authenticated, the agent for the association shall enforce the appropriate access privileges.

   Authentication is performed within a component. Security procedures are enforced based upon three conditions supplied by the communication profiles over which the component executes:

   1) The level of security to be applied.
   2) A value that represents a logical security parameter, which is to be supplied to the appropriate security functions.
   3) A value that represents the originating address of the request that is being presented to the application.

c) The Security Agent shall not assign severity codes based upon the security violations that have been attempted. The assignment of such codes is the responsibility of the component that is receiving the security violation reports.

9 Maintenance aspects

Maintenance is an important part of the life cycle, which comes at the end of a long process (design, implementation, exploitation of a system). The level of maintenance problems will reflect the quality of the design and implementation of the integrated components, each of which is produced by different sources. Reduced reliability, increased executable size, and reduced performance are among the likely consequences of a poor implementation. Reduced testability, reduced usability and reduced modifiability are important primary causes. Secondary causes include increased link time, reduced comprehension and increased compile-time.

The IEC 61968 series specification of component interfaces does not place requirements about how each component should be designed internally. However, design is encouraged to be modular and de-coupled from other component designs. That is, components should be largely self contained and have little interdependence.

The IEC 61968 series requires that a compliant utility inter-application infrastructure shall:

a) allow a subscriber component to be deployed by system administrators independently of the publisher or other subscribers, and there may be any number of other subscribers. Once it is deployed, the system is aware of the subscriber's registration, and it will deliver any exchanges that fit the registration.
b) ensure that the system handles all component location transparently so that those components can be relocated in any host without changing the component code.
c) provide initialisation facilities which can synchronize a component start-up in two ways:
   - Deliver the last values of all registrations.
   - Deliver all instances of exchanges which would have been delivered since the component was last active (provided the instances are in the history).
d) provide a component registration that shall be able to be "active" whether or not the component itself is running at the moment. (Thus the system shall be able to hold messages for subscribers who come on line periodically or sporadically, and also for continuous components which can fail and return.)
e) allow components to fail and return without any requirement to re-initialize other components. Components will use services for this capability. This warm-starting facility can assume that all events which the component would have received while it was down will be available. The component will thus be able to continue to operate in an event driven mode once it has updated its local data store from a previous known (i.e., marked) state.
f) provide notice to all registered components that they must "cold-start" if the utility's IEC 61968 series system is down longer than its survival duration, or if there is a failure of the Services, or if for any other reason, the flow of event information cannot be trusted. This requires that components re-initialise themselves without assuming that they have received every event for which they have registered.
g) ensure that a publisher component shall be able to issue a cold-start related to any event type for which it has not been able to guarantee a continuous stream of events.
h) make provision to interface to, as needed, utility standard network management services.
i) support configuration management of the utility's IEC 61968 series system. Provision shall be made for the administration and deployment of different versions of the same components within the same utility system.

Annex A
(informative)

Distribution management domain

In describing integrated systems, it is important to keep in mind the basic meaning of the following words, and how they are presently used:

- Management: effective regulation and direction.
- Automation: working without human participation.
- System: a set of organized operations working to support a particular activity (set of applications). Generally, a system is in the context of this work is a computer based technology.

In the world of integrated systems, systems are also a subset of a system. A system which is built up of many subsystems uses the subsystems to support particular activities better than if the subsystems are operating independently.

Consider the hierarchy of possibilities involving data exchange with some of the examples of their implementation. In Figure A.1, the basic building blocks, which have a particular functionality, are the programs or packages or applications. A suite of applications combines together to a system. Several systems may be required to provide the support for a department with a specified responsibility. Thus, data is being exchanged between applications, between systems, and between departments. Finally, each company has commitments for information exchange with other companies with which it is dealing.

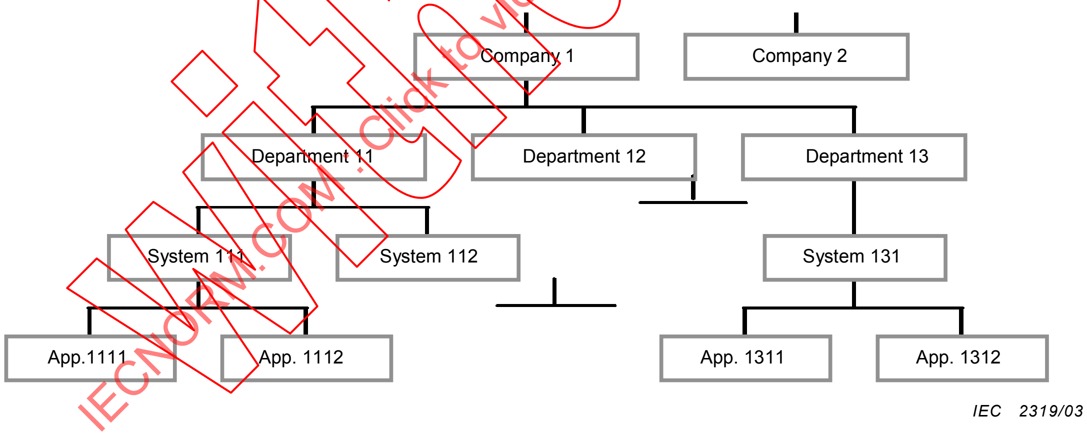
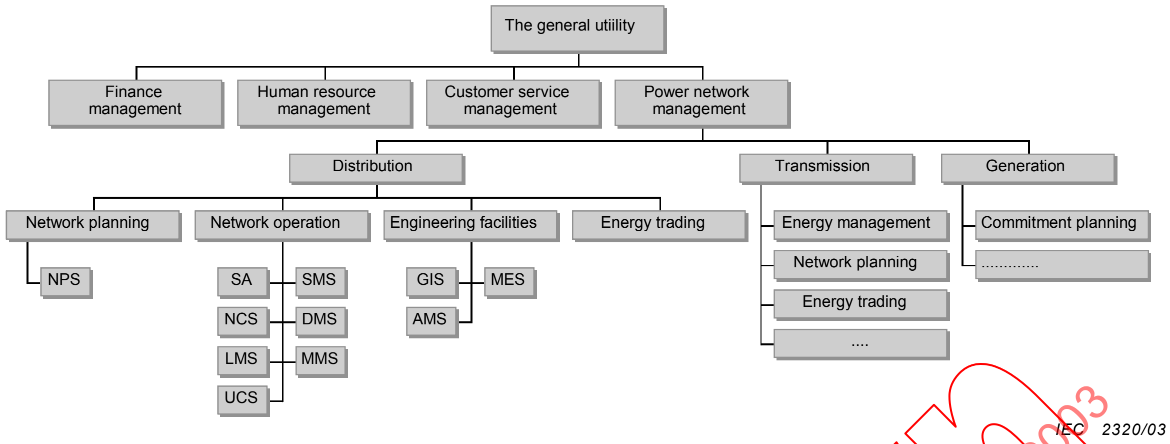


Figure A.1 – Hierarchy of complexity in a system environment

As integration and data exchange become automated, the systems merge to become themselves subsystems of the next higher system. Thus, in Figure A.1, System111 and System112 become subsystems of the Department11 system. At the next level of integration with automated data exchange, the Department11 system and the Department12 system, etc. become themselves subsystems of the Company1 system.

The fact that departments have names, and that some departments or specific responsibilities have become automated earlier than others has led to particular naming conventions being applied to describe the support system. We will continue to use the general names as usually accepted, pointing out the content and limits of the system to provide consistent interface definition. Within the corporate utility environment, a typical structure which helps to classify the key system interfaces follows from Figure A.2.



Key

NPS:	Network Planning System	SMS:	SCADA Management System
SA:	Security Analysis	MMS:	Materials Management System
NCS:	Network Control System	AMS:	Asset Management System
LMS:	Load Management System	MES:	Materials and Engineering Standards
UCS:	Utility Communication System		

Figure A.2 – General utility structure

The departments involved in finance management and human resource management are the internal corporate services departments whereas the customer service management and power network management departments are the utility business departments. Note also that a utility for the purposes of this working group has a distribution power network, but need not necessarily have transmission and/or generation.

Table A.1 – Examples of data exchange in a company environment

Information exchange	Example of information
Company – Company	
Utility – Supplier	Equipment order
Utility – Utility	Energy exchange statistics
Utility – Component	Tariffs and billing
Department – Department	
Network planning department – Operations department.	Network extensions
Trading department – Operations department	Load forecasts
Operations department – Engineering department	Switching statistics
System – System	
Substation control system – DMS	Relay status
Power station control system – EMS	Unit efficiency statistics
Program – Program	
State estimation – Optimal power flow	Network data
Unit commitment – Economic dispatch	Cost curves

The structure of data being exchanged tends to increase with the complexity of the tasks involved on either side of the exchange, as depicted in Table A.1. Furthermore, the deeper the data structure is within the system, for example data exchange between two applications, the less transparent it is to the end user.

The type of data that is regularly used by the utility (see Table A.2) and the spectrum of users and suppliers of data implies that the basic data must be “owned” by one department, to avoid:

- errors arising from multiple points of data entry;
- lack of consistency with software interfaces;
- expensive changes with new or upgraded software;
- loss of overview of authorised data.

Table A.2 – Data categories

Data category	Examples of data
Network and plant	Network description – topology, models, owners Element types, construction data CAD plans
Planning and analysis	Network expansion data for planning scenarios Historical item data – average, maximum, minimum, trends Historical network data – state estimation results
Financial	Energy accounting data Economy data Tariff structures Exchange agreements and contracts Billing and accounts data
Plant maintenance	Maintenance schedules Work orders Topographical data
System maintenance	Maintenance schedules Communications data Measurement cross reference data

The standardisation of data brings with it reduction in errors, reduction in time consuming data entry, and improved process control. On the other hand, together with standardisation, system-wide rules within the utility must be implemented to cover a number of delicate problems such as:

- authorisation of new data;
- the right to change data;
- main point of storage;
- security of data;
- handling of non-standard data;
- backup of data;
- verification of data;
- uniqueness and identification of data.

Annex B
(informative)

IEC 61968 series development process

B.1 General

This annex summarizes the modeling concepts, work steps, and deliverables of IEC Technical Committee 57 Working Group 14. It clarifies the purpose and manner of coordinating work with IEC Technical Committee 57 Working Group 13, the EPRI CCAPI and UCA projects and the Open Applications Group. This annex is only intended to provide general guidelines for the development of IEC 61968 standards and their use.

B.1.1 Application of the IEC 61968 series by a utility (see Figures B.1 to B.4)

Step A (see Figures B.1 to B.4) of the utility application process flow is the installation of suitable infrastructure to enable integration. Steps B to G of the utility application process flow are concerned with the analysis of the specific utility requirements leading to a detailed specification of utility specific message types. It is expected that these specifications will be produced as a printed report in a similar manner to future parts of the IEC 61968 series. However a utility and its suppliers may agree to exchange specifications in an appropriate electronic form, for example as produced by visual modelling tools.

Steps H to N of the utility application process flow describe the implementation and deployment of these utility specific message types. In general, an application supplier is expected to be responsible for modifying applications to produce or interpret the utility specific message types. The utility system integrator is expected to be responsible for the configuration of the Information Exchange Model (IEM) within the infrastructure. The IEM may support full or partial automatic configuration from machine-readable data produced by the applications or from electronic copies of the message specifications produced in step G.

There are three parts to the process:

– definition of the interface architecture and the major abstract components;
– definition of interface specifications of message types that describe dynamic changes;
– definition of a static entity model to provide a common way of describing what data may be exchanged.

The development of the static entity model and the messages is an iterative process.

Two process flows shown in Figures B.1 to B.4 give an overview of:

a) IEC Technical Committee 57 Working Group 14 process for developing future parts of the IEC 61968 series;
b) An overview of an utility application of the IEC 61968 series.

The steps shown in the first process flow are described in the remaining clauses of this annex.

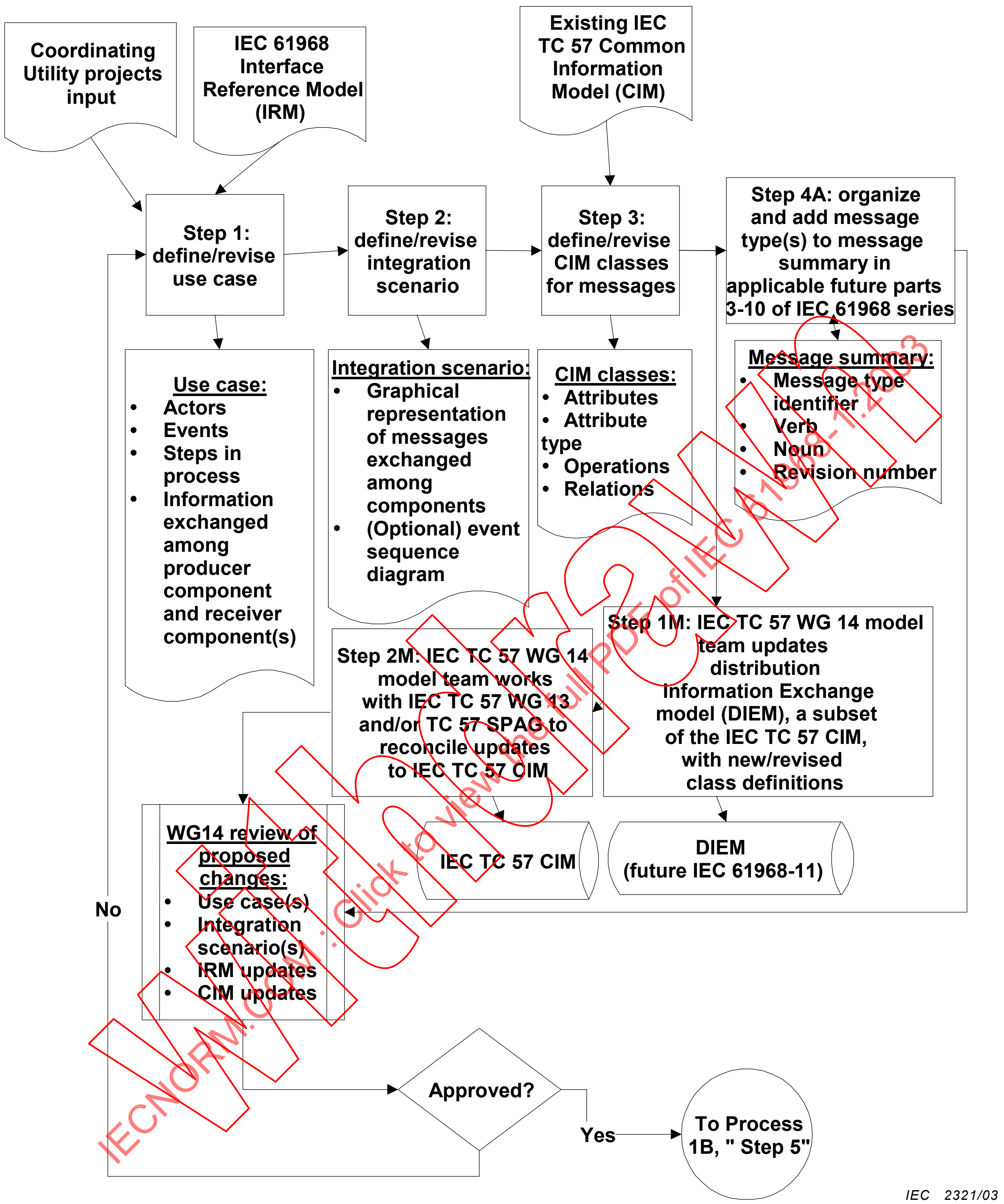


Figure B.1 – Process 1A: IEC Technical Committee 57 Working Group 14 process for developing future parts of the IEC 61968 series

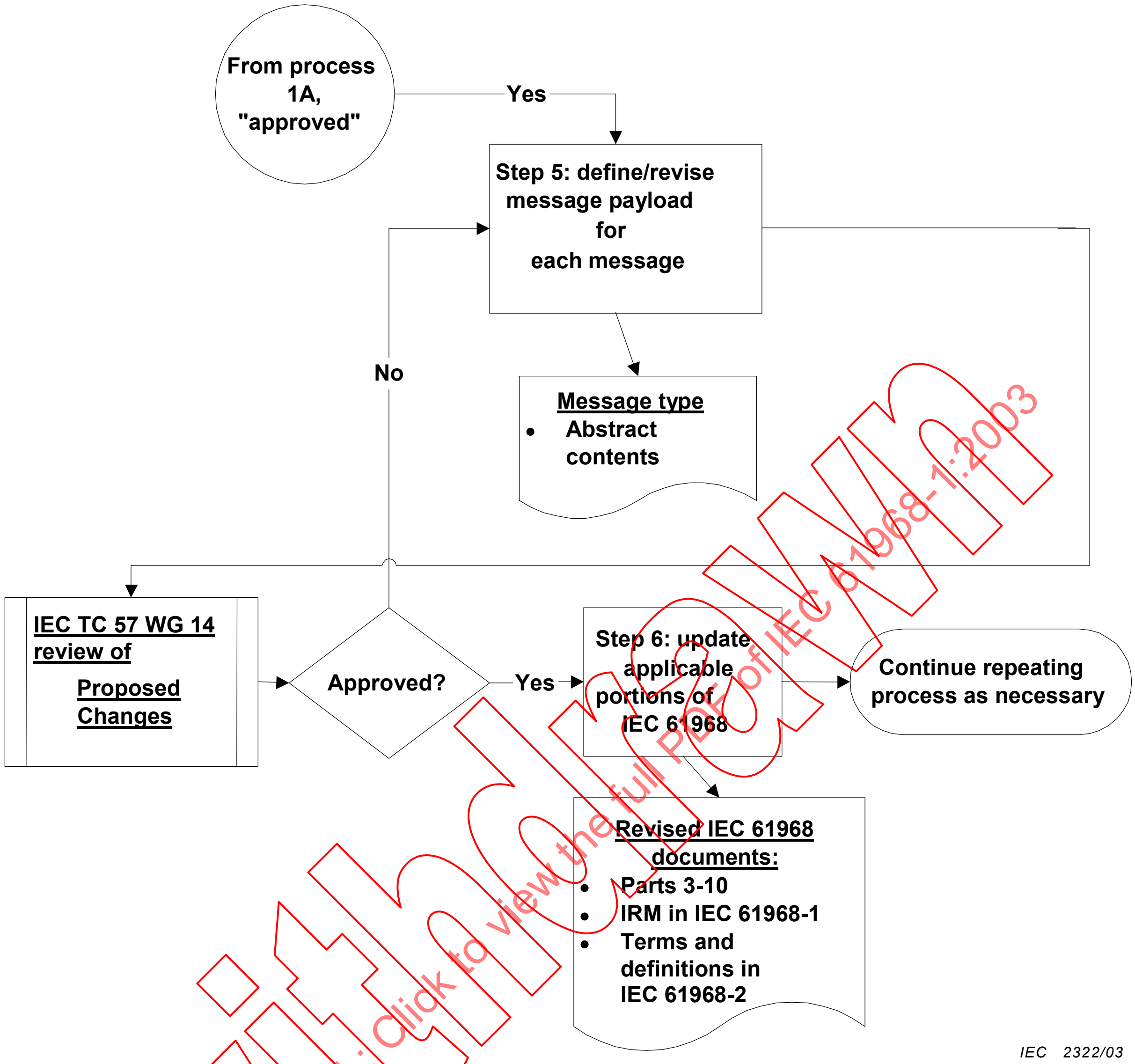


Figure B.2 – Process 1B: (Continuation) IEC Technical Committee 57 Working Group 14 process for developing future parts of the IEC 61968 series

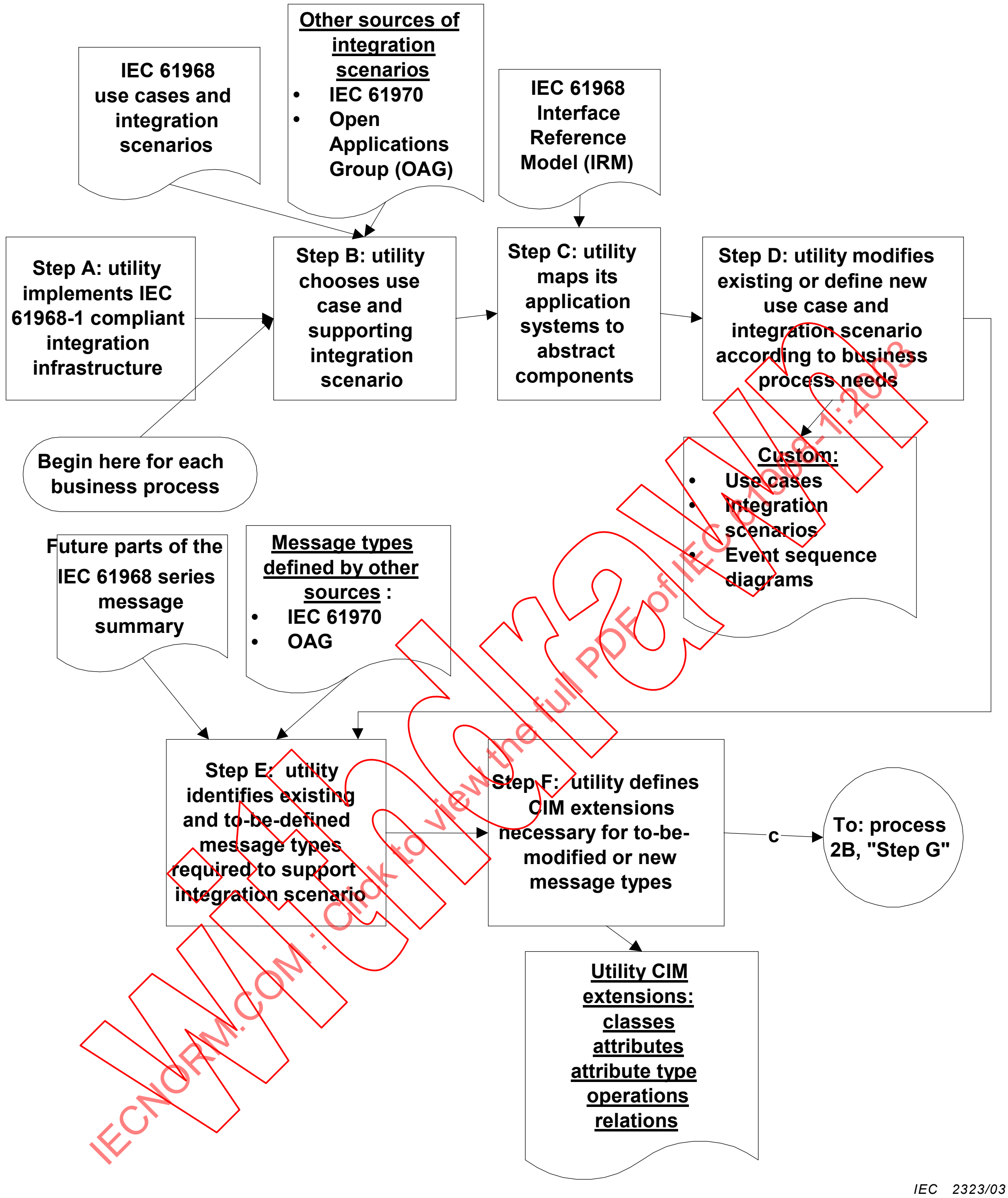


Figure B.3 – Process 2A: Typical business subfunctions of DMS and external systems

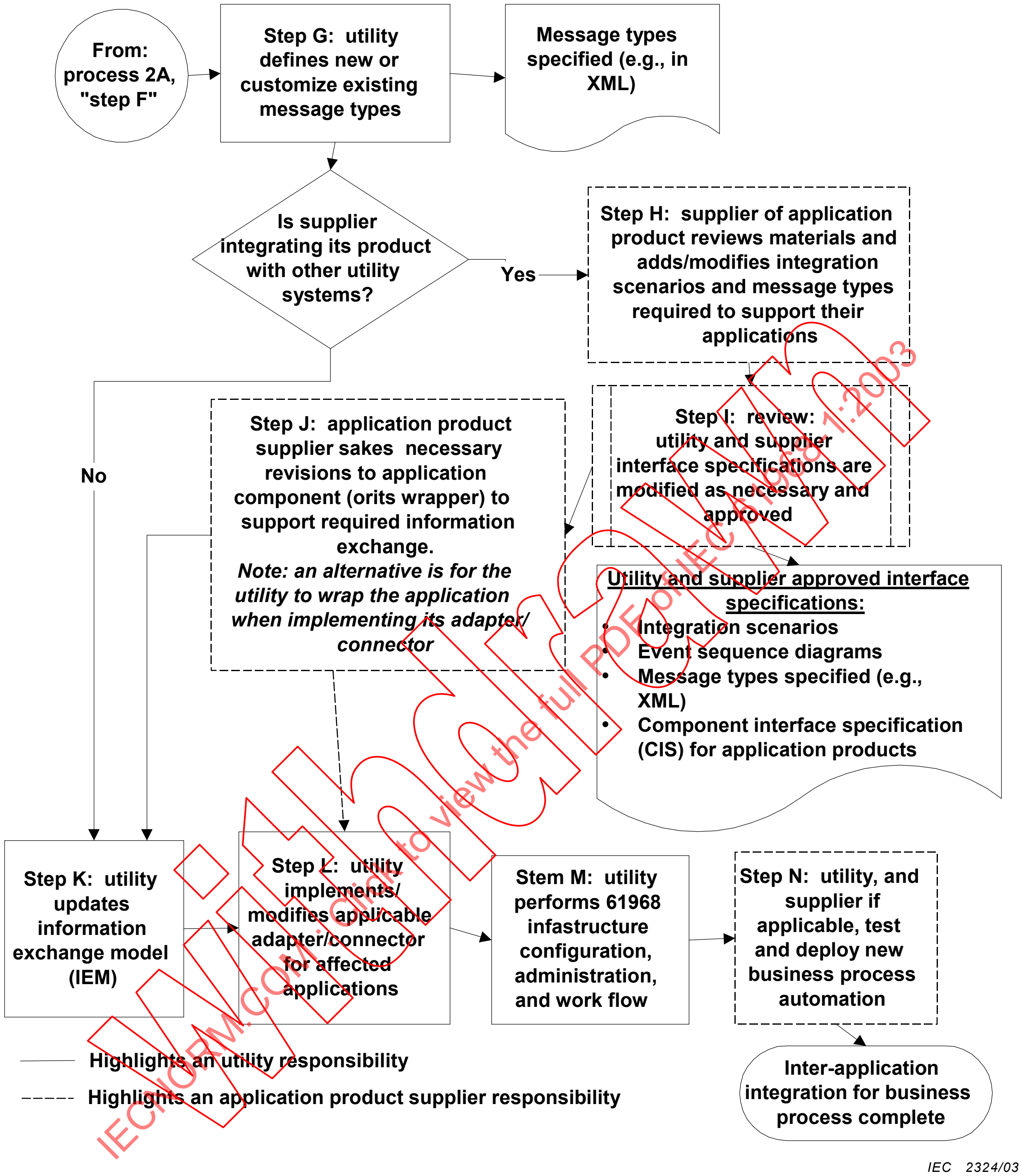


Figure B.4 – Process 2B: (continuation) an overview of an utility's application of the IEC 61968 standard

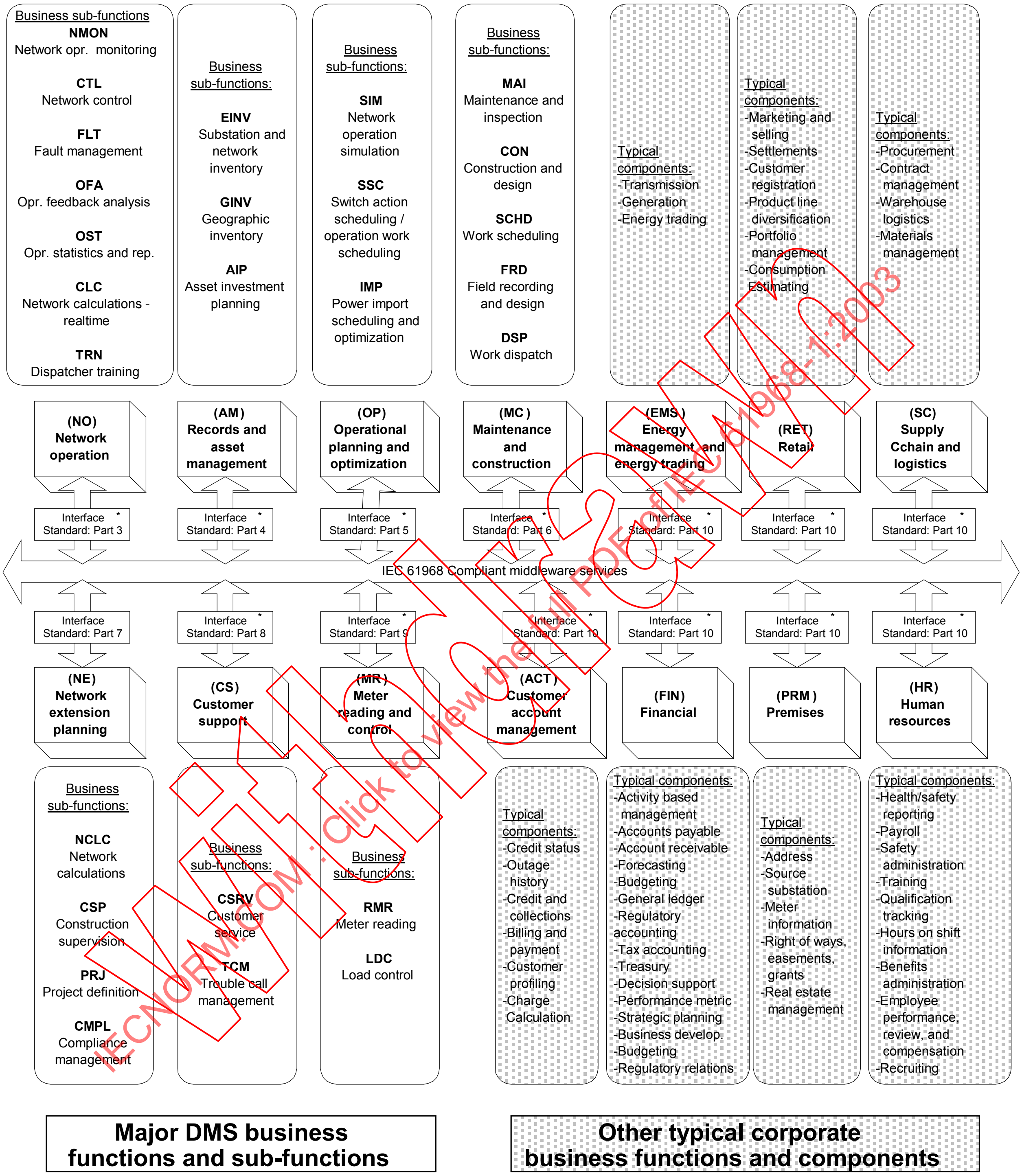
B.2 IEC 61968-1: Interface architecture and general requirements

B.2.1 Establish interface architecture and general requirements in IEC 61968-1

IEC 61968-1 is the first in a series of standards that, taken as a whole, define interfaces for the major elements of an interface architecture for Distribution Management Systems (DMS). This standard identifies and establishes requirements for standard interfaces based on an interface architecture. Subsequent standards will be developed in accordance with new work item proposals that cover each interface identified in IEC 61968-1.

At this stage, use cases, along with other available resources – like those from the CIRED Distribution Automation Working Group, EPRI CCAPI and UCA projects and the Open Applications Group (OAG) – will be used to establish general requirements of a utility's inter-application integration infrastructure and to support the definition of the Interface Reference Model (IRM), which is shown in 3.3. It is recognized that the IRM will need to be adjusted as IEC Technical Committee 57 Working Group 14 learns during the development of Parts 3-10. A key objective of IEC 61968-1 is to clearly express the "rules of engagement" for information exchange among applications so that work on future parts of the IEC 61968 series can be performed by separate teams operating in parallel, each team achieving consistent results with other teams.

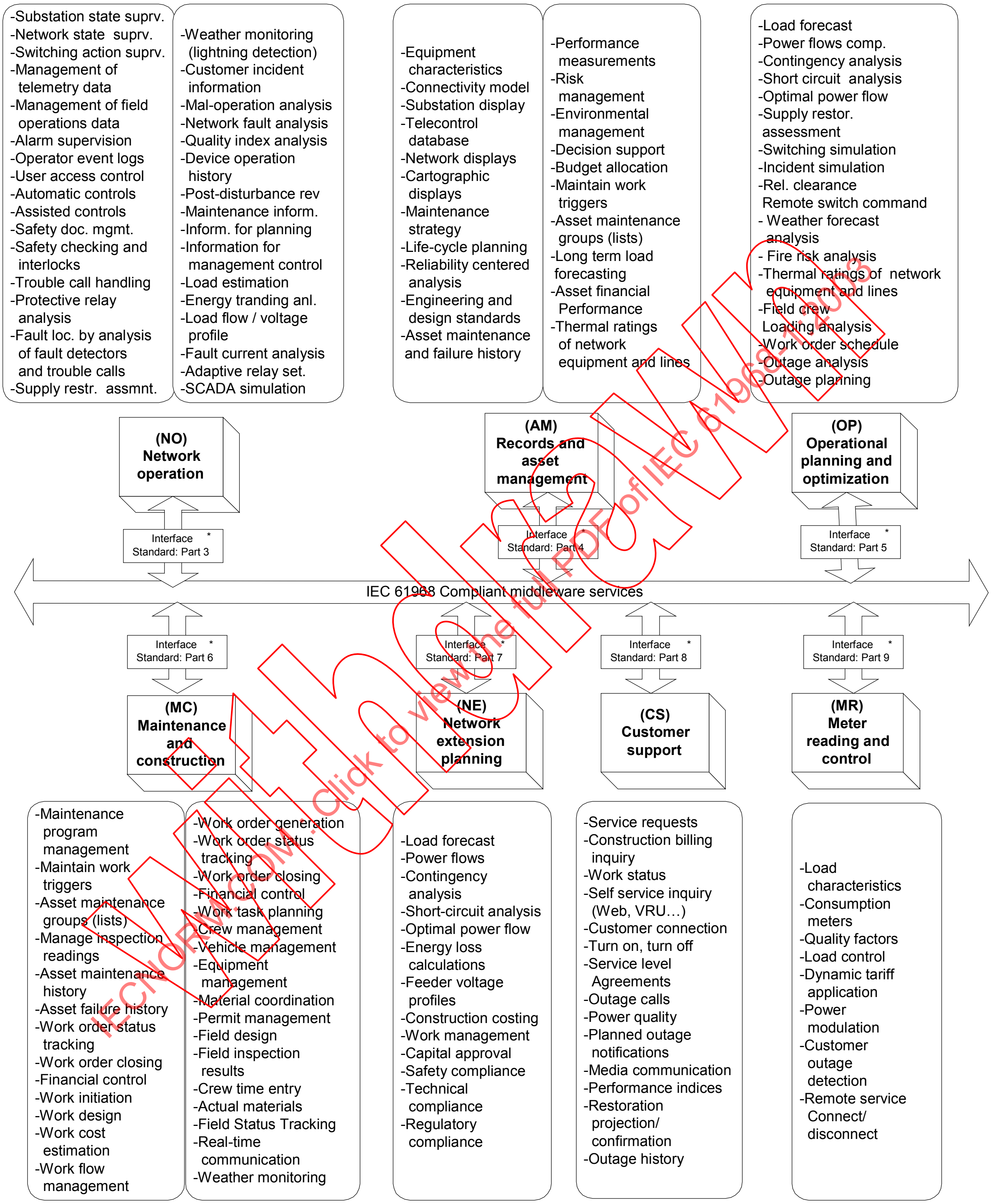
Figures B.5 and B.6 show typical components of the major DMS business functions related to the IEC 61968 interface architecture.



* Under consideration

IEC 2325/03

Figure B.5 – Typical components of major DMS business functions – Part 1



Typical components of the major DMS business functions

IEC 2326/03

* Under consideration

Figure B.6 – Typical components of major DMS business functions – Part 2

B.3 Future parts of the IEC 61968 series: interface standards for business functional areas

A vertical team is established to develop each Part 3-10 interface specification identified in the IRM. The general process followed by each team is described in the following subclauses.

The deliverables of each team include:

Normative:

- Message type table

  (for example, NewOutageRecord, UpdateOutageRecord, CancelOutageRecord)

  Columns include: class name, message type name, reference to use case(s).
- Message type definitions

  content (class/attribute pairs included in messages, references to future IEC 61968-11).

Informative:

- Integration scenarios (and reference to use case) and/or event sequence diagrams
  - description (text);
  - normal usage (for example, request/reply, publish/subscribe, subscription topic, security level);
  - pre-conditions (for the message type);
  - post-conditions;
  - error conditions (application level only, not transport or work flow).

- Relevant Common Information Model (CIM) package(s).

B.3.1 Step 1: define generic use cases

Each vertical team is to modify existing use cases and develop new use cases that establish typical information exchange requirements to/from the interface for that team's business function. Business functions, one per Part 3-10 of the IRM, are groupings of abstract application components. The purpose of use cases is to identify information to be exchanged among these components. It is not necessary to define the producer/consumer and message type columns during this step; this in done in step four of the process.

The aim of each Part 3-10 CD is to address 80 % of the most commonly needed information exchange requirements.

NOTE Developing standards for information to be exchanged among abstract components within an irm business function (i.e., internal to a vertical team's domain) is beyond the scope of IEC Technical Committee 57 Working Group 14's current work plans. However, if in the team's judgement certain intra-business function information exchanges are commonly needed, the team may elect to define these information exchange message types. During the course of its work, it should also suggest change to the interface reference model to more properly reflect inter-application integration needs of the industry.

When finished with this step for a given business process (use case), answers for the following should be available:

- WHY is information exchange required i.e. what are the use cases?
- WHERE is it all happening i.e. what is a typical context?
- WHO are the actors who use the systems and/or applications?

The use case template is shown in Table B.1.

Table B.1 – Use case template

Use case <number>: <use case name>

Summary:

Actor(s):

Name	Role description

Participating Business Functions:

Acronym	Business function/abstract component	Services or information provided

Assumptions/design considerations:

Normal sequence:

Use case step	Event	Description of process	Information to be exchanged	Producer to receiver abstract component	Message type (verb/noun)

Integration scenarios

Information model for normal sequence:

Class	Class attributes	Attribute type	Operations	Relations

Pre-conditions:

Exceptions/alternate sequences:

Post-conditions:

Information model for alternate sequence B: as-built update:

Interface class	Class attributes	Attribute type	Operations	Relations

Message type table:

Message type identifier	Message type (verb/noun)	Message type content (class attribute)	Revision number

References:

Issues:

ID	Description	Status

Revision history:

No	Date	Author	Description
1.			Original.
2			

Example steps in a use case are shown in Table B.2.

Table B.2 – Example steps in a Use Case (From: Data Acquisition for External EMS)

Use case step	Event	Description of process	Information to be exchanged	Producer to receiver abstract component	Message type (verb/noun)
1.1	External system requests mapping of names to keys	External system sends list of names	Company.name Substation.name Equipment.name Measurement.name	EXT-EMS to AM-EINV (or NO-NMON)	RequestMeasurement Identities
1.2	Return numeric keys	Looks up names Zero or negative key means name(s) not found	As above plus MeasurementUnit.name Measurement.key	AM-EINV (or NO-NMON) to EXT-EMS	ShowMeasurement Identities
2	External system subscribes to required measurements	EXT sends list of keys NMON sets up subscription table	Measurement.key	EXT-EMS to NO-NMON	SubscribeMeasurementKeys
3	NMON system sends current values This is an implicit acknowledgement to the subscription request	For each entry in each subscription table find most recent value	Measurement.key MeasurementValue.value MeasurementValue .quality	NO-NMON to EXT-EMS	ShowMeasurement Values
4	Telemetry system sends measurement event using RTU protocol	NMON Interprets RTU message; stores data in real time database; Calculates alarm states if any; raises alarms as necessary; logs changes as necessary.	Measurement.key MeasurementValue.value .quality .timestamp	In terms of the IEC 61968 series, this is all internal to NO-NMON	Not applicable

B.3.2 Step 2: define integration scenarios

Summarize results of each message exchange among IRM parts in support of the use case with an integration scenario diagram.

Coordinate the definition of these scenarios with other vertical teams. UML class definitions for the abstract components of the IRM are the entities that exchange information in the Integration Scenario Diagrams. Vertical teams work with the model development team (described below) to develop and maintain the IEC Technical Committee 57 Working Group 14 model. For a more complex interaction, it may be beneficial to first develop an event sequence diagram in UML notation that articulates each type of message used in the sequence of message exchanges among IRM parts in support of the use case.

When finished with this step, we should have the answers for:

- WHEN (due to what events) should producers start data transfers?

This diagram shows the participating components and major information exchanges. The numbers refer to the sequence steps.

See Figure B.7.

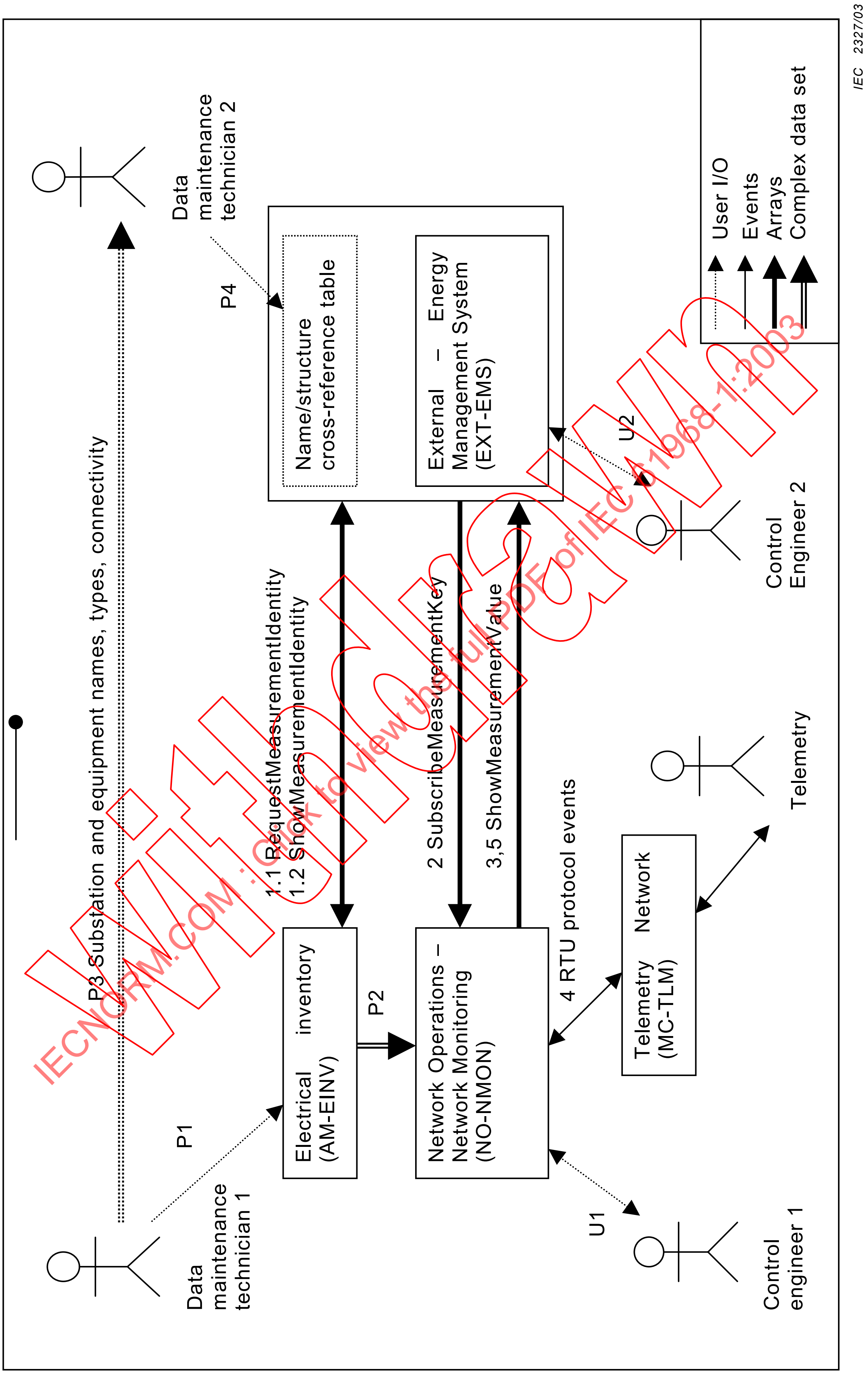


Figure B.7 – Integration scenario example (from: data acquisition for external EMS)

B.3.3 Step 3: identify and/or define CIM class(es)

Referencing the use case template (see Table B.1), this step is performed by defining an information model, as is shown in the example in Table B.3.

Table B.3 – Information model (from: data acquisition for external EMS)

Class	Class attributes	Attribute type	Domain	Relations
Company	Name	String	Globally unique	Company(1).operates. (1..N)Substations
Substation {isA.PowerSystemResource}	Name	String	Unique within Company	Substation(1).parent. .child(1..N)ConductingEquipment {Inherit from PowerSystemResource}
Equipment = ConductingEquipment {isA.PowerSystemResource}	Name	String	Unique with Substation	ConductingEquipment(1).has. (1..N)Terminals
Terminal	Name	String	Unique for type of equipment	Terminal(1).has. (1..N)Measurements
PowerSystemResource	Name	String		PowerSystemResource(1).has. (0..N)Measurements
Measurement	Name Key	String Long integer		Measurement(1).has. (1)MeasurementUnit Measurement(1).has. (0..N)MeasurementLimit Measurement(1).has. (1..N)MeasurementValue
MeasurementUnit	Name	Enumeration or string	kV, MW, MVAR, kA kW, KVAR, V, A etc.	
MeasurementLimit				Internal to NO-NMON system Or could be duplicated in EXT-EMS system
MeasurementValue	Value	Double		Value of measured quantity
MeasurementValue	Quality	Bits		As in the IEC 61850 series
MeasurementValue	TimeStamp	Date/time		
MeasurementValue	AlarmState	Enumeration or String	NORMAL, HIGH, LOW etc.	

B.3.4 Step 4: organise and add message type(s) to message summary in each applicable part of IEC 61968

Organise message types into a message type table.

Nouns are identified in the distribution information exchange model (future IEC 61968-11), which is a subset of the IEC Technical Committee 57 common information model. In general, verbs in Table B.5 shall be used unless they are inadequate to properly express the action. Key verbs from OAG are listed in Table B.6 for reference.

The IEC 61968 series does not use an identical set of verbs as OAG because we believe that the current OAG verbs are too specific in some areas and not specific enough in others. IEC 61968 requires a set of verbs to cover a publish and subscribe model from a master system point of view and a request and reply model from a requesting system point of view. A systematic way of accomplishing this is to create a set of verbs for the requesting purpose and another set of verbs with the passive voice for the publishing purpose. Verbs that apply to the master system (the system of records for the given message) will result in all referenced and/or replicated documents being updated. Verbs that apply to the requesting systems will result in a document being created or updated in the master system of that document if the request is processed successfully by the master system. This would also require integration use cases to identify a single master system for a given message document.

Table B.4 lists the commonly used verbs for the IEC 61968 series. This list of verbs can be used to form the finite number of message types under the standard. It is also recommended that the DocumentStatus attribute be used as a way to further identify the intent of CHANGE request, such as “approve”, “disapprove”, “issue”, “post”, etc. The same principal could be applied to the CHANGED message type to indicate actions such as “approved”, “disapproved”, “issued”, “posted” etc.

The following assumptions should apply when using these verbs:

- For a given message document or its parts, there is usually one system that owns the creating, updating, and cancelling/deleting/closing of that document or one for each part. The system ownership could also be extended to the attribute level if necessary to allow for multiple systems updating a document in a workflow scenario.
- A message document has a life cycle in the integration systems and is identified by a unique message id across systems upon its creation or request of creation.
- The publish and subscribe model is implied for every verb, including the ones with the passive voice.

Table B.4 – Commonly used verbs

Proposed verbs	Meaning	Message body	OAG verbs
CREATE	The CREATE verb is used to publish a request to the master system to create a new document. The master system may in turn publish the new document using the verb CREATED. The master system may also use the verb REPLY to respond to the CREATE request, indicating whether the request has been processed successfully or not.	All sections (data required to create the document)	CREATE, ADD, LOAD
CHANGE	The CHANGE verb is used to publish a request to the master system to make a change in the document based on the information in the message. The master system may in turn publish the changed document using the verb CHANGED to notify that the document has been changed since last published. The master system may also use the verb REPLY to respond to the CHANGE request, indicating whether the request has been processed successfully or not.	All sections (key(s) + data to be changed)	CHANGE, ALLOCATE, ISSUE, POST, PROCESS, RECEIVE, TRANSFER, UPDATE. These are specific forms of change and could be accomplished using the DocumentStatus.
CANCEL	The CANCEL verb is used to publish a request to the master system to cancel the document. The master system may in turn publish the cancelled message using the verb CANCELED to notify that the document has been cancelled since last published. The master system may also use the verb REPLY to respond to the CANCEL request, indicating whether the request has been processed successfully or not. The CANCEL verb is used when the business content of the document is no longer valid due to error(s).	Header information + message content key(s)	CANCEL
CLOSE	The CLOSE verb is used to publish a request to the master system to close the document. The master system may in turn publish the closed message using the verb CLOSED to notify that the document has been closed since last published. The master system may also use the verb REPLY to respond to the CLOSE request, indicating whether the request has been processed successfully or not. The CLOSE verb is used when the business document reaches the end of its life cycle due to successful completion of a business process.	Header information + message content key(s)	N/A
DELETE	The DELETE verb is used to publish a request to the master system to delete the document. The master system may in turn publish the closed message using the verb DELETED to notify that the document has been deleted since last published. The master system may also use the verb REPLY to respond to the DELETE request, indicating whether the request has been processed successfully or not. The DELETE verb is used when the business document should no longer be kept in the integrated systems either due to error(s) or due to archiving needs.	Header information + message content key(s)	N/A
GET	The GET verb is used to publish a request to the master system to get the current data for a given document reference code or a set of documents. The master system may in turn publish the document using the SHOW verb, if the document is available, or use the verb REPLY to respond to the GET request, indicating that the document is not available.	One or more document reference codes + Key(s)	GET, GETLIST

Proposed verbs	Meaning	Message body	OAG verbs
CREATED	The CREATED verb is used to publish the creation of a document as a result of either an external request or an internal action within the master system of that document. This is the first time that data for this document reference code has been published as the result of internal or external request; in which case, it would use the same document reference as the CREATE message. This message type is usually subscribed by interested systems and could be used for mass updates. There is no need to reply to this message type.	All sections	SYNC
CHANGED	The CHANGED verb is used to publish the change of a document as a result of either an external request or an internal action within the master system of that document. This could be a generic change in the content of the document or a specific status change such as “approved”, “issued” etc. This message type is usually subscribed by interested systems and could be used for mass updates. There is no need to reply to this message type.	All sections (key(s) + changed content)	SYNC
CLOSED	The CLOSED verb is used to publish the normal closure of a document as a result of either an external request or an internal action within the master system of that document. This message type is usually subscribed by interested systems and could be used for mass updates. There is no need to reply to this message type.	Header information + message content key(s)	N/A
CANCELED	The CANCELED verb is used to publish the cancellation of a document as a result of either an external request or an internal action within the master system of that document. This message type is usually subscribed by interested systems and could be used for mass updates. There is no need to reply to this message type.	Header information + message content key(s)	N/A
DELETED	The DELETED verb is used to publish the deletion of a document as a result of either an external request or an internal action within the master system of that document. This message type is usually subscribed by interested systems and could be used for mass updates. There is no need to reply to this message type.	Header information + message content key(s)	N/A
SHOW	The SHOW verb is used to publish the most current content of a document as a result of either an external GET request or an internal action within the master system of that document. This message type is usually subscribed by the requesting system(s) or other interested systems. There is no need to reply to this message type.	All sections	SHOW, LIST
REPLY	The REPLY verb is used to publish the processing result of an external request to the master system to create, change, delete, cancel, or close a document. The REPLY message type could contain specific confirmation information as to whether the request is processed successfully or not and provide alternatives if applicable. This message type is usually subscribed by the requesting systems. There is no need to reply to this message type.	Header information + message content key(s) + confirmation information + alternatives (optional)	ACKNOWLEDGE, CONFIRM, RESPOND

Proposed verbs	Meaning	Message body	OAG verbs
SUBSCRIBE	The SUBSCRIBE verb is used to publish the request to ask the master system of a document to publish a CHANGED document whenever there is a change to the document. It implies that the master system will not publish the CHANGED document unless there are one or more subscribers for the changed information.	Header information + message content key(s)	N/A
UNSUBSCRIBE	The UNSUBSCRIBE verb is used to publish the request to ask the master system of a document to stop publishing a CHANGED document whenever there is a change to the document. It implies that the master system will not publish the CHANGED document only when there are no subscribers at all.	Header information + message content key(s)	N/A

In order that information may be exchanged with applications other than those included in the IEC 61968 series of standards, verb use is intended to be consistent with the key verbs used by the Open Applications Group, which are summarized in Table B.5.