INTERNATIONAL STANDARD

ISO 13374-3

First edition
2012-02-15

Condition monitoring and diagnostics of machines — Data processing, communication and presentation —

Part 3: **Communication**

Surveillance et diagnostic d'état des machines — Traitement, échange et présentation des données —

Partie 3: Échange

Reference number
ISO 13374-3:2012(E)

COPYRIGHT PROTECTED DOCUMENT

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

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 13374-3 was prepared by Technical Committee ISO/TC 108, *Mechanical vibration, shock and condition monitoring*, Subcommittee SC 5, *Condition monitoring and diagnostics of machines*.

ISO 13374 consists of the following parts, under the general title *Condition monitoring and diagnostics of machines — Data processing, communication and presentation*:

— *Part 1: General guidelines*

— *Part 2: Data processing*

— *Part 3: Communication*

The following part is planned:

— *Part 4: Presentation*

Introduction

The various computer software systems written for condition monitoring and diagnostics (CM&D) of machines that are currently in use cannot easily exchange data or operate in a plug-and-play fashion without an extensive communication infrastructure. The lack of an all-purpose communication system makes it difficult to integrate various CM&D sub-systems and provide a unified view of the condition of machinery to users. The intent of ISO 13374 is to provide the basic requirements for open CM&D software architecture in order to allow CM&D information to be processed, communicated and displayed by various software packages independent of platform-specific or hardware-specific protocols.

ISO 13374-1 gives a general overview of data processing, communication and presentation. ISO 13374-2 provides greater details of the data-processing methodology and requirements present in today's software-enhanced systems. This part of ISO 13374 provides the requirements of the data communication architecture for open CM&D systems.

Condition monitoring and diagnostics of machines — Data processing, communication and presentation —

Part 3: Communication

1 Scope

This part of ISO 13374 specifies requirements for data communication for an open condition monitoring and diagnostics (CM&D) reference information architecture and for a reference processing architecture. Software design professionals require communications to be defined for exchange of CM&D information between software systems. This part of ISO 13374 facilitates the interoperability of CM&D systems.

2 Normative references

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

ISO 8601, *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO 13372, *Condition monitoring and diagnostics of machines — Vocabulary*

ISO 13374-1:2003, *Condition monitoring and diagnostics of machines — Data processing, communication and presentation — Part 1: General guidelines*

ISO 13374-2:2007, *Condition monitoring and diagnostics of machines — Data processing, communication and presentation — Part 2: Data processing*

ISO/IEC 19501, *Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 13372 apply.

4 Open CM&D information architecture communication requirements

4.1 Overview

An information architecture describes all the data objects and their properties (or attributes), property data types, data object relationships, reference data, and data documents for a given system or application. As specified in ISO 13374-2, an open CM&D information architecture describes the content for each of the five layers shown in Figure 1.

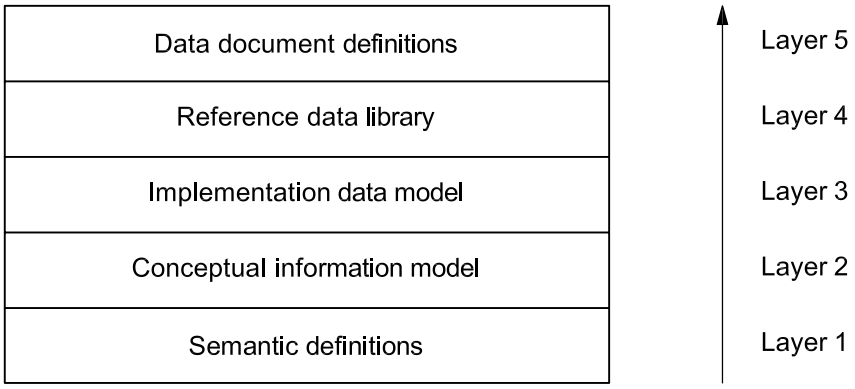

Figure 1 — CM&D Information architecture layers (from ISO 13374-2:2007)

During a communications exchange between applications in an open CM&D information architecture, the message content shall reference and validate against a defined layer 5 data document definition and comply with a defined layer 4 reference data library. Communication message implementations vary according to application requirements. Annex A details options for implementation.

4.2 Reference data library communication requirements

An open CM&D information architecture shall specify the method for communication receivers to access the defined layer 4 reference data library. The architecture shall also specify the methodology for the publication of updates from the reference data library owner to predefined subscribers.

4.3 Communications initiation requirements

An open CM&D information architecture shall specify the initiation requirements of the provider application for each method of communication the architecture includes. All date and time notations in the initiation information should reference back to a specific instant in time using the Gregorian (Common Era or CE) calendar, with a lexical representation based upon ISO 8601. Communication initiation shall also reference a defined layer 5 data document definition to which the subsequent message content complies.

4.4 Message content requirements

An open CM&D information architecture shall specify the message content requirements of the provider application for each method of communication the architecture includes. The message content definition shall reference the appropriate data document definition(s), along with the specific data format rendering of the document(s), including the compression and encryption utilized.

5 Open CM&D processing architecture communication requirements

5.1 Overview

A processing architecture describes all the interactions or transactions between modules internal to the software system itself, external to end-user interactions or external to other software system interactions. As specified in ISO 13374-1, an open CM&D processing architecture specification shall utilize the processing architecture shown in Figure 2.

This architecture is defined as blocks of data-processing functionality. After each block in the system has been properly configured, the basic data are converted into digital form in data acquisition (DA) and are processed in various ways as they are transformed into actionable information, resulting in advisory generation (AG). As the processing progresses from DA to AG, data from preceding blocks need to be transferred to subsequent blocks and additional information acquired from or sent to external systems. Similarly, as the data evolve into information, both standard technical displays and graphical presentation formats are required. In many

applications, data archiving is required in order to maintain a history of the output of each block. The DA, DM and SD blocks are responsible for assessing data quality. Output should be identified as good, bad or undetermined.

This part of ISO 13374 defines the communication requirements for any open CM&D processing architecture. With such an approach, the data-processing blocks from various suppliers can be integrated into a complete, functional system.

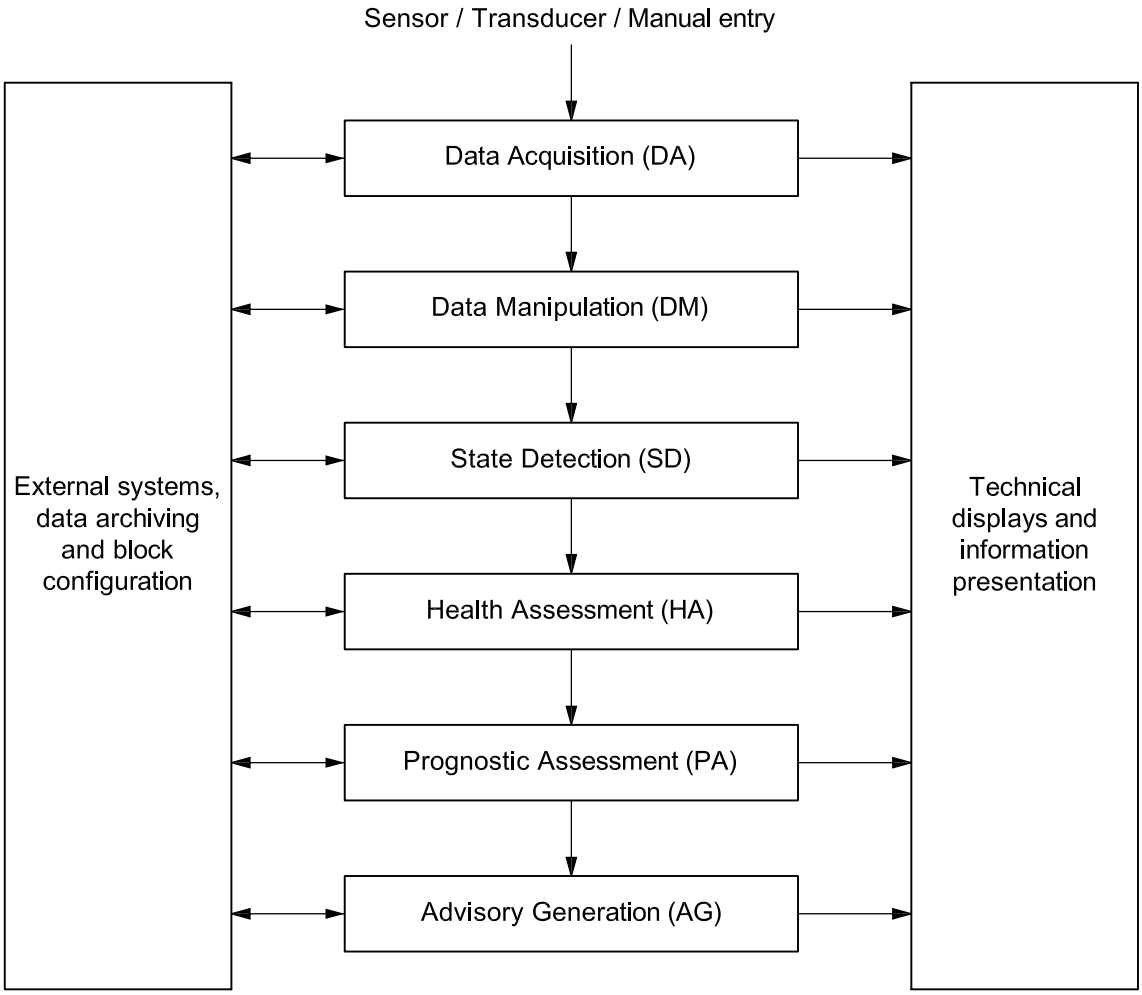

Figure 2 — Data-processing block diagram (from ISO 13374-1:2003)

5.2 Diverse technologies and UML representation

5.2.1 Introduction

There are normally different software and hardware environments as data come from sensors and are analysed by higher-level CM&D information systems. CM&D systems often start with data acquisition in embedded environments with real-time constraints. Information is then processed and refined in subsequent system blocks and made available for health assessment, prognostics, and advisory generation. These requirements currently lead to disparate technology choices. The technologies and software used by the “analysis-oriented” processing blocks (HA, PA and AG) are often different from those used in the “data-oriented” processing blocks (DA, DM and SD).

The amount of data communicated in the data-oriented blocks is vast compared to the small quantity of information generated by analysis-oriented blocks. The data-oriented blocks are normally designed for high-speed processing and often with real-time methodologies. For analysis-oriented blocks, results should be

timely, but are not usually required in milliseconds or real time. In addition, technology continues to evolve. Programming languages, network protocols, and data storage methods change over time.

An ISO/IEC 19501-compliant Unified Modelling Language (UML) model shall be specified to support the communications of the open CM&D data-processing architecture that holds the base information classes and interfaces required. As shown in Figure 3, the UML shall then be utilized to directly map into specific technologies such as XML-based web services or binary embedded system communications.

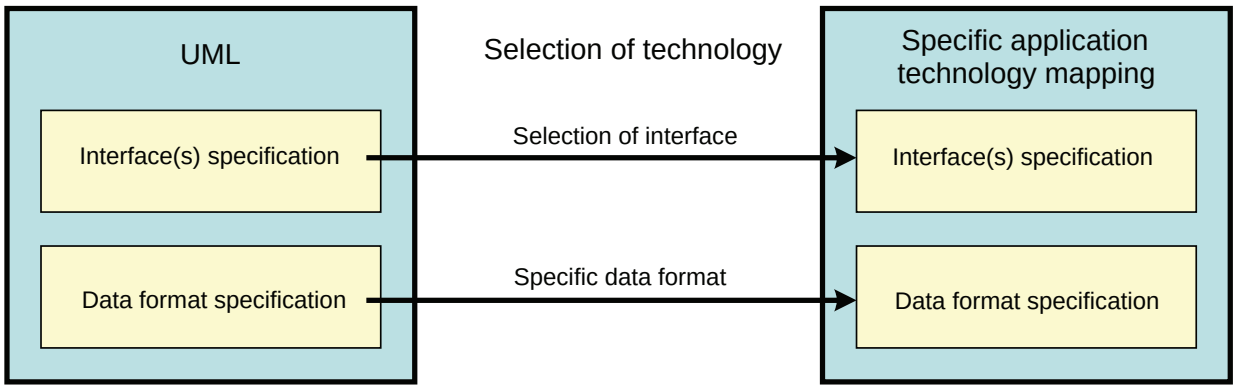

Figure 3 — UML to specific technology

5.2.2 Standard data content

When the content of the data is standardized, the conversion from one technology form to another becomes a simple one-to-one mapping effort. Thus, a binary formatted message in an embedded system can be converted using a generic adapter to an XML format when required.

5.2.3 Relationship to an information management system

In the design and management of operations of a CM&D data-processing system, it is important to have an information management system that is compliant with the open CM&D information architecture specified in Clause 4. The information management system holds not only the operational information, but also the metadata that describe the information that flows in a system. This can include the description and source of sensor signals and the processing performed on those signals. It can also include information about the agents, whether human or software, that perform analysis assessments.

These metadata enable engineering analysis and allow operational results to be used in higher-level enterprise application processing for business, logistics, and command and control decision making.

5.3 Interface types and general interaction

5.3.1 Introduction

The diverse set of technologies used in CM&D systems that use information provided by those systems requires multiple interface types. There are two major types of communication services: provider services and consumer (also called “DataUser”) services. Provider services collect and process information and provide results to interested users by some means. Consumer services utilize the provider’s CM&D data to deliver additional capabilities.

The CM&D processing system shall support the implementation of provider and/or consumer service(s). An EntryPoint shall provide the interface used for a particular service.

5.3.2 Provider interface

5.3.2.1 Introduction

All the arrows that flow downwards from the blocks defined in Figure 2 indicate data content from a provider interface. The data output of each block is a provider of information that interested consumers can receive. There are two major types of provider interfaces: synchronous and asynchronous. Systems may implement either or both types.

5.3.2.2 Synchronous interface

Providers that support the synchronous interface implement a direct call/return mechanism. A consumer block makes a call indicating which information it is interested in and the call does not return until the information requested is available. All of the requested information is then returned. A web service is a typical implementation of this type of interface. An example implementation is shown in Figure 4.

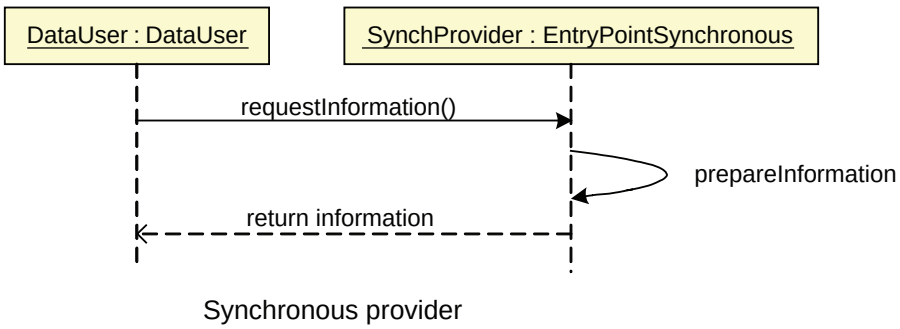

Figure 4 — Example implementation of a synchronous information request/response

In addition to processing a data request, the provider system shall support the ability for any data-processing block or external application to request processing modification. Configuration setups and threshold control are two examples. A synchronous provider shall perform the modification, if possible, and return a status (success or an error code) based on its ability to process the modification. An example implementation is shown in Figure 5.

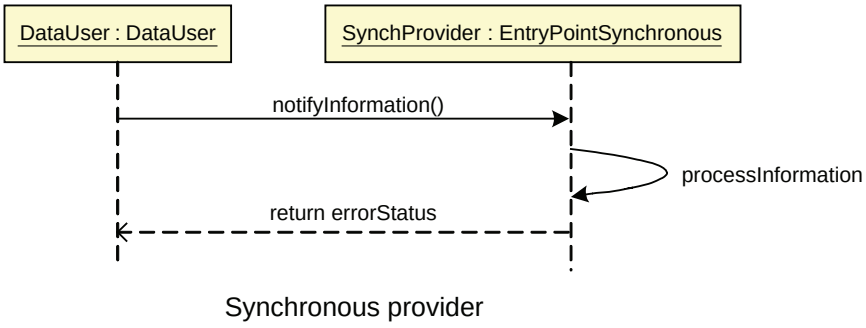

Figure 5 — Example implementation of a synchronous process modification request/response

5.3.2.3 Asynchronous interface

5.3.2.3.1 Introduction

An asynchronous interface implements a “call-but-do-not-wait” mechanism. An asynchronous interface shall allow a provider to send unsolicited information to consumers once the information about how to send

information to the consumer has been setup for the provider. Connections or data channels are established and the requested information may then be sent continually without request as it becomes available.

5.3.2.3.2 Type 1 asynchronous interface

A type 1 asynchronous interface specifies to the provider the content and the method to return information to the consumer as it becomes available. This return interface is termed a **sink**. Information is sent from the provider to the consumer via the sink interface. The provider retains information on how to send information via the consumer's sink interface. This type of interface functions as a publish-subscribe type service.

The provider's type 1 asynchronous interface shall implement a "notify connection" method that allows consumers to specify how to return information via its sink implementation. It shall also implement a remove connection interface, which allows the user to remove its reference from the provider. The consumer implements a sink interface that allows the provider to indicate when the connection information has been received and when the connection information has been removed. An example implementation is shown in Figure 6.

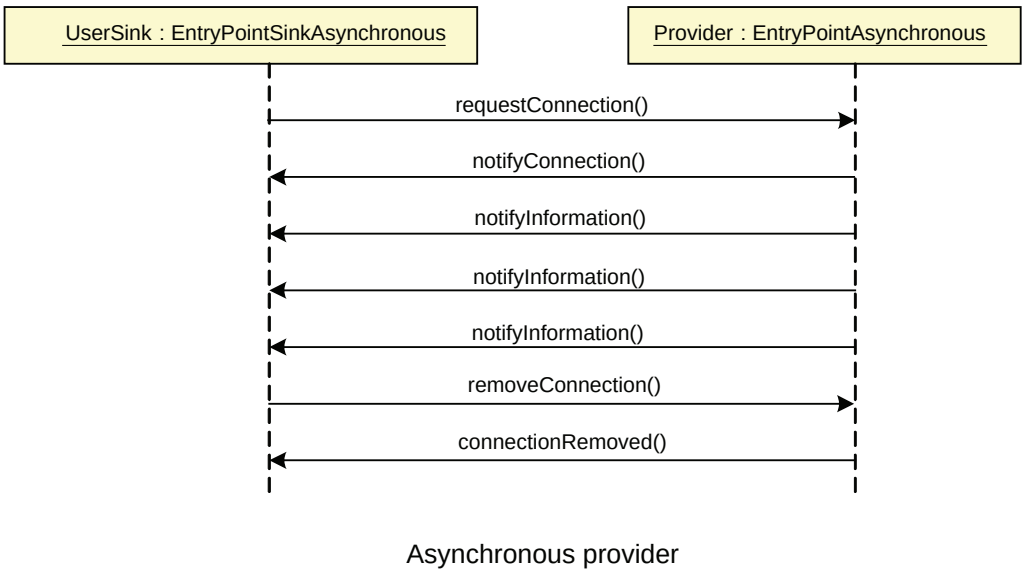

Figure 6 — Example implementation of an asynchronous information request/response

The consumer shall be allowed to indicate the requested subset of information to be returned to its sink. The consumer shall implement a sink interface for the receipt of all types of data that the consumer has subscribed to and that a provider may send. The sink shall allow for unsolicited communication of information from the provider.

This interface allows for a number of modes of communication. There shall be the capability for a "send-all-data", "send-only-on-threshold-data", and "send-only-on-request". A user provides the information to the provider about which mode is preferred. The provider has the option of sending more than requested to the consumer, but not less than requested.

5.3.2.3.3 Type 2 asynchronous interface

Embedded systems often have special communication requirements. Some systems require non-blocking one-way transfers of data. These systems may also have configuration constraints that do not allow the system to push data updates to many asynchronous users.

For systems with these types of constraints, the connection to consumers via a sink "channel" shall be established in an initialization process. Consumers shall receive data in an asynchronous manner from the provider as it becomes available. The only requirement for this type of system is the capability to send data as soon as possible in the standard format to predefined consumers. Users are required to implement the sink interface that can receive all types of information that a provider can send.

5.3.3 Consumer services

A consumer service, such as a data historian/archiver, preventive maintenance scheduling system or operations command advisory system may opt to be a user of CM&D results. The consumer service shall provide an interface that allows a data provider to send it unsolicited information. The consumer service responds with an indication of whether the information was successfully received and processed or if there were any errors. Errors could include transmission problems as well as data content processing errors. An example implementation is shown in Figure 7.

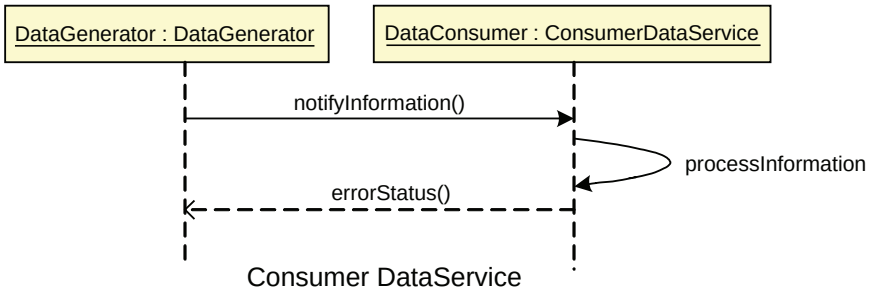

Figure 7 — Example implementation of consumer service

5.4 Specific ISO 13374-2 interface method requirements

5.4.1 Connection methods

The asynchronous interface type 1 shall implement a method to establish and remove connection information. The associated asynchronous sink interface shall implement a method to indicate when the connection has been established and when it has been removed.

5.4.2 Data event methods

Each block in the ISO 13374-2 data-processing architecture provides for data output. Every interface type shall establish a method for the communication of such data output events. The synchronous and asynchronous interface type 1 shall provide a method for requesting output data. The asynchronous interface sink type 1 shall provide a method for receiving data events. The asynchronous interface provider type 1 shall implement a method for the transmission of data events.

5.4.3 Configuration methods

Each block in the ISO 13374-2 data-processing architecture provides for the input and output of configuration information. The synchronous and asynchronous type 1 interfaces shall implement a method for input and output of configuration information. The provider shall determine to what extent actual configuration content shall be supported in accordance with application needs. If it is not supported, this shall be indicated to consumers.

5.4.4 Control methods

Control information is the ability to modify a process block. This can be in the form of expected operational parameters or preferred thresholds for alarms. The synchronous and asynchronous type 1 interfaces shall implement a method for returning control parameter setups. They shall also implement a method for changing control parameters. The provider shall determine to what extent control information is supported in accordance with application needs. If it is not supported, this shall be indicated to consumers.

5.4.5 Explanation methods

Explanation information is information that was used to develop a process block result output. It is an optional capability. If supported, the synchronous and asynchronous type 1 interfaces shall implement a method for

returning explanation information. The provider shall determine to what extent explanation information is supported in accordance with application needs. If it is not supported, this shall be indicated to consumers.

5.4.6 Application-specific methods

Every application requires initialization information and possibly additional application-specific information. The synchronous and asynchronous type 1 interfaces shall implement a method for inputting and returning application-specific information. The provider shall determine to what extent application-specific information is supported in accordance with application needs. If application-specific information is not supported the provider shall indicate that it is not supported. The consumer service is not required to support application-specific information.

5.4.7 Sender and receiver information methods

Methods to transfer metadata regarding the sender of the information shall be supported. Metadata regarding the receiver application where information should be sent, when applicable based on the transfer mode, shall be supported.

5.4.8 Error messages

Every application requires a method for the indication of errors in internal operation and to notify users.

5.4.9 Block processing methods

Table 1 provides the basic block processing methods that shall be utilized by each supported block in an open CM&D data-processing system.

Table 1 — Information types

Information	Mandatory?	Meaning
DataEvent	Yes	DataEvent communicates system CM&D outputs applicable to each block level. For example, a module at the DA block level communicates sensor information with DA DataEvent types. A module at the DM block level communicates processed information in DM-type DataEvent types.
Configuration	No	Configuration information indicates how a module is set up to process information. This includes preferred inputs, algorithm descriptions, and types of output. This could also include a list of monitored components and types of faults that are monitored.
Control	No	Control is the ability to modify how a module is performing the processing. This is application-specific.
Explanation	No	Explanation is the ability to indicate which data were used for a particular output. This could either be the data themselves or some type of pointer to the data used.
Application-specific	No	This information type is for setup information and other application-specific information needs.
Sender information	Yes	Sender information indicates the source of information. There shall be a way to indicate the sender of the information.
Receiver information	No	Receiver information is information about where the information should be sent. This information is used by some communication types, but not all. The use of this is dependent upon the application type.

5.5 Data provider specification support considerations

The provider of information selects the data-processing block(s), e.g. DA, DM, SD, HA, PA or AG, it supports and the block processing methods supported for each block. The provider also selects which optional interfaces and services that it supports. These are chosen based on the needs of the application environment. Figure 8 summarizes the areas that need to be specified.

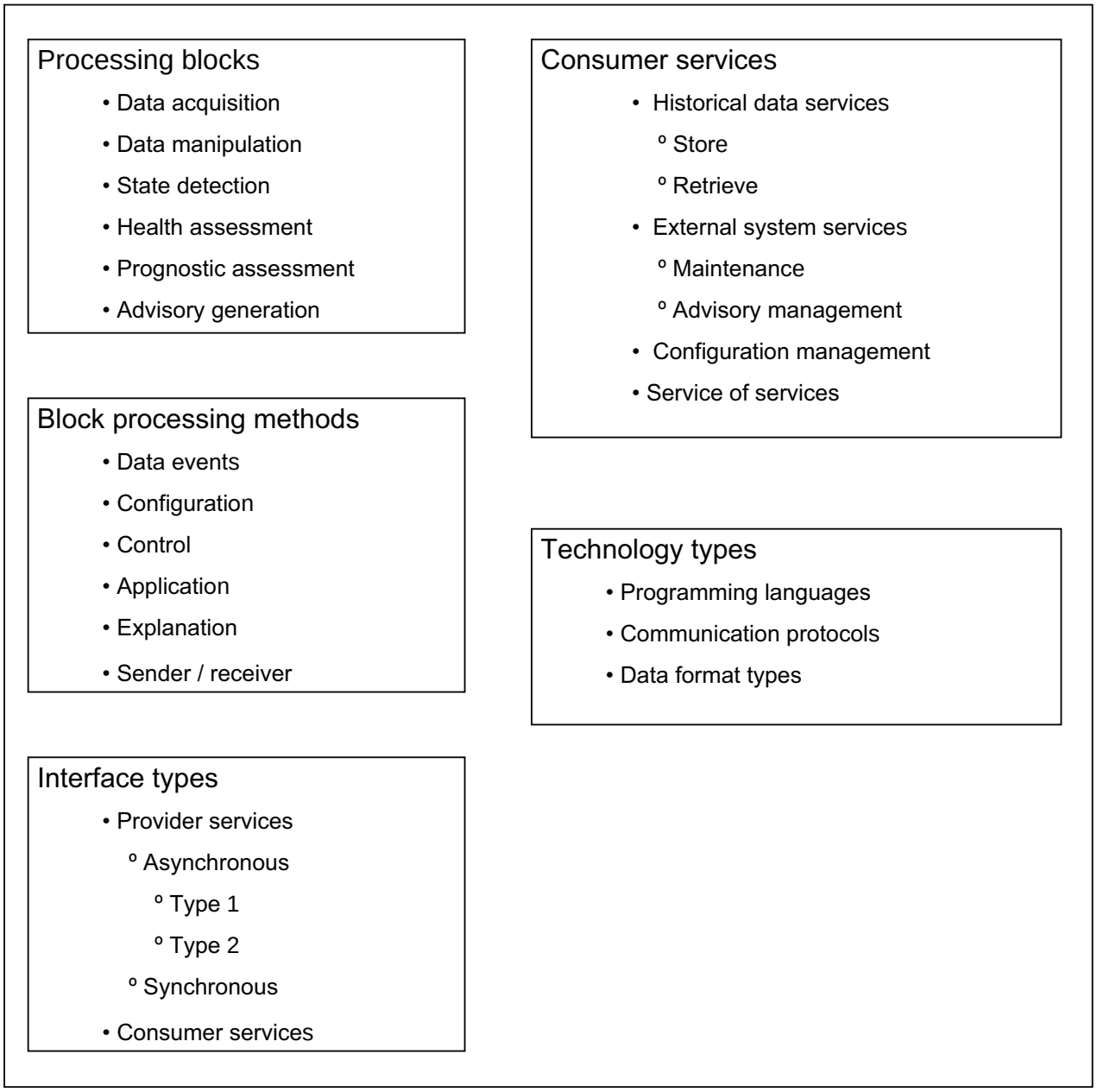

Figure 8 — Data provider specification support areas

Annex A
(informative)

Open CM&D information architecture based on IEC 62264-5[1]

A.1 Message formats

A.1.1 Typical message structure

A.1.1.1 Introduction

Communication message structures for common transactions are specified in IEC 62264-5[1]. This annex provides an example of a compliant open CM&D information architecture. For message-based communication, IEC 62264-5[1] defines two main areas in a message, as shown in Figure A.1, an *application identification area* and a *data area*. Within the *data area* there is often a *verb area* and a *noun area*.

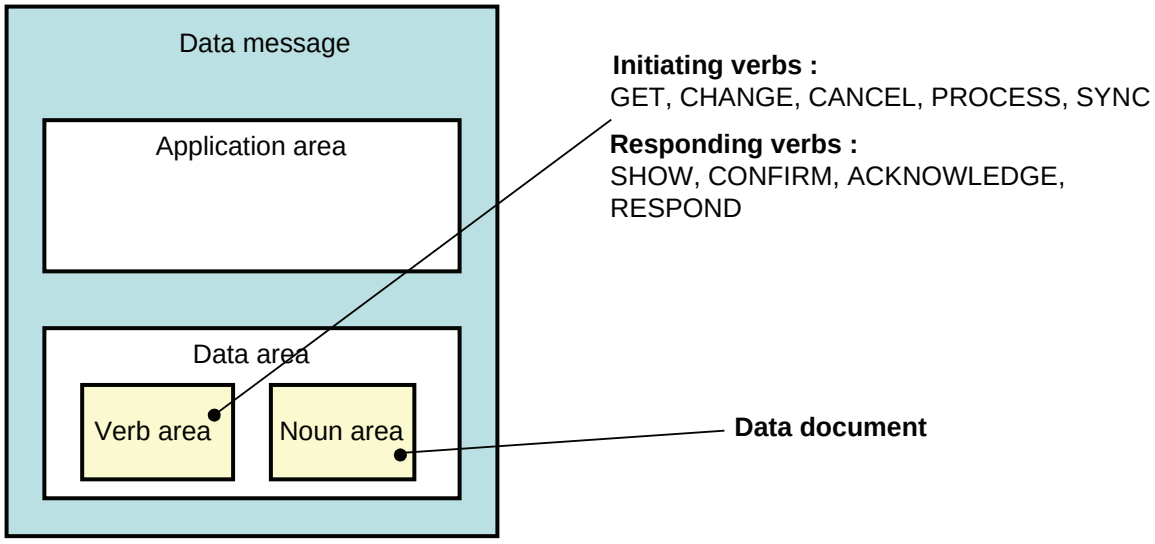

Figure A.1 — IEC 62264-5[1] message structure

A.1.1.2 Application area

The application identification area carries information that a receiving application uses to handle a message. The application identification area is used for the application layer of communication, such as indicating a required confirmation of message processing. This information typically includes the electronic address of the sender, an indication of the confirmation requirement, and the date and time the message was created. The application identification area may also include other information required for identification and authentication of the messages. Figure A.2 illustrates a typical layout for an application identification area. Dates and times include time zone information in order to unambiguously identify times, such as co-ordinated universal time or ISO 8601 CE (Common Era) calendar extended format.

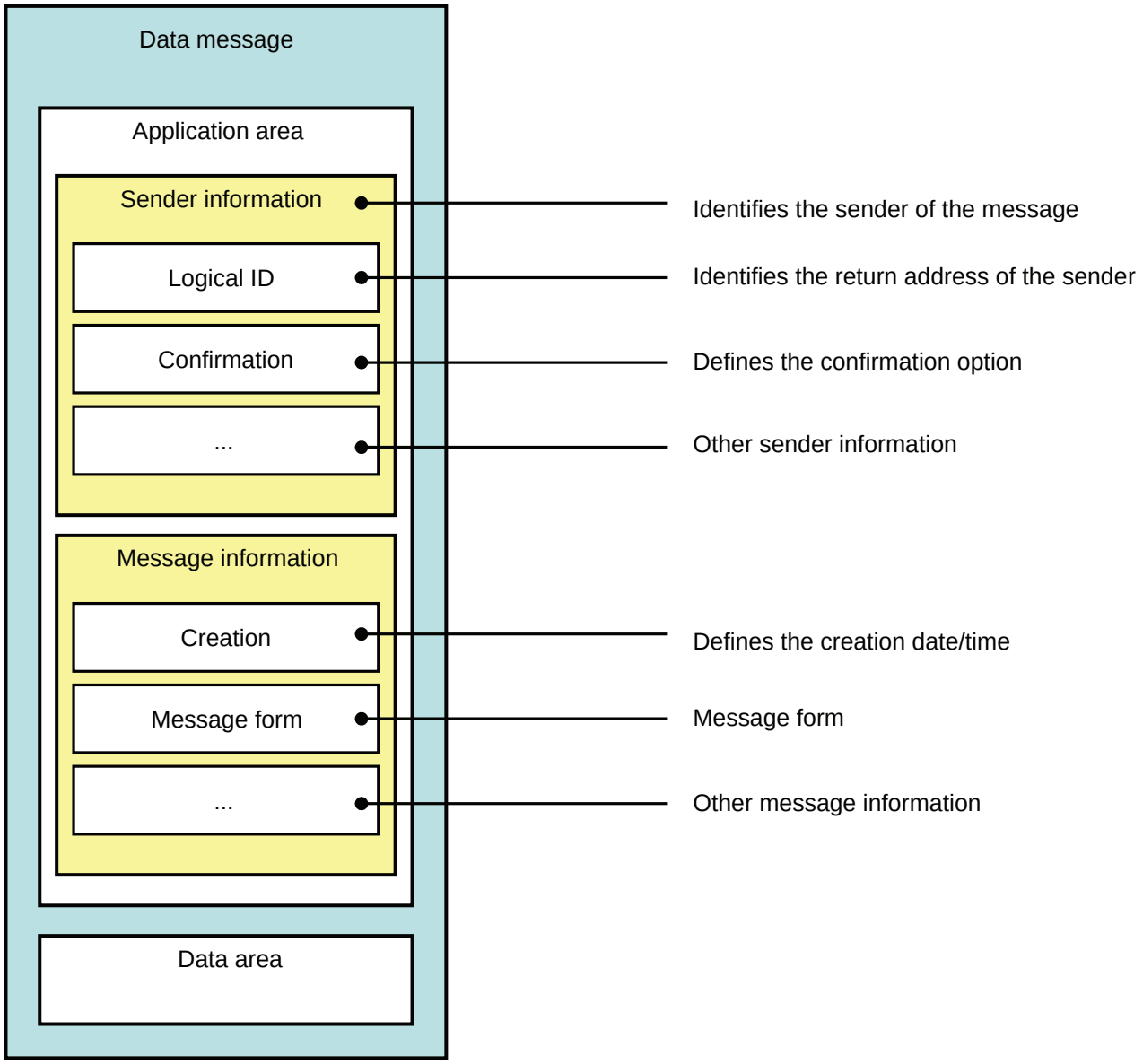

Figure A.2 — Typical layout of an application identification area

A.1.1.3 Data area

The data area in a message normally contains a verb area and a noun area. The verb-noun combinations define messages that are intended to have a unique and unambiguous meaning.

A.1.1.3.1 Verb area

A.1.1.3.1.1 Overview

The verb area contains operation names (methods) and associated elements that represent the actions to be performed by the receiving application or the response to a request by the sending application. Verbs are used to effect communication of information between a sender and receiver. Verbs fall into two categories: initiating verbs and responding verbs. Table A.1 details commonly used initiating verbs and Table A.2 lists commonly used responding verbs.

Table A.1 — Initiating verbs

Initiating verbs	Meaning
GET	The GET verb is used to communicate a request for information on an object or list of objects. Valid responding verbs: SHOW and CONFIRM.
PROCESS	The PROCESS verb is used to request processing of the associated noun by the receiving application. Valid responding verbs: ACKNOWLEDGE and CONFIRM.
CHANGE	The CHANGE verb is used when the sender of the message is sending a request for the data to be changed. Valid responding verbs: RESPOND and CONFIRM.
CANCEL	The CANCEL verb is used when the sender of the CANCEL message is sending a request for the data to be cancelled. Valid responding verb: CONFIRM.
SYNC ADD	A SYNC ADD verb is sent by the owner of the information and indicates that the owner of the information has added new information. Valid responding verb: CONFIRM.
SYNC CHANGE	A SYNC CHANGE verb is sent by the owner of the information and is used to disseminate information on changed objects to subscribed users. Valid responding verb: CONFIRM.
SYNC DELETE	A SYNC DELETE verb is sent by the owner of the information and indicates that the provider of the information has deleted the information. Valid responding verb: CONFIRM.

Table A.2 — Responding verbs

Responding verbs	Meaning
SHOW	The SHOW verb is used when responding to a GET message. Valid responding verb: CONFIRM.
ACKNOWLEDGE	The ACKNOWLEDGE verb is used to indicate an application's receipt of a PROCESS request. The response to a PROCESS message is an ACKNOWLEDGE message. Valid responses include: ACCEPTED, REJECTED, and MODIFIED. No valid responding verb.
CONFIRM	The CONFIRM verb is used to signify the application receipt and processing of any message other than ACKNOWLEDGE, RESPOND, and CONFIRM messages that have requested a confirmation "OnError" or "always". No valid responding verb. If the message could not be processed, an error condition should be returned with a description of the error.
RESPOND	The RESPOND verb is used to signify the application receipt and processing of a CHANGE message. Valid responses include: ACCEPTED, REJECTED, and MODIFIED. No valid responding verb.

A.1.1.3.1.2 Initiating verb descriptions

A.1.1.3.1.2.1 GET

The GET verb is used to communicate a request for information on an object or list of objects.

The response to the GET message is a SHOW message.

Figure A.3 — GET and SHOW transaction

The GET is designed to retrieve one or more objects and any contained objects by using the ID attribute.

Within a GET message, the ID of the requested object is passed to the provider of the information. Where a single ID is not sufficient identification, such as when a property of an object is needed, then the ID of the encapsulating object, and the ID or value of the encapsulated object (the property) is passed to the provider of the data. The identifying IDs are specified in the sections for each object type.

When a wildcard definition is used in the ID, then the GET returns a list of objects matching the wildcard specification.

A.1.1.3.1.2.2 PROCESS

The PROCESS verb is used to request processing of the associated noun by the receiving application. A PROCESS message is sent to an entity that can process the object. In a typical exchange scenario, a PROCESS message is considered to be the equivalent of a formal command.

NOTE A PROCESS verb is often the equivalent of a command to add an object, but usually the receiving entity does further processing of the information.

A PROCESS verb area contains an optional element with one of the following additional definitions: Never or Always (see Table A.3). If the optional element is not specified, the default is Never.

Table A.3 — Acknowledge request options

Name	Description
Never	No ACKNOWLEDGE message requested.
Always	Always send an ACKNOWLEDGE message.

A.1.1.3.1.2.3 CHANGE

The CHANGE verb is used when the sender of the message is sending a request for the data to be changed. The noun area contains the new data. Figure A.4 illustrates a CHANGE message with a RESPOND message.

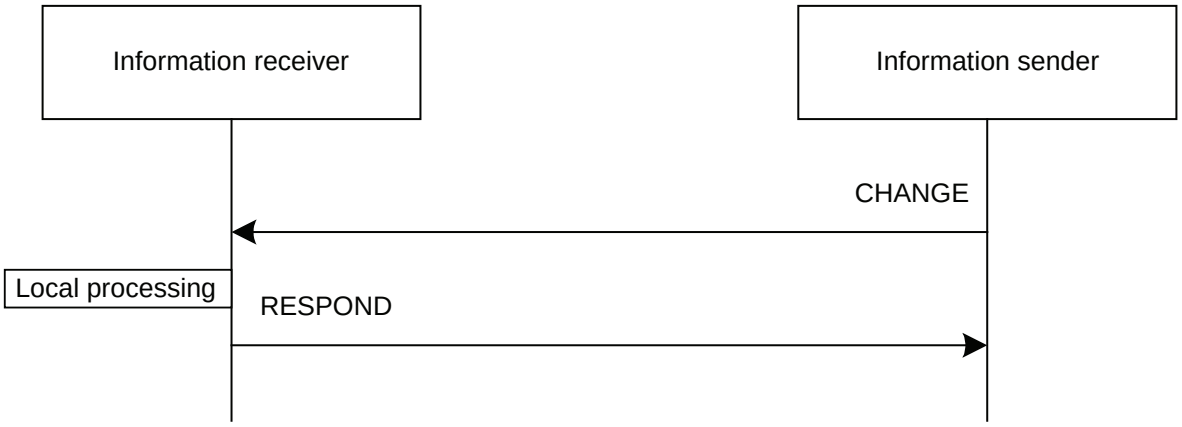

Figure A.4 — CHANGE/RESPOND transaction

A CHANGE verb area contains an optional element with one of the following additional definitions: Never or Always. (See Table A.4.) If the optional element is not specified, then it defaults to Never.

Table A.4 — Respond options

Name	Description
Never	No RESPOND message requested.
Always	Always send a RESPOND message.

A.1.1.3.1.2.4 CANCEL

The CANCEL verb is used when the sender of the CANCEL message is sending a request for the data to be cancelled.

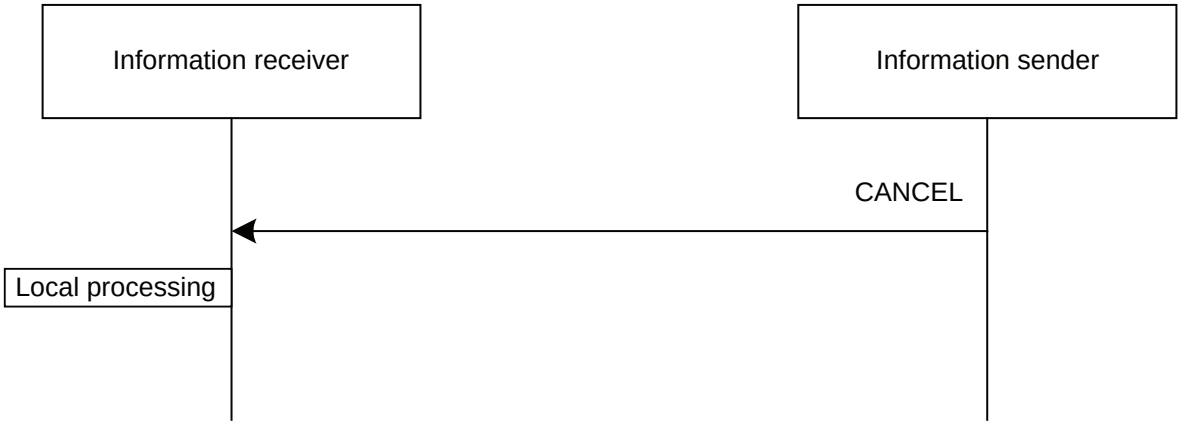

Figure A.5 — CANCEL message

A.1.1.3.1.2.5 SYNC

The SYNC verb is used when the owner of the data is publishing the information or change in information to subscribers.

NOTE 1 SYNC is short for synchronize and implies synchronized or aligned data; it does not mean synchronous communications.

There should only be one application that sends SYNC messages for any specific element of information.

NOTE 2 Other applications may send SYNC messages for elements of information that they own.

The owner of the information sends the SYNC message.

The SYNC message contains one of the following modifiers in the verb area: ADD, CHANGE or DELETE.

The timing of the publication and scope of the published information is not defined in a message. It is determined by an out-of-band agreement between the publisher and subscriber.

EXAMPLE This verb is commonly used when mass changes are necessary, such as when a device publishes an update for multiple systems or when a publish-subscribe mechanism is used as a company's integration architecture.

A SYNC ADD verb is sent by the owner of the information and indicates that the owner of the information has added new information. The SYNC ADD message includes the object instances added and the values of all attributes of these objects.

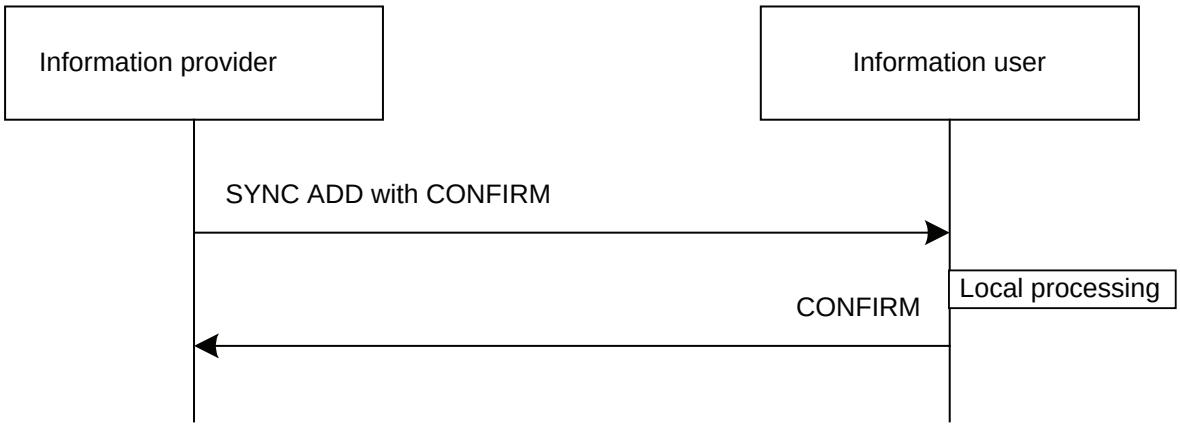

Figure A.6 — SYNC ADD transaction with confirmation

A SYNC CHANGE verb is sent by the owner of the information and is used to disseminate information on changed objects to subscribed users. The SYNC CHANGE message includes the object instances changed with the values of the attributes changed.

A.1.1.3.1.2.6 SYNC DELETE

A SYNC DELETE verb is sent by the owner of the information and indicates that the provider of the information has deleted the information. The SYNC DELETE message includes the object instances deleted.

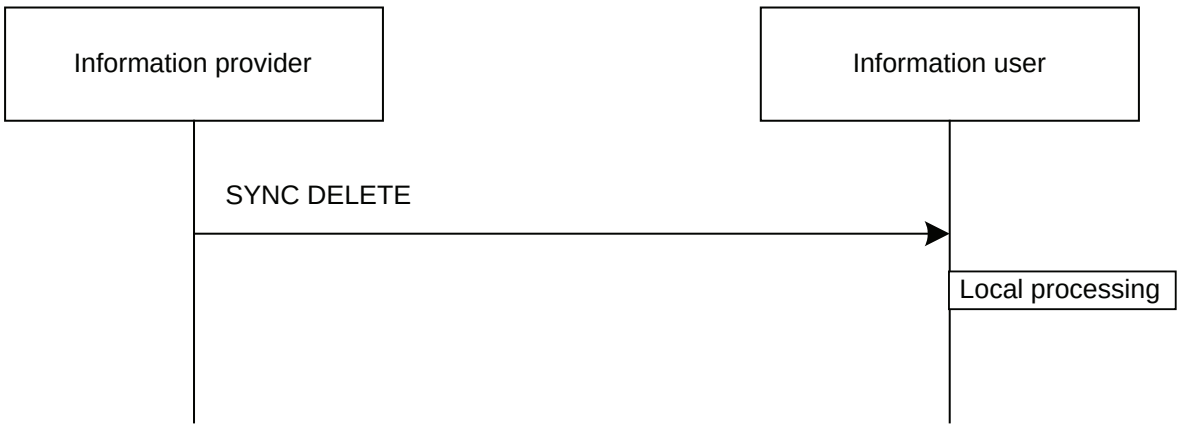

Figure A.7 — SYNC DELETE transaction with no confirmation

NOTE A SYNC DELETE message only indicates that the provider has deleted the information from publication. The information can still be archived or retained in accordance with business policies, but is not available for further publishing. The information user has the responsibility to determine the correct action, such as retaining or archiving their information.

A.1.1.3.1.3 Responding verb descriptions

A.1.1.3.1.3.1 SHOW

The SHOW verb is used when responding to a GET message.

Figure A.8 illustrates a transaction with a GET message followed by a SHOW message and a CONFIRM message (because of the “Confirm Always” option specified with the GET message).

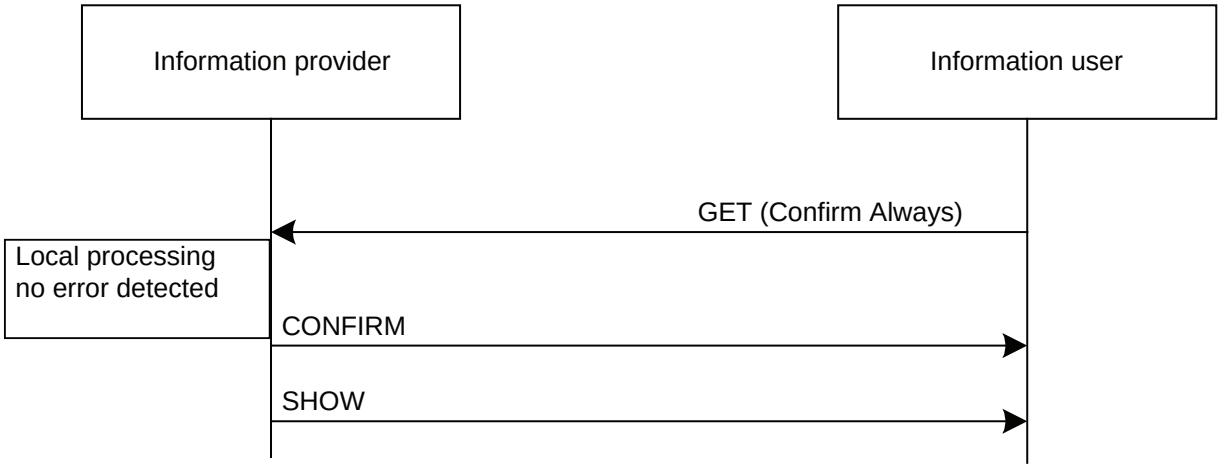

Figure A.8 — GET and SHOW transaction with a Confirm Always

NOTE The order of arrival of the CONFIRM message, SHOW message and any other response message is not defined.

A.1.1.3.1.3.2 ACKNOWLEDGE

The ACKNOWLEDGE verb is used to indicate an application’s receipt of a PROCESS request. The response to a PROCESS message is an ACKNOWLEDGE message. The ACKNOWLEDGE message may return the original or modified data. Figure A.9 illustrates a PROCESS message with a response ACKNOWLEDGE message.

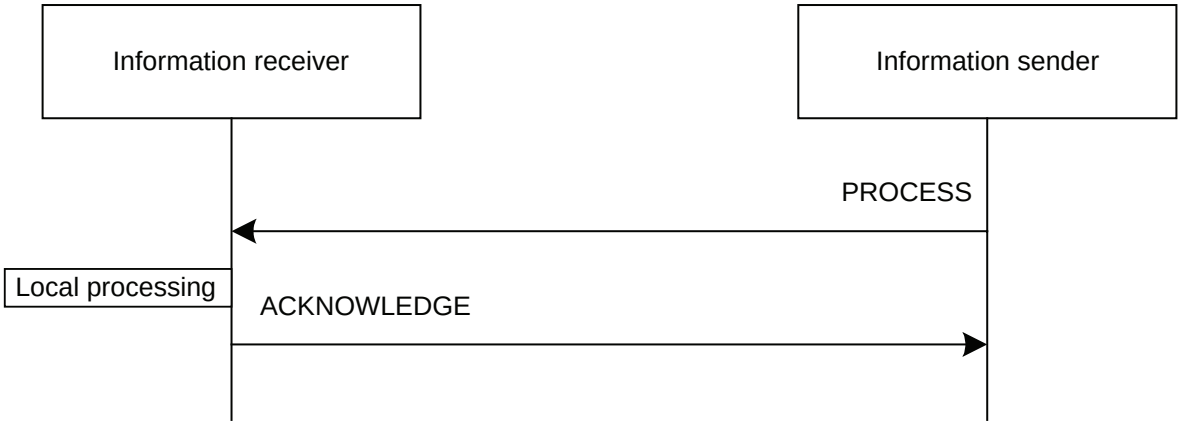

Figure A.9 — PROCESS/ACKNOWLEDGE transaction

An ACKNOWLEDGE verb area contains an element with one of the following additional definitions: ACCEPTED, REJECTED or MODIFIED. (See Table A.5.)

Table A.5 — Acknowledge element

Acknowledge element	Definition
ACCEPTED	The information was accepted by the receiver of the information and was processed according to the business rules of the receiver.
REJECTED	The information was rejected by the receiver of the information and was not processed by the receiver. The message data area contains an identification of the reason for rejection.
MODIFIED	The information was accepted by the receiver of the information, but was modified for correct processing, the modified data is returned with the ACKNOWLEDGE. The message data area contains an identification of the type of modification.

EXAMPLE Figure A.10 shows a CM&D monitoring schedule (route) sequence sent from a master CM&D configuration system to a CM&D execution system. The initial PROCESS message with a monitoring schedule is received with a specified polling periodicity and an ACKNOWLEDGE message with a MODIFIED flag is returned with a longer polling periodicity since the execution system calculated that the requested one was not possible for execution. Upon receipt of this modified schedule, the CM&D monitoring scheduling system decides to shorten the schedule by removing a sensor, but keeping the periodicity as first requested and resends this to the CM&D execution system. The CM&D execution system accepts the schedule and returns an ACKNOWLEDGE message with an ACCEPTED flag.

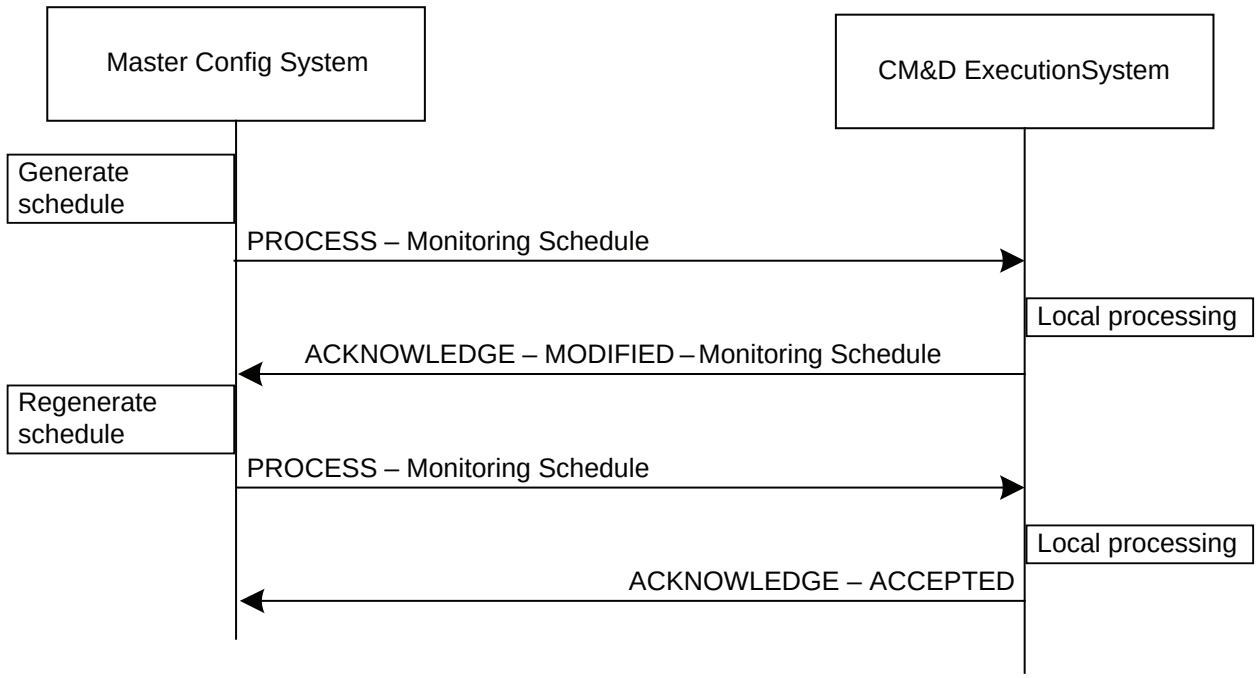

Figure A.10 — Example of acknowledge to a process message

A.1.1.3.1.3.3 CONFIRM

A CONFIRM verb is used in a CONFIRM message for confirmation of receipt and processing of any message other than the CONFIRM, RESPOND or ACKNOWLEDGE messages. See Figure A.11 for an example of confirmation with detected errors.

Confirmation is an option controlled by the sending business application. It is a request to the receiving application to send back a confirmation message to the sender of the initiating message.

The CONFIRM message indicates an identification of the initiating message being confirmed.

The CONFIRM message indicates the successful processing of the initiating message or returns error conditions if the initiating message could not be processed. The error condition should include a description of the error.

If an error occurs in the processing of the initiating message by the receiving application and the sender has set the confirmation element to either OnError or Always, then the receiving application provides a CONFIRM message. If no confirmation option was specified, then the default value is Confirm Never.

Error handling at the application layer is through the confirmation element in the application identification area.

The application error handling is in addition to any communication layer error handling that may be provided by the infrastructure framework, web service or middleware.

A confirmation request has the values defined in Table A.6.

Table A.6 — Confirmation request options

Name	Description
Never	No confirmation requested.
OnError	Send back a confirmation only if an error has occurred.
Always	Always send a confirmation regardless of the local processing.

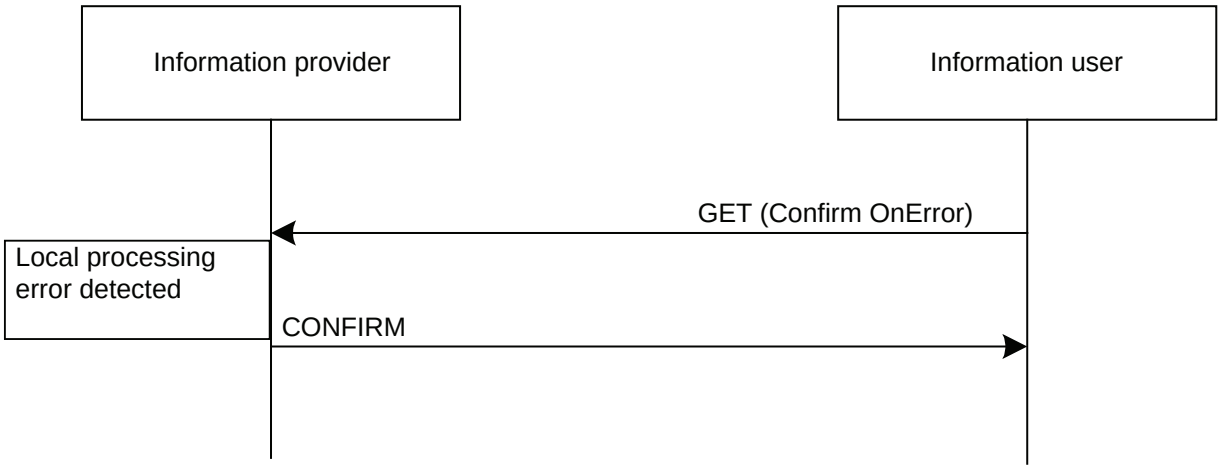

Figure A.11 — Example of a GET message with Confirm OnError

(The order of arrival of the CONFIRM message and any other response message is not defined in this part of ISO 13374.)

The error description, code or text associated with a CONFIRM message is contained in the noun area.

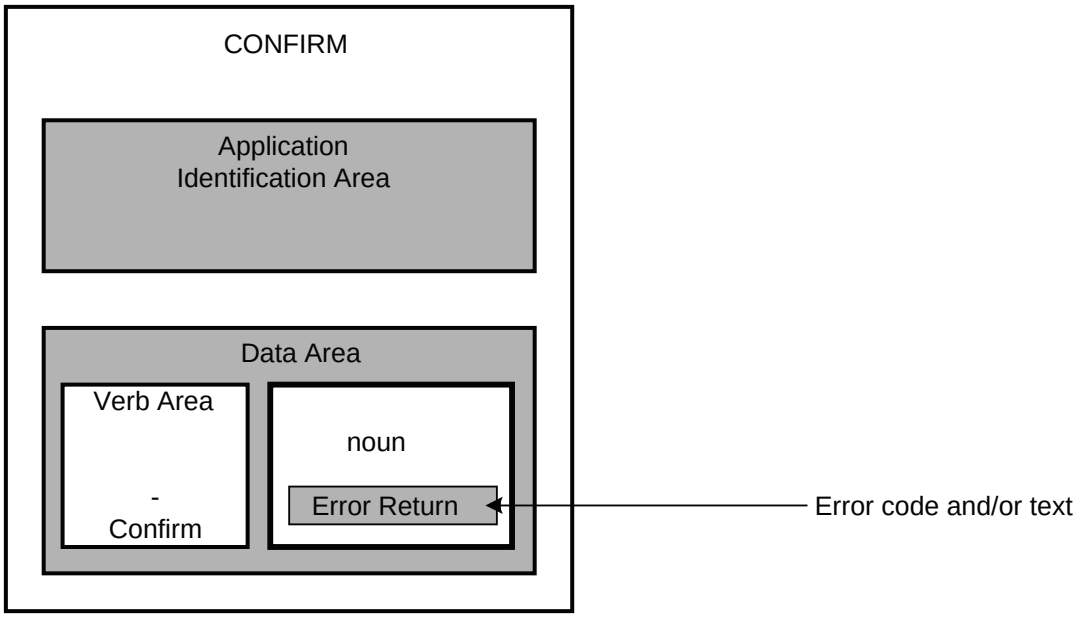

Figure A.12 — Confirm message

Specific error codes or error text are not defined in this part and are implementation-specific.

A.1.1.3.1.3.4 RESPOND

The RESPOND verb is used to signify the application receipt and processing of a CHANGE message. The RESPOND message is used when responding to a CHANGE message. The RESPOND message may return the original or modified data.

A RESPOND verb area contains an element with one of the following additional definitions: ACCEPTED, REJECTED or MODIFIED. (See Table A.7.)

Table A.7 — Respond element

Respond element	Definition
ACCEPTED	The information was accepted by the receiver of the information and was changed according to the business rules of the receiver.
REJECTED	The information was rejected by the receiver of the information and was not changed by the receiver. The message data area contains an identification of the reason for rejection.
MODIFIED	The information was accepted by the receiver of the information but was modified for correct processing and the modified data were returned with the RESPOND. The message data area contains an identification of the type of modification.

A.1.2 Noun area

The noun area contains nouns and associated elements that represent one or more Data Document definition objects defined in such a manner that they can be unambiguously communicated as an object in this transport definition.

A.2 Communication methods

A.2.1 Typical communication method options

The following communications options can be utilized in enterprise application integration.

A.2.1.1 “Push”: Point-to-point “Push Data” sender-receiver

Can be performed in an asynchronous manner (sender does not wait for a response from the receiver) or a synchronous manner (sender waits for response from the receiver). Valid for PROCESS, CHANGE, CANCEL, and SYNC transactions.

A.2.1.2 “Push Scatter/Update”: Point-to-middleware-to-targets “Push Data” sender-receiver

Can be performed in an asynchronous manner (sender does not wait for a response from the middleware) or a synchronous manner (sender waits for response from the receiver). Valid for PROCESS, CHANGE, CANCEL, and SYNC transactions.

A.2.1.3 “Pull”: Point-to-point “Pull Data” client-server

Can be performed in an asynchronous manner (client does not wait for a response from the server) or a synchronous manner (client waits for a response from the server). Valid for GET transactions.

A.2.1.4 “Scatter/Gather Pull”: Point-to-middleware-to-sources Broadcast “Pull Data” client-server

Can be performed in an asynchronous manner (client does not wait for a response from the middleware receiver) or a synchronous manner (client waits for a response from the middleware receiver). Valid for GET transactions.

A.2.1.5 “Broadcast”: Point-to-everyone broadcast sender-receivers without middleware distribution engine

Can be performed in an asynchronous manner (sender does not wait for a response from the receiver) or a synchronous manner (sender waits for a response from the server). Valid for SYNC transactions.

A.2.1.6 “Broadcast Optimized”: Point-to-middleware-to-everyone broadcast sender-receivers with middleware distribution engine (middleware handles message deliveries)

Can be performed in an asynchronous manner (sender does not wait for a response from the middleware receiver) or a synchronous manner (sender waits for a response from the middleware receiver). Valid for SYNC transactions.

A.2.1.7 “Pub-Sub”: Publisher-to-subscribers without middleware distribution engine

Can be performed in an asynchronous manner (publisher does not wait for a response from the subscriber receiver) or a synchronous manner (publisher waits for a response from the subscriber receiver). Valid for SYNC transactions.

A.2.1.8 “Pub-Sub Optimized”: Publisher-to-middleware-to-subscriber(s) with middleware distribution engine, also called “Fire-and-forget” (Middleware handles subscriber management and message deliveries)

Can be performed in an asynchronous manner (publisher does not wait for a response from the middleware receiver) or a synchronous manner (publisher waits for a response from the middleware receiver). Valid for SYNC transactions.

A.2.1.9 “Blackboard”: Writer-to-blackboard-and-readers-to-blackboard (Blackboard provides a common shared area for multiple readers to access the written information during a restricted timeframe)

Can be performed in an asynchronous manner (writer does not wait for a response from the Blackboard receiver) or a synchronous manner (writer waits for a response from the Blackboard receiver). Valid for SYNC transactions.

Bibliography

[1] IEC 62264-5, *Enterprise-control system integration — Part 5: Business to manufacturing transactions*

ICS 17.160.00; 35.240.99

Price based on 21 pages