

FINAL VERSION

VERSION FINALE



**Energy management system application program interface (EMS-API) –
Part 453: Diagram layout profile**

**Interface de programmation d'application pour système de gestion d'énergie
(EMS-API) –
Partie 453: Profil de disposition de diagramme**

CONTENTS

FOREWORD.....	4
INTRODUCTION.....	6
1 Scope.....	7
2 Normative references	7
3 Terms and definitions	7
4 Use Cases.....	8
4.1 General use cases for diagram exchange	8
4.2 Simple bay diagram example	10
5 Diagram layout exchange profile.....	12
5.1 General.....	12
5.2 Diagram layout profile classes	13
5.2.1 General	13
5.2.2 Diagram.....	14
5.2.3 DiagramObject.....	14
5.2.4 DiagramObjectGluePoint	15
5.2.5 DiagramObjectPoint.....	16
5.2.6 DiagramObjectStyle	16
5.2.7 TextDiagramObject.....	16
5.2.8 VisibilityLayer	17
5.2.9 Abstract classes – IdentifiedObject	18
5.2.10 Enumerations – OrientationKind	18
6 Graphical rendering	19
6.1 General.....	19
6.2 Single point objects.....	19
6.3 Multiple point objects	20
6.4 Gluing points.....	21
6.5 Diagram object style	22
6.6 Diagram layout exchange process	22
7 Examples	24
7.1 Data instantiation and encoding	24
7.2 Simple bay example use case.....	26
Annex A (informative) Benefits and format conversion from IEC 61970-453 Edition 1 to Edition 2	28
Bibliography.....	29
Figure 1 – System overview.....	9
Figure 2 – Bay diagram as rendered	10
Figure 3 – Bay diagram drawn in GIS style	11
Figure 4 – Bay diagram drawn in SCADA style.....	12
Figure 5 – Diagram layout information model	13
Figure 6 – Conventions used for representing object instances and associations	19
Figure 7 – Single point diagram objects	20
Figure 8 – Multiple point diagram objects	21

Figure 9 – Disconnector with glue point	21
Figure 10 – Disconnector and breaker with glue points	22
Figure 11 – Profiles within IEC standards	23
Figure 12 – Disconnector diagram object instantiation	24
Figure 13 – IEC 61970-552 Encoding for disconnector diagram data	25
Figure 14 – Bay diagram example with objects outlined GIS style	26
Figure 15 – Bay diagram example with objects outlined SCADA/EMS style	27

Single user licence
EEESC WG on Smart Grids
IECNORM.COM : Click to view the full PDF of IEC 61970 WG 453:2018
No reproduction or circulation
Sep 2024

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**ENERGY MANAGEMENT SYSTEM APPLICATION
PROGRAM INTERFACE (EMS-API) –****Part 453: Diagram layout profile****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 itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 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.

This consolidated version of the official IEC Standard and its amendment has been prepared for user convenience.

IEC 61970-453 edition 2.1 contains the second edition (2014-02) [documents 57/1409/FDIS and 57/1430/RVD] and its amendment 1 (2018-11) [documents 57/2038/FDIS and 57/2054/RVD].

This Final version does not show where the technical content is modified by amendment 1. A separate Redline version with all changes highlighted is available in this publication.

International Standard IEC 61970-453 has been prepared by IEC technical committee 57: Power systems management and associated information exchange.

This edition includes the following significant technical changes with respect to the previous edition:

- a) The SVG elements and its data model have been replaced by the Diagram Layout Package, which is now an integral part of the IEC 61970-301 (CIM) model.
- b) The exchange is in accordance with and is a part of the IEC 61970 profile concept.
- c) A glue point object has been introduced to model explicit connections between graphics elements.

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

A list of all parts in the IEC 61970 series, published under the general title *Energy management system application program interface (EMS-API)*, can be found on the IEC website.

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

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

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

This standard is part of the IEC 61970 series that define an application program interface (API¹) for an Energy Management System (EMS²).

The IEC 61970-3x series specify a Common Information Model (CIM³): a logical view of the physical aspects of EMS information. The IEC 61970-3x series includes IEC 61970-301, *Common Information Model (CIM) Base*.

This standard is one of the IEC 61970-4x series that define utility control centre component interface specifications (CIS⁴). IEC 61970-4x specifies the functional requirements for interfaces that a component (or application) shall implement to exchange information with other components (or applications) and/or to access publicly available data in a standard way. The component interfaces describe the specific message contents and services that can be used by applications for this purpose. The implementation of these messages in a particular technology is described in the IEC 61970-5x series.

Energy Management Systems employ a variety of schematic and quasi-geographic presentations in their user interfaces. These are sometimes generated automatically, but more often are hand-drawn and require considerable labour to create and maintain. Most of this labour goes into the arrangement, or 'layout' of the power system elements within the overall diagram. When network models are exchanged, as defined in IEC 61970-452 and IEC 61968-13 standards, it is desirable to be able to exchange these layouts.

IEC 61970-453 specifies guidelines for the exchange of diagram layout information for schematic data that is encoded using IEC 61970-552.

¹ Footnote 1 applies to the French version only.

² Footnote 2 applies to the French version only.

³ Footnote 3 applies to the French version only.

⁴ Footnote 4 applies to the French version only.

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

Part 453: Diagram layout profile

1 Scope

This part of IEC 61970 is a member of the IEC 61970-450 to 499 series that, taken as a whole, defines, at an abstract level, the content and exchange mechanisms used for data transmitted between control centre components.

Included in this part of IEC 61970 are the general use cases for exchange of diagram layout data, and guidelines for linking the layout definitions with CIM data. Guidelines for management of schematic definitions through multiple revisions are also included.

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.

IEC 60050, *International electrotechnical vocabulary*

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

IEC 61970-501, *Energy management system application program interface (EMS-API) – Part 501: Common Information Model Resource Description Framework (CIM RDF) schema*

IEC/TR 62541-1, *OPC Unified Architecture – Part 1: Overview and concepts*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 60050, as well as the following, apply.

3.1 domain object⁵

instance of a class that models a Real-World Object⁶ with a unique identity

Note 1 to entry: A domain object inherits from a CIM *IdentifiedObject*. A domain object is normally not a diagram object.

3.2 diagram⁷

electronic equivalent of a seamless paper plan

⁵ Footnote 5 applies only to the French version.

⁶ Footnote 6 applies only to the French version.

⁷ Footnote 7 applies to the French version only.

Note 1 to entry: The diagram is an identified container for the diagram objects. Examples of diagrams include substation schematics, transportation or distribution network orthogonal schematics, or pseudo-geographical schematics. A diagram has a well-defined coordinate space.

3.3

diagram object⁸

representation of domain objects or static background

Note 1 to entry: The diagram is composed of diagram objects.

Note 2 to entry: An example for domain objects includes breakers. An example for static background object includes lakes.

3.4

diagram object style

definition of how to render diagram objects possibly based on the state of domain objects

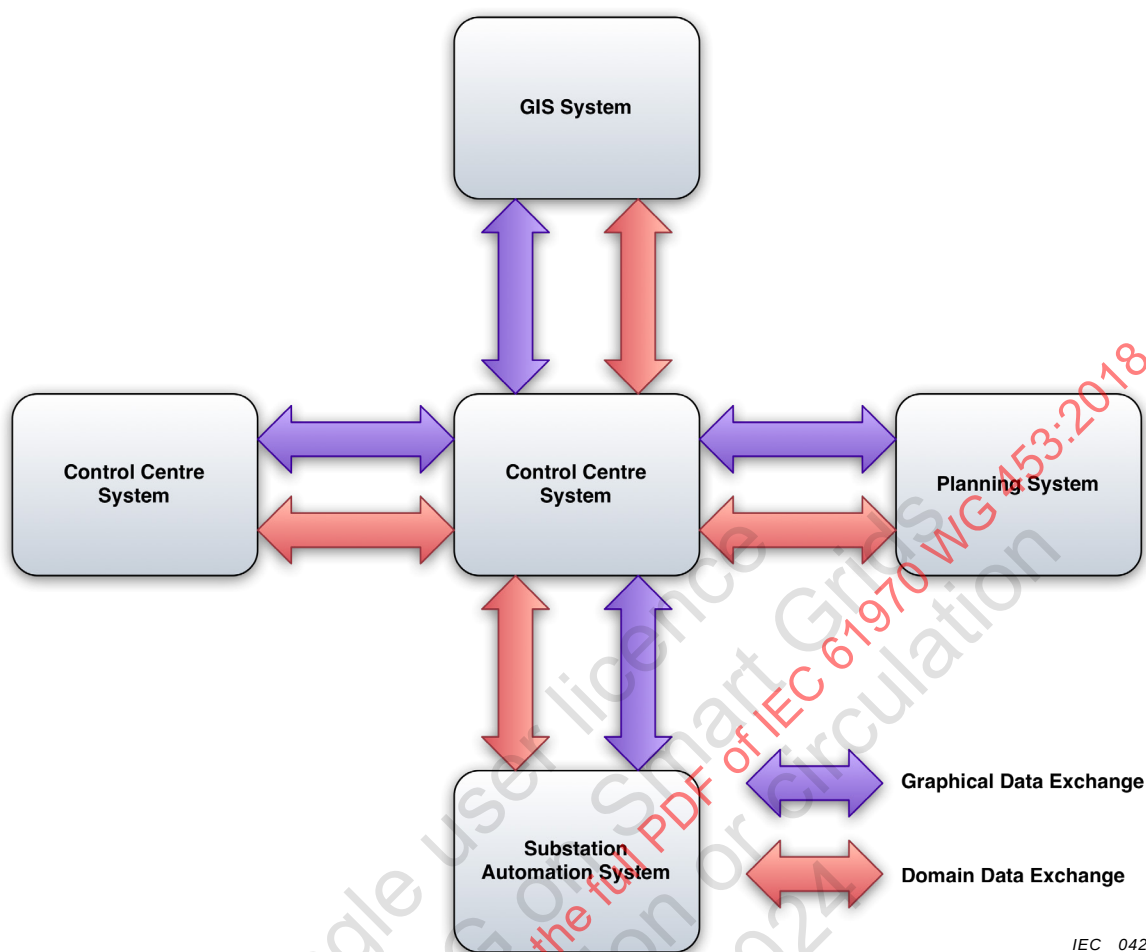
Note 1 to entry: Typically, the diagram object style is resolved in a very specific way for each system.

4 Use Cases

4.1 General use cases for diagram exchange

Figure 1 shows a high-level view of using diagram layout data exchange with potential systems that can make use of the diagram layout data.

⁸ Footnote 8 applies to the French version only.



IEC 0423/14

Figure 1 – System overview

An examination of the use cases for such exchanges revealed that the diagrams being exchanged are not fixed graphic presentations. Instead the diagrams vary considerably in appearance as attributes of CIM objects change, and they support important user interaction. For example, a SCADA one-line in its creator's system is supporting live control of the power system, while in a receiver's system it is merely used for reference so that a neighbour can understand what the system looks like beyond its own border. These variations are typically not easy or appropriate to map between the source and receiving systems because of the degree of difference in the way those systems are designed.

In the specification of diagrams, one common element is that the style of displaying an object of a certain kind is usually defined once and then re-used, but the placement of objects must be adjusted whenever new elements are added to the data model. This placement and maintenance of placement is where most of the labour investment takes place that users would like to preserve. As a result, this standard is limited to the exchange of diagram layouts (meaning the arrangement of CIM objects in a display space) rather than a complete exchange of all characteristics of a graphic presentation.

With this proposed standard, instead of maintaining duplicate schematics for different applications, the schematics are exported by one system and imported by the other system. Diagram layout profile is an extension to the CIM power system model exchange IEC 61970-452, and will be orchestrated along with the existing CIM XML model exchange and updates provided using the existing CIM XML Incremental file format as defined in IEC 61970-552. IEC 61970-552 also describes how payload headers provide information as to how payloads fit together.

This process can be applied for initial schematics construction as well as for continuous maintenance.

The importing system can create its graphics displays from the imported data, or the diagram layout data can serve as additional documentation and means of understanding for the domain data exchange.

4.2 Simple bay diagram example

Diagrams are constructed using different approaches. This subclause illustrates this using a simple bay drawing. Figure 2 shows a typical representation of a bay.

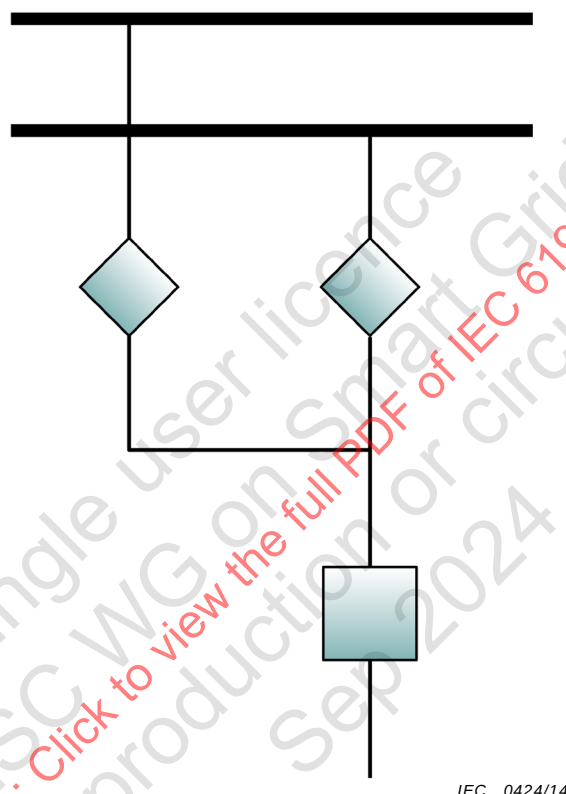


Figure 2 – Bay diagram as rendered

In a GIS system, lines are typically drawn up to the centre point of the symbol. This is shown in Figure 3 (transparent symbol is used for illustration). This construction method is independent of the symbol size, i.e. the drawing looks “pretty” even when the importing system used a smaller symbol size than the original one.

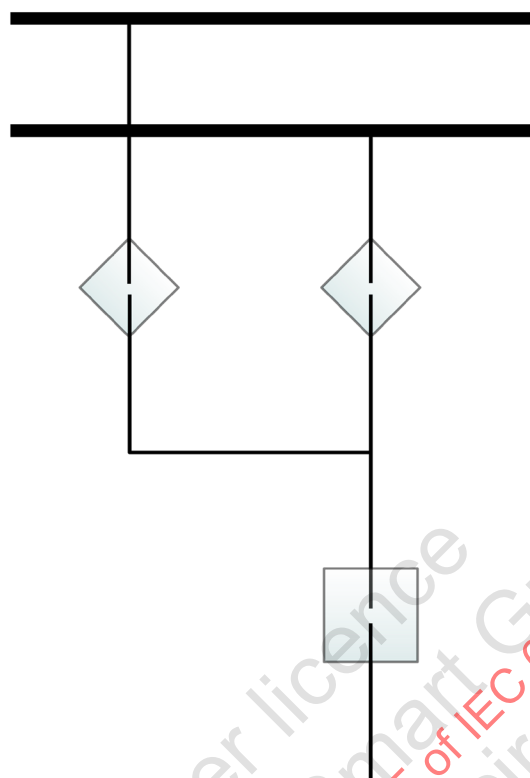
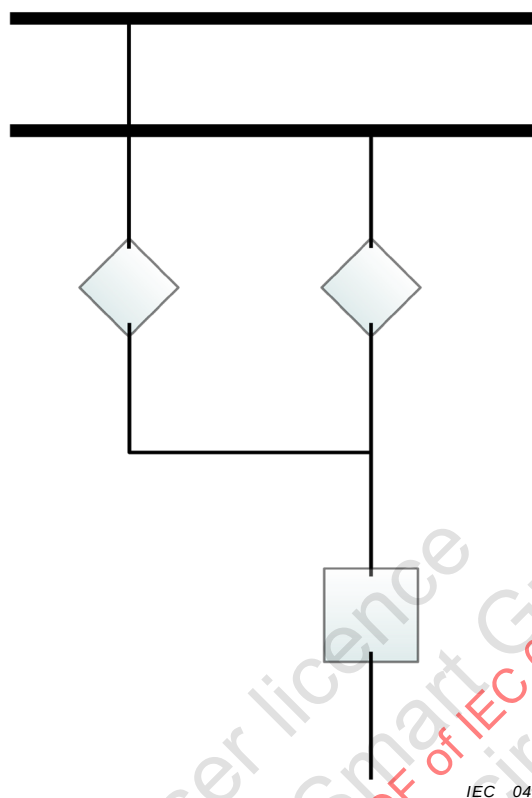


Figure 3 – Bay diagram drawn in GIS style

In a SCADA system, lines are typically drawn to align with the terminals of the object represented by the symbol. This is shown in Figure 4 (transparent symbol is used for illustration). This construction method gives a closer representation of the electrical topology, but depends on a certain symbol size.



IEC 0426/14

Figure 4 – Bay diagram drawn in SCADA style

5 Diagram layout exchange profile

5.1 General

Profile namespace: <http://iec.ch/TC57/61970-453/DiagramLayout/3#>

This standard specifies an exchange profile for schematics with the following characteristics.

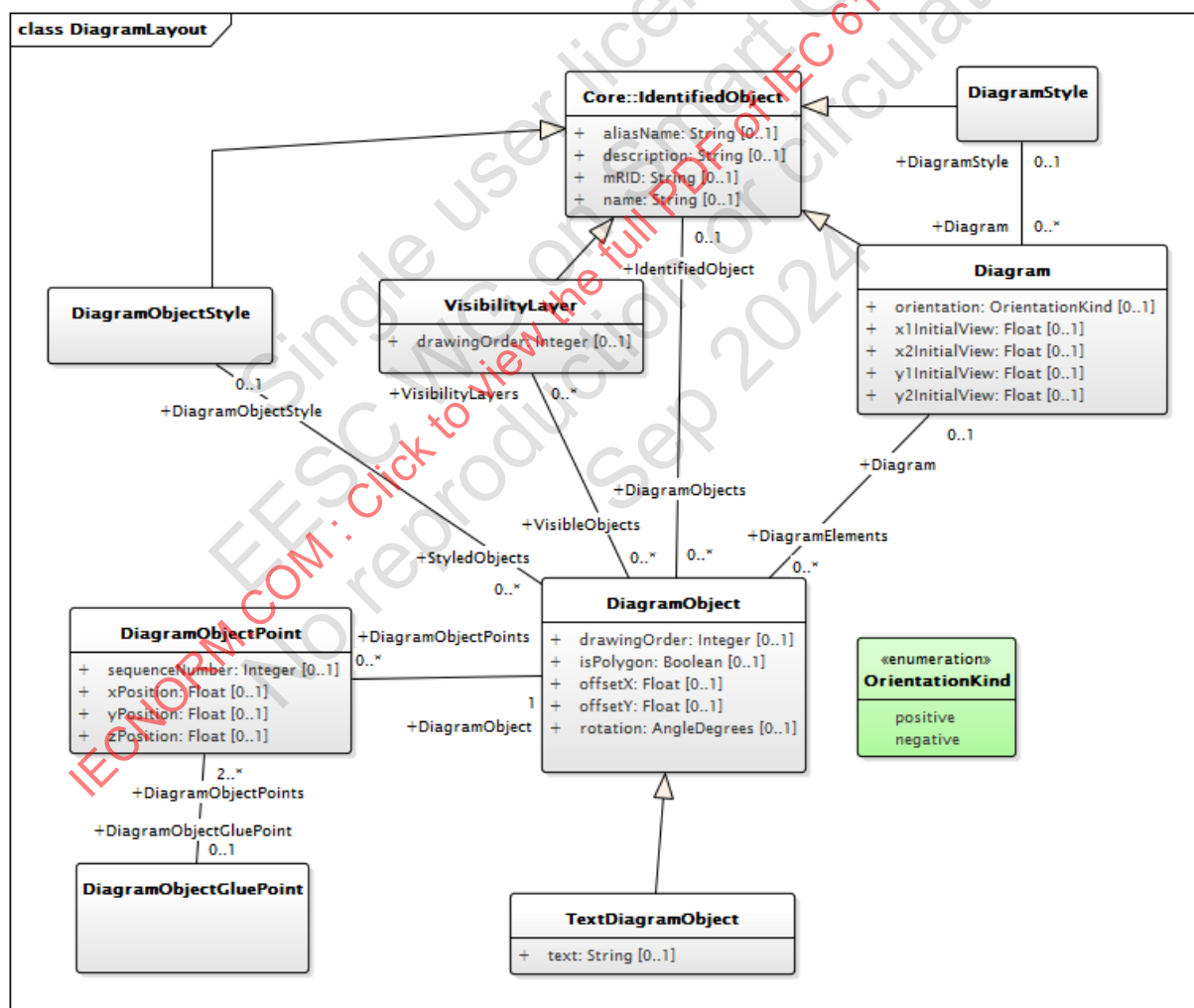
- A generic method for linking the diagram object to the domain data. Domain data and diagram layout data can be exchanged together with the domain data, or separately from each other with the assumption that domain data will have already been imported should diagram layout data be imported separately.
- It does not require or imply a specific domain data format. Therefore, it supports domain data modelled according to the IEC 61970-301 Common information model (CIM) that is exchanged in the IEC 61970-501 format (CIM RDF schema). Diagram layout data references domain data in compliance with IEC/TR 62541-1, *OPC Unified Architecture*.
- The intended usage of this standard is that the source of a diagram will use the standard to encode the layout of their diagram as they created it. It is then incumbent upon any receiver to supply the means of using that diagram layout in their system.
 - In the simplest situation, if the source uses Switch object placements, the receiver creates a generic Switch template in the receiver's system that will be used to render Switches (i.e. there is no exchange of Switch templates).
 - In more complex situations, a 1:1 correspondence may not be desirable, and receivers may have to create generic transformations in order to use exchanges.
 - However, in both situations, it is expected that once an investment is made in a strategy for rendering, repeated imports may be carried out automatically.
- Supports the exchange of diagram objects that have no relationship to domain data, i.e., pure static background objects.

- Supports multiple representations of the same domain object in the same or different diagrams.
- Allows the diagram to be used as the unit of exchange, providing a straightforward approach to partial exchange, or an exchange between systems that have a separate model and repository for diagram layout data.
- Supports assignment of diagram objects to layers or other means, for showing or hiding information based on zoom level and/or user interest.
- A generic method for proprietary extensions to enable full round tripping (export and import back into the same system) without information loss within a system, and without breaking the standard exchange format.

5.2 Diagram layout profile classes

5.2.1 General

The diagram layout classes are a part of IEC 61970-301 and as such are associated to the domain data using the standard UML association relationship. A new sub-package in IEC 61970-301 contains the classes required for expressing schematic data for domain objects as illustrated in Figure 5.



IEC

Figure 5 – Diagram layout information model

5.2.2 Diagram

5.2.2.1 General

The Diagram represents the diagram being exchanged. The initialView attributes can be used to specify an initial view with the x,y coordinates of the diagonal points. The coordinate system is a standard Cartesian coordinate system and the orientation attribute defines a positive or negative orientation. A positive orientation gives standard “right-hand” orientation, with negative orientation indicating a “left-hand”; orientation. For 2D diagrams, a positive orientation will result in X values increasing from left to right and Y values increasing from bottom to top. A negative orientation gives the “left-hand”; orientation (favoured by computer graphics displays) with X values increasing from left to right and Y values increasing from top to bottom.

5.2.2.2 Native members

orientation	0..1	OrientationKind	The Diagram represents the diagram being exchanged. The initialView attributes can be used to specify an initial view with the x,y coordinates of the diagonal points. The coordinate system is a standard Cartesian coordinate system and the orientation attribute defines a positive or negative orientation.
x1InitialView	0..1	Float	X coordinate of the first corner of the initial view
x2InitialView	0..1	Float	X coordinate of the second corner of the initial view
y1InitialView	0..1	Float	Y coordinate of the first corner of the initial view
y2InitialView	0..1	Float	Y coordinate of the second corner of the initial view

5.2.2.3 Inherited members

mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject
aliasName	0..1	String	see IdentifiedObject
description	0..1	String	see IdentifiedObject

5.2.3 DiagramObject

5.2.3.1 General

The DiagramObject class defines an object that defines one or more points in a given space. This object can be associated with anything that subclasses *IdentifiedObject* in IEC 61970-301.

5.2.3.2 Native members

drawingOrder	0..1	Integer	The drawing order of this element. The higher the number, the later the element is drawn in sequence. This is used to ensure that elements that overlap are rendered in the correct order.
isPolygon	0..1	Boolean	Defines whether or not the diagram objects points define the boundaries of a polygon or the routing of a polyline. If this value is true then a receiving application should consider the first and last points to be connected.
offsetX	0..1	Float	The offset in the X direction. This is used for defining the offset from centre for rendering an icon (the default is that a single point specifies the centre of the icon). The offset is in per-unit with 0 indicating there is no offset from the horizontal centre of the icon. -0,5 indicates it is offset by 50 % to the left and 0,5 indicates an offset of 50 % to the right.
offsetY	0..1	Float	The offset in the Y direction. This is used for defining the offset from centre for rendering an icon (the default is that a single point specifies the centre of the icon). The offset is in per-unit with 0 indicating there is no offset from the vertical centre of the icon. The offset direction is dependent on the orientation of the diagram, with -0,5 and 0,5 indicating an offset of ± 50 % on the vertical axis.
rotation	0..1	AngleDegrees	Sets the angle of rotation (in Degrees) of the diagram object in a clockwise direction from the normal
Diagram	1..1	Diagram	A diagram object is part of a Diagram
DiagramObjectPoints	0..unbounded	DiagramObjectPoint	A diagram object can have 0 or more points to reflect its layout position, routing (for polylines) or boundary (for polygons)
DiagramObjectStyle	0..1	DiagramObjectStyle	A diagram object has a style associated that provides a reference for the style used in the originating system
IdentifiedObject	0..1	IdentifiedObject	The domain object that this diagram object is associated with
DiagramStyle	0..1	DiagramStyle	A Diagram may have a DiagramStyle.

5.2.3.3 Inherited members

mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject
aliasName	0..1	String	see IdentifiedObject
description	0..1	String	see IdentifiedObject

5.2.4 DiagramObjectGluePoint

This is used for grouping DiagramObjectPoints from different DiagramObjects that are considered to be “glued” together in a diagram even if they are not at the exact same coordinates.

5.2.5 DiagramObjectPoint

5.2.5.1 General

A DiagramObjectPoint is a point in a given space defined by 3 coordinates and associated to a DiagramObject. The coordinates may be positive or negative as the origin does not have to be in the corner of a diagram. The sequence attribute is used when a DiagramObject has more than one DiagramObjectPoint in which case this defines the drawing order. A DiagramObject may represent any CIM object. For single line diagrams such objects can include:

- Analogue measurement values
- Breaker/disconnector
- Power transformer
- Transmission line

5.2.5.2 Native members

sequenceNumber	0..1	Integer	The sequence position of the point, used for defining the order of points for DiagramObjects acting as a polyline or polygon with more than one point
xPosition	1..1	Float	The X coordinate of this point
yPosition	1..1	Float	The Y coordinate of this point
zPosition	0..1	Float	The Z coordinate of this point
DiagramObject	1..1	DiagramObject	The diagram object with which the points are associated
DiagramObjectGluePoint	0..1	DiagramObjectGluePoint	The 'glue' point that this point is associated with

5.2.6 DiagramObjectStyle

5.2.6.1 General

DiagramObjectStyle is a reference to a style used by the originating system for a DiagramObject. A DiagramObjectStyle describes information such as:

- line thickness
- shape, e.g circle, rectangle ...
- colour

5.2.6.2 Inherited members

mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject
aliasName	0..1	String	see IdentifiedObject
description	0..1	String	see IdentifiedObject

5.2.7 TextDiagramObject

5.2.7.1 General

A TextDiagramObject is a diagram object for placing free-text or text derived from an associated domain object.

5.2.7.2 Native members

text	0..1	String	The text that is displayed by this text diagram object
------	------	--------	--

5.2.7.3 Inherited members

drawingOrder	0..1	Integer	see DiagramObject
isPolygon	0..1	Boolean	see DiagramObject
offsetX	0..1	Float	see DiagramObject
offsetY	0..1	Float	see DiagramObject
rotation	0..1	AngleDegrees	see DiagramObject
Diagram	1..1	Diagram	see DiagramObject
DiagramObjectPoints	0..unbounded	DiagramObjectPoint	see DiagramObject
DiagramObjectStyle	0..1	DiagramObjectStyle	see DiagramObject
IdentifiedObject	0..1	IdentifiedObject	see DiagramObject
mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject
aliasName	0..1	String	see IdentifiedObject
description	0..1	String	see IdentifiedObject

5.2.8 VisibilityLayer

5.2.8.1 General

Layers are typically used for grouping diagram objects according to themes and scales. Themes are used to display or hide certain information (e.g., lakes, borders), while scales are used for hiding or displaying information depending on the current zoom level (hide text when it is too small to be read, or when it exceeds the screen size). This is also called de-cluttering.

The diagram layout profile will support an m:n relationship between diagram objects and layers. It will be the task of the importing system to convert an m:n case into an appropriate 1:n representation if the importing system does not support m:n.

5.2.8.2 Native members

drawingOrder	0..1	Integer	The drawing order for this layer. As with the drawingOrder for diagram objects, the higher the number, the later the layer and the objects within it are rendered.
VisibleObjects	1..unbounded	DiagramObject	A visibility layer can contain one or more diagram objects

5.2.8.3 Inherited members

mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject
aliasName	0..1	String	see IdentifiedObject
description	0..1	String	see IdentifiedObject

5.2.9 Abstract classes – IdentifiedObject

5.2.9.1 General

An IdentifiedObject is a root class to provide common identification for all classes needing identification and naming attributes.

5.2.9.2 Native members

name	mult	type	description
aliasName	0..1	String	The aliasName is free text human readable name of the object alternative to IdentifiedObject.name. It may be non unique and may not correlate to a naming hierarchy. The attribute aliasName is retained because of backwards compatibility between CIM releases. It is however recommended to replace aliasName with the Name class as aliasName is planned for retirement at a future time.
description	0..1	String	The description is a free human readable text describing or naming the object. It may be non unique and may not correlate to a naming hierarchy.
mRID	0..1	String	Master resource identifier issued by a model authority. The mRID is globally unique within an exchange context. Global uniqueness is easily achieved by using a UUID, as specified in RFC 4122, for the mRID. The use of UUID is strongly recommended. For CIMXML data files in RDF syntax conforming to IEC 61970-552 Edition 1, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements.
name	0..1	String	The name is any free human readable and possibly non unique text naming the object.

5.2.10 Enumerations – OrientationKind

negative
positive

5.2.11 DiagramStyle

5.2.11.1 General

The diagram style refers to a style used by the originating system for a diagram. A diagram style describes information such as schematic, geographic, bus-branch etc.

5.2.11.2 Native members

Diagram	0..*	Diagram	A DiagramStyle can be used by many Diagrams.
---------	------	---------	--

5.2.11.3 Inherited members

aliasName	0..1	String	see IdentifiedObject
description	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject

6 Graphical rendering

6.1 General

Section 6 illustrates the use of the object model described in section 5. It uses the conventions shown in Figure 6 for representing object instances and relationships.

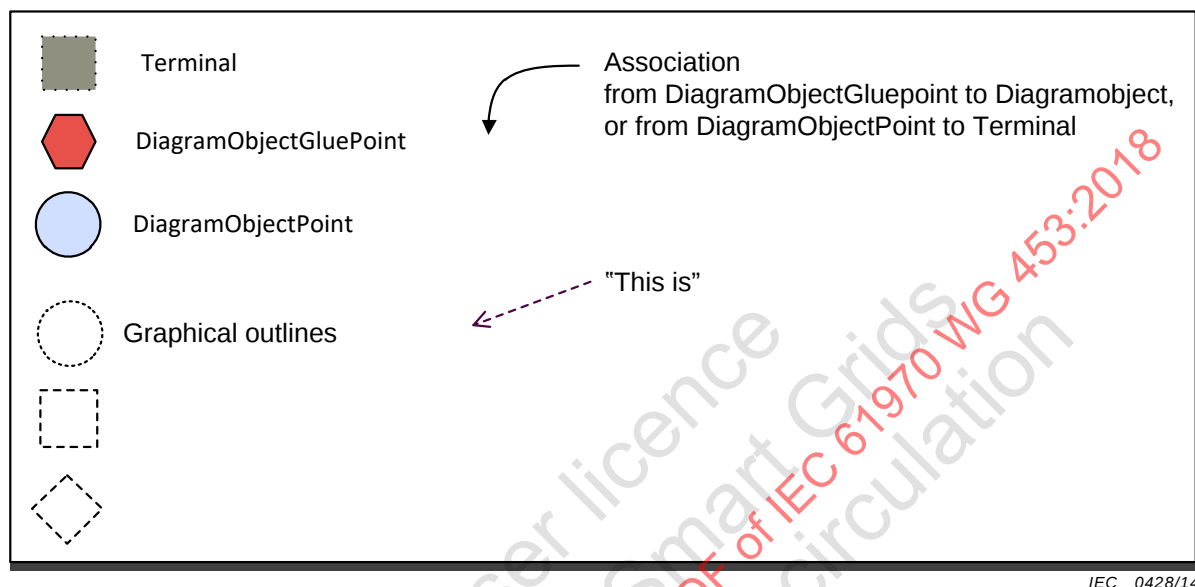
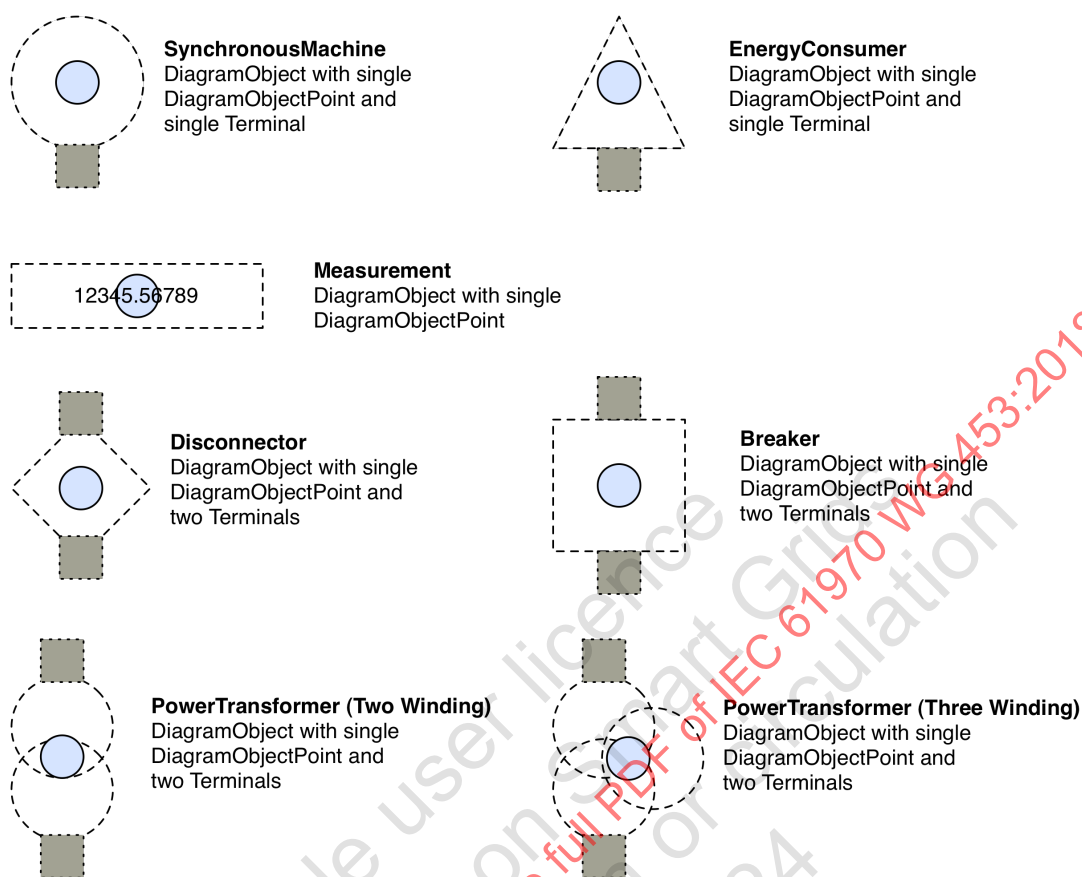


Figure 6 – Conventions used for representing object instances and associations

6.2 Single point objects

The simplest Diagram Objects have a single point in the form of a DiagramObjectPoint. These are used for specifying the centre-point of an icon, which can be pieces of network equipment or measurement locations on a diagram. This is shown in Figure 7.



IEC 0429/14

Figure 7 – Single point diagram objects

6.3 Multiple point objects

For those diagram objects that cannot be accurately rendered using a single point, multiple DiagramObjectPoints are used within a single DiagramObject. Terminals themselves can be given multiple points so that the interconnection of equipment can be displayed correctly as shown in Figure 8.

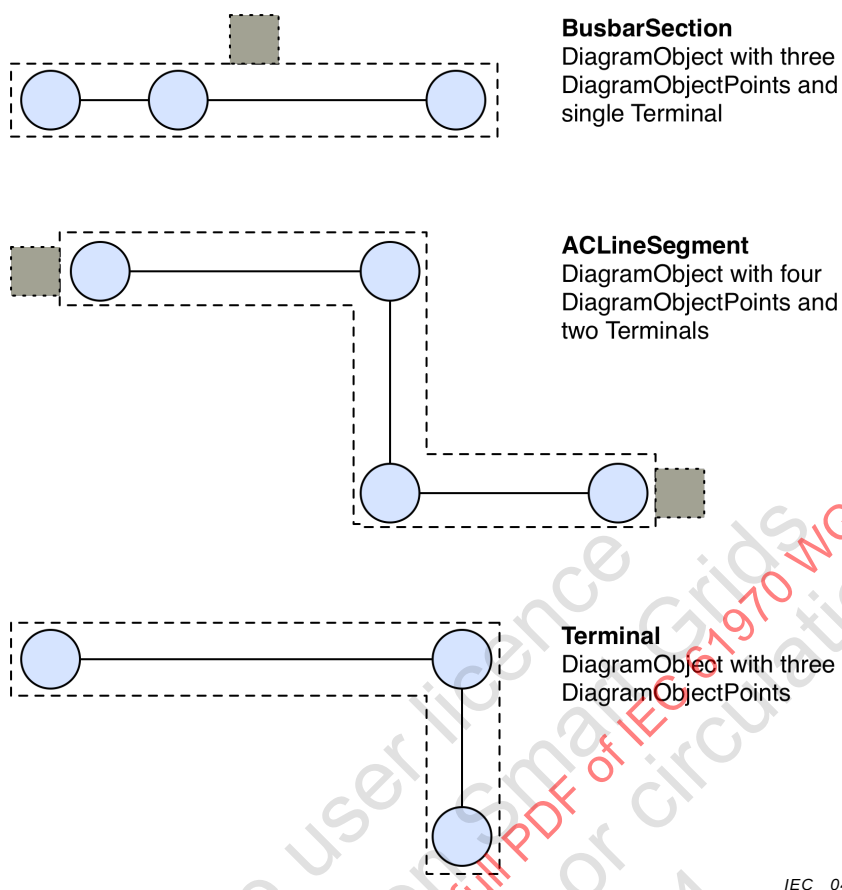


Figure 8 – Multiple point diagram objects

6.4 Gluing points

DiagramObjectGluePoints are used to identify when two or more points on different DiagramObjects are considered to be 'glued' together on a diagram. This information is required so that a receiver can identify the case of multiple, synchronised points, which may not be at the same coordinates and thus ensure that modifications to one are reflected in the other points. An example shown in Figure 9 has a Disconnecter with two Terminals and three DiagramObjectPoints, all part of different DiagramObjects since each Terminal and the Disconnecter are separate objects in IEC 61970-301.

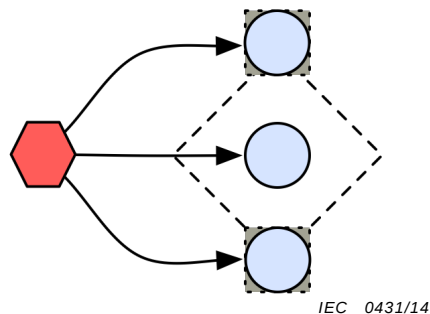
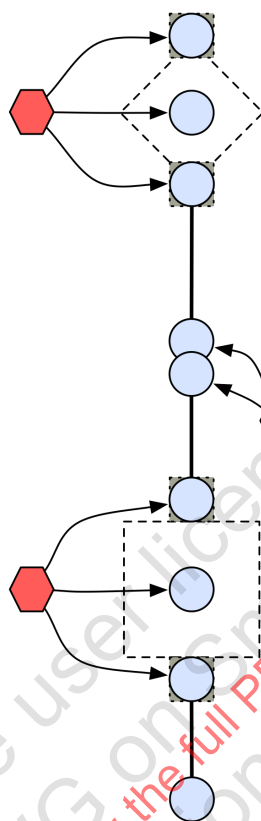


Figure 9 – Disconnecter with glue point

The glue point here is used to identify that the three DiagramObjectPoints are related to each other even though they are at different coordinates.

In the electrical topology model defined in IEC 61970-301 there is a ConnectivityNode object at this point. However its inclusion here would be superfluous as the DiagramObjectGluePoint provides a mechanism for identifying that these DiagramObjectPoints are connected, as illustrated in Figure 10.



IEC 0432/14

Figure 10 – Disconnect and breaker with glue points

6.5 Diagram object style

The diagram object style defines how to convert the state of a domain object into a visible representation, using information such as:

- line thickness
- shape, e.g circle, rectangle
- colour

The definition of the styles and their use for graphical rendering are typically solved in very specific ways for each system and are not part of this standard. The diagram layout transfers only make references to the diagram object style, not the definition of the diagram object style itself. The importing system will have to map the imported references to its best matching local diagram object style. This shall be by an agreement between the sending and receiving systems, which is outside of this standard's scope.

As an example, if the importing system can derive that a domain object is representing a remote controlled pole mounted load break switch with alarm indication, this might be enough to establish a sufficiently identical visualisation in the importing system.

6.6 Diagram layout exchange process

Diagram layout exchanges allow the exchanging of diagram layout data independent of and separate from domain data. Using the approach of separating data into profiles for specific

types of data exchanges, the diagram layout profile detailed here can be utilised along with other profiles defined in other IEC standards.

IEC 61970-452 defines profiles for the exchange of power system models with an equipment profile defining electrical equipment and connectivity. IEC 61970-456 defines profiles for measurements, state variables and topology data that are dependent on the equipment profile defined in IEC 61970-452. This standard uses the same approach with the diagram layout profile utilising the same Equipment profile.

Figure 11 gives an overview of these profiles:

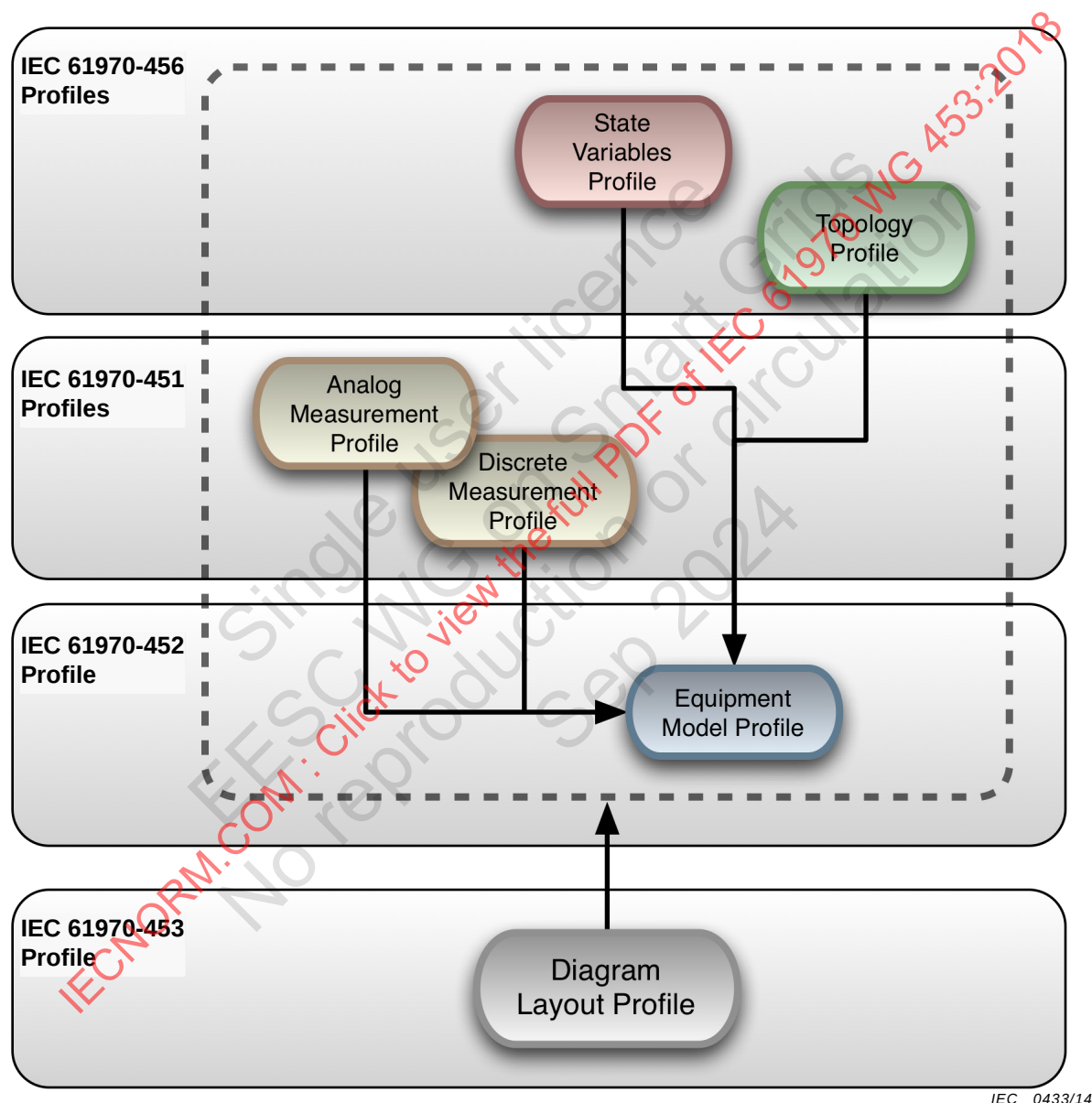


Figure 11 – Profiles within IEC standards

When importing diagram layout data separate from domain (i.e. Equipment) data, the following rules apply.

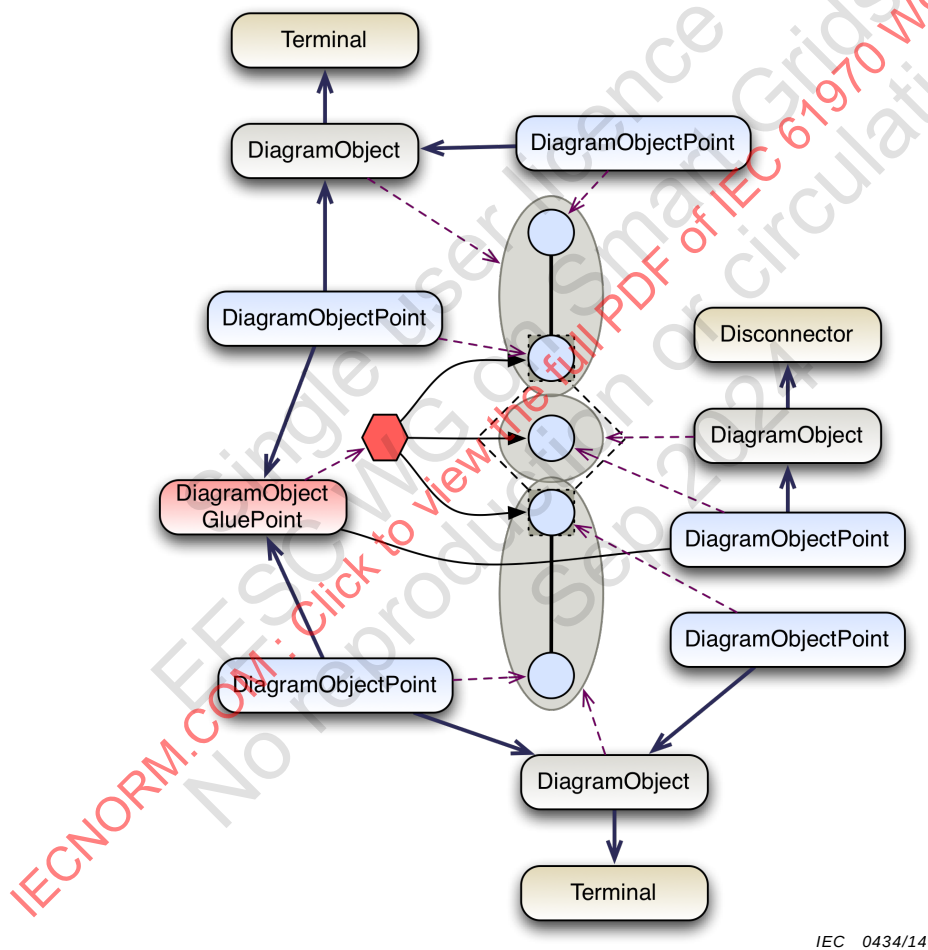
- The domain data import has to take place before the diagram layout import, otherwise it is not possible to resolve references from diagram objects to domain objects.
- It is the task of the importing program to reasonably report and handle inconsistencies between diagram objects and domain data, for example missing domain data.

For instance data exchange using CIM XML encoding as defined in IEC 61970-552, the header information will define the unique identifier of the data set on which the schematic layout data depends, thus allowing an importing system to determine whether it can successfully process the data.

7 Examples

7.1 Data instantiation and encoding

Taking a single disconnector with two terminals the diagrammatical view consists of eleven objects using five classes from the UML: the Disconnecter, two Terminals, three DiagramObjects, five DiagramObjectPoints and a single DiagramObjectGluePoint. This is illustrated in Figure 12. This diagram shows the objects created, the associations between them and their corresponding position in the diagrammatical view used in the previous figures.



IEC 0434/14

Figure 12 – Disconnecter diagram object instantiation

When encoded in the format defined by IEC 61970-552 the resulting XML is shown in Figure 13:

```
<cim:DiagramObjectGluePoint rdf:ID="gp0">
</cim:DiagramObjectGluePoint>

<cim:Disconnecter rdf:ID="discon0">
  <cim:IdentifiedObject.name>Disconnecter</cim:IdentifiedObject.name>
</cim:Disconnecter>
<cim:DiagramObject rdf:ID="disconDO">
  <cim:DiagramObject.IdentifiedObject rdf:resource="#discon0"/>
</cim:DiagramObject>
<cim:DiagramObjectPoint rdf:ID="disconDOP0">
  <cim:DiagramObjectPoint.xPosition>10</cim:DiagramObjectPoint.xPosition>
  <cim:DiagramObjectPoint.yPosition>10</cim:DiagramObjectPoint.yPosition>
  <cim:DiagramObjectPoint.DiagramObject rdf:resource="#disconDO"/>
  <cim:DiagramObjectPoint.GluePoint rdf:resource="#gp0"/>
</cim:DiagramObjectPoint>

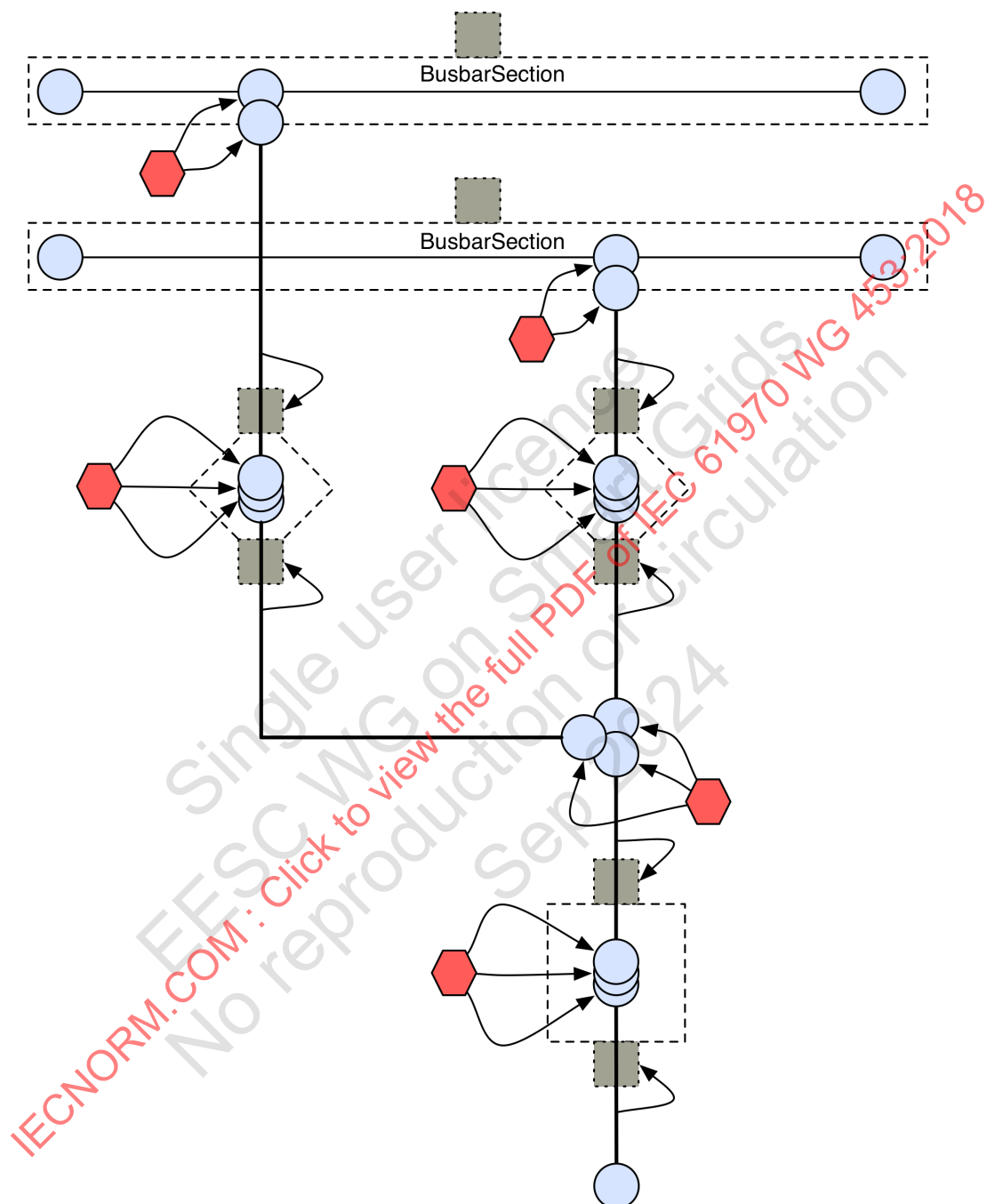
<cim:Terminal rdf:ID="t0">
  <cim:IdentifiedObject.name>Terminal 0</cim:IdentifiedObject.name>
  <cim:Terminal.ConductingEquipment rdf:resource="#discon0"/>
</cim:Terminal>
<cim:DiagramObject rdf:ID="t0DO">
  <cim:DiagramObject.IdentifiedObject rdf:resource="#t0"/>
</cim:DiagramObject>
<cim:DiagramObjectPoint rdf:ID="t0DOP0">
  <cim:DiagramObjectPoint.xPosition>10</cim:DiagramObjectPoint.xPosition>
  <cim:DiagramObjectPoint.yPosition>8</cim:DiagramObjectPoint.yPosition>
  <cim:DiagramObjectPoint.DiagramObject rdf:resource="#t0DO"/>
  <cim:DiagramObjectPoint.DiagramObjectGluePoint rdf:resource="#gp0"/>
</cim:DiagramObjectPoint>
<cim:DiagramObjectPoint rdf:ID="t0DOP1">
  <cim:DiagramObjectPoint.xPosition>10</cim:DiagramObjectPoint.xPosition>
  <cim:DiagramObjectPoint.yPosition>6</cim:DiagramObjectPoint.yPosition>
  <cim:DiagramObjectPoint.DiagramObject rdf:resource="#t0DO"/>
</cim:DiagramObjectPoint>

<cim:Terminal rdf:ID="t1">
  <cim:IdentifiedObject.name>Terminal 0</cim:IdentifiedObject.name>
  <cim:Terminal.ConductingEquipment rdf:resource="#discon0"/>
</cim:Terminal>
<cim:DiagramObject rdf:ID="t1DO">
  <cim:DiagramObject.IdentifiedObject rdf:resource="#t1"/>
</cim:DiagramObject>
<cim:DiagramObjectPoint rdf:ID="t1DOP0">
  <cim:DiagramObjectPoint.xPosition>10</cim:DiagramObjectPoint.xPosition>
  <cim:DiagramObjectPoint.yPosition>12</cim:DiagramObjectPoint.yPosition>
  <cim:DiagramObjectPoint.DiagramObject rdf:resource="#t1DO"/>
  <cim:DiagramObjectPoint.DiagramObjectGluePoint rdf:resource="#gp0"/>
</cim:DiagramObjectPoint>
<cim:DiagramObjectPoint rdf:ID="t1DOP1">
  <cim:DiagramObjectPoint.xPosition>10</cim:DiagramObjectPoint.xPosition>
  <cim:DiagramObjectPoint.yPosition>16</cim:DiagramObjectPoint.yPosition>
  <cim:DiagramObjectPoint.DiagramObject rdf:resource="#t1DO"/>
</cim:DiagramObjectPoint>
```

Figure 13 – IEC 61970-552 Encoding for disconnecter diagram data

7.2 Simple bay example use case

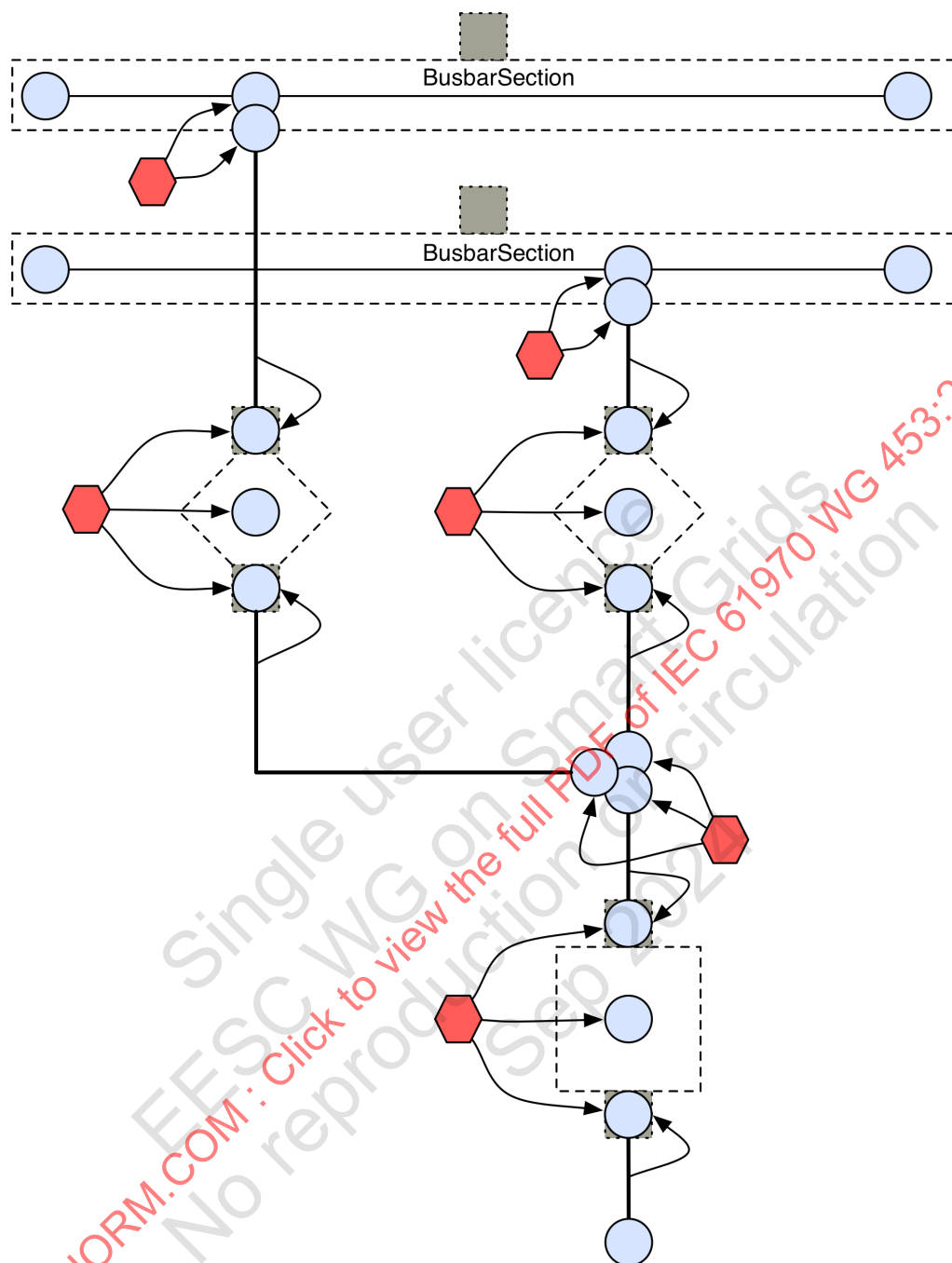
Figure 14 illustrates how to apply the exchange model to the GIS construction type simple bay example use case (lines drawn to the centre point).



IEC 0435/14

Figure 14 – Bay diagram example with objects outlined GIS style

Figure 15 illustrates how to apply the exchange model to the SCADA construction type simple bay example use case (lines drawn to the terminals).



IEC 0436/14

Figure 15 – Bay diagram example with objects outlined SCADA/EMS style

Annex A (informative)

Benefits and format conversion from IEC 61970-453 Edition 1 to Edition 2

This edition includes the following significant technical changes with respect to the previous edition:

- a) The SVG elements and its data model have been replaced by the Diagram Layout Package, which is now an integral part of the IEC 61970-301 (CIM) model.
- b) The exchange is in accordance with and is a part of the IEC 61970 profile concept.
- c) A glue point object has been introduced to model explicit connections between graphics elements.

Benefits resulting from the changes made to Edition 2:

- Clear separation between layout and rendering eliminates ambiguities in the use of SVG and style features in Edition 1.
- The same encoding can be used as for other IEC 61970-301 packages, enabling the exchange either as a separate document, or combined with data from other IEC 61970-301 packages.
- The glue point allows the resolution of differences in rendering between the sending and the receiving system, and it supports editing in the receiving system without losing connections between graphic elements.

The basic concept of separating layout, rendering and domain data has not been changed. It should be possible to convert the format from Edition 1 to Edition 2 to a large extent by using transformation rules. A summary of the changes is shown in Table A.1.

Table A.1 – Summary of changes between Edition 1 and Edition 2

Feature	Edition 1	Edition 2
Schematic diagram	CgoDiagram (SVG element)	Diagram
Graphical Object within a diagram	CgoGraphicalObject, CgoSymbol, CgoBasicShape (SVG elements)	DiagramObject, TextDiagramObject, DiagramObjectPoint
Element grouping	CgoGroup	<not supported>
References	Metadata (SVG concept)	Explicit relationship model
Rendering reference	Metadata PresentationLogic_Ref	DiagramObjectStyle
Layer membership	Metadata Layer_Ref	VisibilityLayer
Domain data reference	Metadata PSR_Ref, Meas_Ref	IdentifiedObject
User Interaction	Metadata ComplexObject	<not supported>
Predefined views	CgoDiagramView	<not supported>
Graphic connection	<not supported>	DiagramObjectGluePoint

Bibliography

IEC 61968-11, *Application integration at electric utilities – System interfaces for distribution management – Part 11: Common information model (CIM) extensions for distribution*

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

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

IEC 61970-452, *Energy management system application program interface (EMS-API) – Part 452: CIM Static transmission network model profiles*

Single user licence
EEESC WG on Smart Grids
IECNORM.COM : Click to view the full PDF of IEC 61970 WG 453:2018
No reproduction or circulation
Sep 2024

SOMMAIRE

AVANT-PROPOS	32
INTRODUCTION	34
1 Domaine d'application	35
2 Références normatives	35
3 Termes et définitions	35
4 Cas d'utilisation	36
4.1 Cas d'utilisation généraux pour échange de diagrammes	36
4.2 Exemple de diagramme de Bay simple	38
5 Profil d'échange de disposition du diagramme	40
5.1 Généralités	40
5.2 Classes de profils de disposition du diagramme	41
5.2.1 Généralités	41
5.2.2 Diagramme	42
5.2.3 DiagramObject	43
5.2.4 DiagramObjectGluePoint	44
5.2.5 DiagramObjectPoint	45
5.2.6 DiagramObjectStyle	45
5.2.7 TextDiagramObject	45
5.2.8 VisibilityLayer	46
5.2.9 Classes abstraites - IdentifiedObject	47
5.2.10 Énumérations – OrientationKind	47
6 Rendu graphique	48
6.1 Généralités	48
6.2 Objets à point unique	48
6.3 Objets à points multiples	50
6.4 Points d'adhérence	50
6.5 Style d'objet de diagramme	51
6.6 Processus d'échange de disposition du diagramme	52
7 Exemples	54
7.1 Instanciation et codage de données	54
7.2 Cas d'utilisation d'un exemple de Bay simple	56
Annexe A (informative) Avantages et conversion de format de l'Edition 1 de l'IEC 61970-453 à l'Edition 2	58
Bibliographie	59
Figure 1 – Vue d'ensemble du système	37
Figure 2 – Diagramme de Bay tel que rendu	38
Figure 3 – Diagramme de Bay dessiné dans un style GIS	39
Figure 4 – Diagramme de Bay dessiné dans un style SCADA	40
Figure 5 – Modèle d'informations de disposition du diagramme	42
Figure 6 – Conventions utilisées pour représenter des instances d'objet et les associations	48
Figure 7 – Objets de diagramme à point unique	49

Figure 8 – Objet de diagramme à points multiples.....	50
Figure 9 – Disconnector (Sectionneur) avec un point d'adhérence	51
Figure 10 – Disconnector (Sectionneur) et Breaker (Disjoncteur) avec points d'adhérence.....	51
Figure 11 – Profils dans les normes IEC	53
Figure 12 – Instanciation d'objet diagramme Disconnector.....	54
Figure 13 – Codage IEC 61970-552 pour les données de diagramme Disconnector	55
Figure 14 – Exemple de diagramme de Bay avec des objets définis dans un style GIS	56
Figure 15 – Exemple de diagramme de Bay avec des objets définis dans un style SCADA/EMS.....	57

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

**INTERFACE DE PROGRAMMATION D'APPLICATION
POUR SYSTÈME DE GESTION D'ÉNERGIE (EMS-API) –****Partie 453: Profil de disposition de diagramme****AVANT-PROPOS**

- 1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

Cette version consolidée de la Norme IEC officielle et de son amendement a été préparée pour la commodité de l'utilisateur.

L'IEC 61970-453 édition 2.1 contient la seconde édition (2014-02) [documents 57/1409/FDIS et 57/1430/RVD] et son amendement 1 (2018-11) [documents 57/2038/FDIS et 57/2054/RVD].

Cette version Finale ne montre pas les modifications apportées au contenu technique par l'amendement 1. Une version Redline montrant toutes les modifications est disponible dans cette publication.

La Norme internationale IEC 61970-453 a été établie par le comité d'études 57 de l'IEC: Gestion des systèmes de puissance et échanges d'informations associés.

Cette édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente:

- a) Les éléments SVG et leur modèle de données ont été remplacés par le paquetage DiagramLayout (Disposition du diagramme), qui fait maintenant partie intégrante du modèle de l'IEC 61970-301 (CIM).
- b) L'échange est conforme au concept de profil de l'IEC 61970 et en fait partie.
- c) Un objet point d'adhérence ("Glue point") a été introduit pour modéliser des connexions explicites entre des éléments graphiques.

Cette publication a été rédigée selon les Directives ISO/IEC, Partie 2.

Une liste de toutes les parties de la série IEC 61970, publiées sous le titre général *Interface de programmation d'application pour système de gestion d'énergie (EMS-API)*, peut être consultée sur le site web de l'IEC.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

INTRODUCTION

La présente norme fait partie de la série de normes IEC 61970 qui définissent une interface de programmation d'application (API¹) pour un système de gestion d'énergie (EMS²).

La série de normes IEC 61970-3x spécifie un Modèle d'Information Commun (CIM³): une vue logique des aspects physiques relatifs aux informations EMS. La série de normes IEC 61970-3x inclut l'IEC 61970-301, *Base de modèle d'information commun (CIM)*.

La présente norme est l'une de la série de normes de l'IEC 61970-4x qui définissent les spécifications d'interface de composants (CIS⁴) pour un centre de conduite d'entreprise de distribution ou transport d'électricité. L'IEC 61970-4x spécifie les exigences fonctionnelles pour les interfaces qu'un composant (ou une application) doit mettre en œuvre pour échanger des informations avec d'autres composants (ou d'autres applications) et/ou pour avoir accès aux données publiques disponibles par une voie normalisée. Les interfaces des composants décrivent le contenu de messages et les services spécifiques qui peuvent être utilisés dans ce but par les applications. La mise en œuvre de ces messages dans une technologie particulière est décrite dans la série IEC 61970-5x.

Les systèmes de gestion d'énergie utilisent, au niveau de leurs interfaces d'utilisateurs, une diversité de présentations schématiques et quasi géographiques. Elles sont parfois générées automatiquement, mais elles sont plus souvent dessinées manuellement et nécessitent un travail considérable de création et de maintenance. La majeure partie de ce travail est consacrée à l'agencement, ou à la 'disposition' des éléments du système de puissance à l'intérieur du diagramme général. Lorsque des modèles de réseau sont échangés, conformément aux normes IEC 61970-452 et IEC 61968-13, il est souhaitable de pouvoir échanger ces dispositions.

L'IEC 61970-453 spécifie les lignes directrices relatives à l'échange d'informations de disposition du diagramme pour les données schématiques qui sont codées selon l'IEC 61970-552.

¹ API = Application program interface.

² EMS = Energy management system.

³ CIM = Common Information Model.

⁴ CIS = Component interface specifications.

INTERFACE DE PROGRAMMATION D'APPLICATION POUR SYSTÈME DE GESTION D'ÉNERGIE (EMS-API) –

Partie 453: Profil de disposition de diagramme

1 Domaine d'application

La présente partie de l'IEC 61970 appartient à la série IEC 61970-450 à 499 qui, considérée dans son ensemble, définit à un niveau abstrait le contenu et les mécanismes d'échange utilisés pour les données émises entre les composants du centre de conduite.

Sont inclus dans la présente partie de l'IEC 61970 les cas d'utilisation généraux pour l'échange de données de disposition du diagramme, et les lignes directrices pour relier les définitions de structure avec des données du CIM. Des lignes directrices pour la gestion des définitions schématiques par le biais de multiples révisions sont également incluses.

2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 60050, *Vocabulaire électrotechnique international*

IEC 61970-301, *Interface de programmation d'application pour système de gestion d'énergie (EMS-API) – Partie 301: Base de modèle d'information commun (CIM)*

IEC 61970-501, *Energy management system application program interface (EMS-API) – Part 501: Common Information Model Resource Description Framework (CIM RDF) schema* (disponible en anglais seulement)

IEC/TR 62541-1, *OPC Unified Architecture – Part 1: Overview and concepts* (disponible en anglais seulement).

3 Termes et définitions

Pour les besoins du présent document, les termes et définitions de l'IEC 60050 ainsi que les suivants s'appliquent.

3.1

objet de domaine⁵

instance d'une classe qui modélise un objet du monde réel⁶ avec une identité unique

Note 1 à l'article: Un objet de domaine hérite d'un *IdentifiedObject* du CIM. Un objet de domaine n'est normalement pas un objet diagramme.

⁵ Objet de domaine = *Domain object*.

⁶ Objet du monde réel = *Real-world object*.

3.2 diagramme⁷

équivalent électronique d'un plan sur papier sans raccord

Note 1 à l'article: Le diagramme est un conteneur identifié pour les objets diagrammes. Des exemples de diagrammes comprennent les schémas de poste, les schémas orthogonaux de réseaux de transport ou de distribution ou les schémas pseudo géographiques. Un diagramme a un espace de coordonnées bien défini.

3.3 objet de diagramme⁸

représentation d'objets de domaine ou d'arrière-plan statique

Note 1 à l'article: Le diagramme est constitué d'objets de diagramme.

Note 2 à l'article: Un exemple d'objets de domaine inclut des "breakers" (disjoncteurs). Un exemple d'objet d'arrière-plan statique inclut les lacs.

3.4 style d'objet de diagramme

définition de la manière de rendre des objets de diagramme en se basant possiblement sur l'état des objets de domaine

Note 1 à l'article: Typiquement, le style d'objet de diagramme est résolu d'une façon très spécifique pour chaque système.

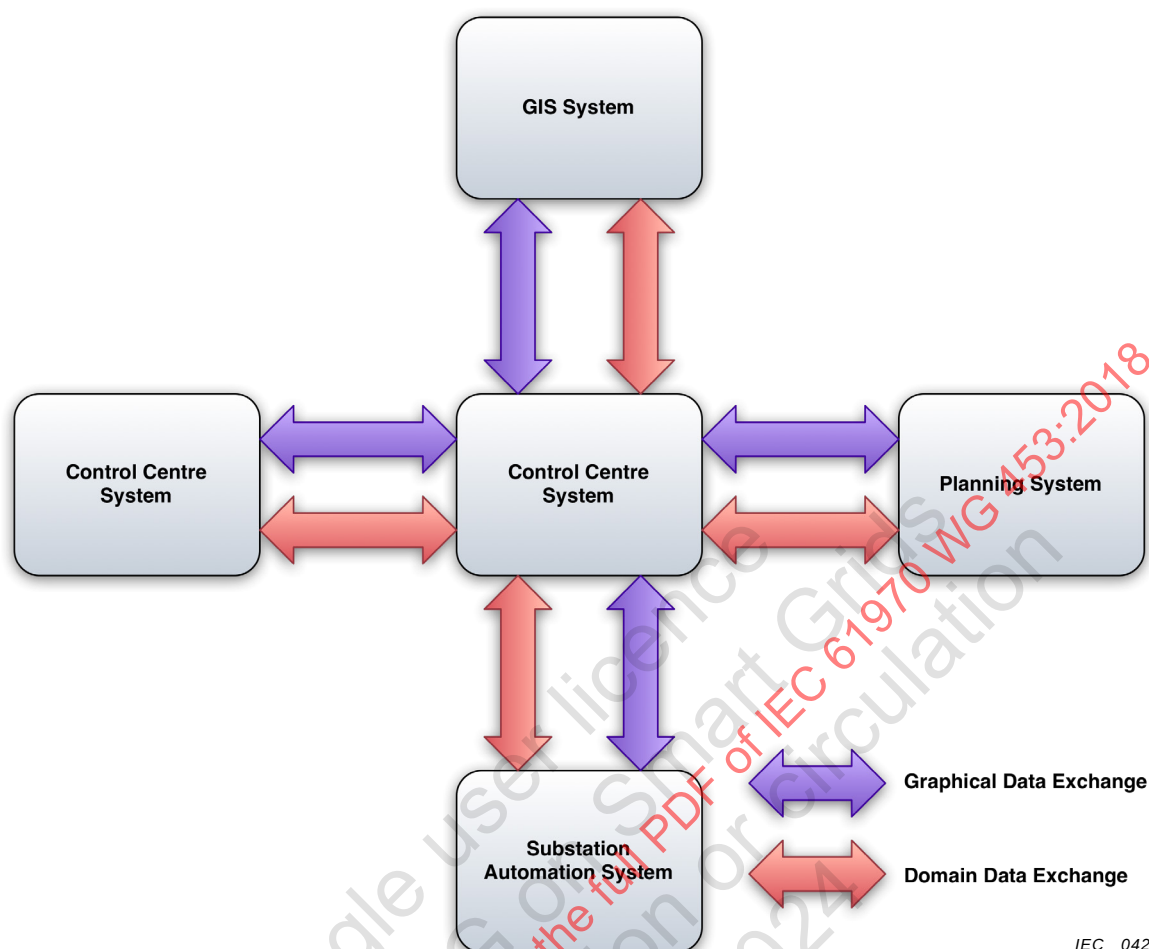
4 Cas d'utilisation

4.1 Cas d'utilisation généraux pour échange de diagrammes

La Figure 1 montre une vue de haut niveau d'utilisation de l'échange de données de disposition du diagramme avec des systèmes potentiels qui peuvent utiliser des données de disposition du diagramme.

⁷ Diagramme = *Diagram*.

⁸ Objet de diagramme = *Diagram object*.



Légende

Anglais	Français
GIS system	Système d'informations géographiques, SIG
Control center system	Système de centre de conduite
Planning system	Système de planification
Substation automation system	Système d'automatisation de postes
Graphical data exchange	Échange de données graphiques
Domain data exchange	Échange de données de domaine

Figure 1 – Vue d'ensemble du système

Un examen des cas d'utilisation de ces échanges a révélé que les diagrammes échangés ne sont pas des présentations graphiques fixes. Au lieu de cela, les diagrammes varient considérablement en apparence quand les attributs des objets CIM changent, et ils prennent en charge une interaction d'utilisateur importante. Par exemple, un schéma unifilaire SCADA (Supervisory Control And Data Acquisition, c'est-à-dire «Système d'acquisition et de contrôle de données») dans son système créateur prend en charge le contrôle temps réel du système de puissance, alors que dans un système récepteur, il est simplement utilisé pour référence afin qu'un voisin puisse comprendre à quoi ressemble le système au-delà de ses propres frontières. Ces variations ne sont généralement pas faciles ou appropriées à mettre en correspondance entre la source et les systèmes de réception en raison du degré de différence de conception de ces systèmes.

Dans la spécification de diagrammes, un élément commun est que le style d'affichage d'un objet d'un certain type est généralement défini une seule fois et ensuite réutilisé, mais le placement d'objets doit être réajusté chaque fois que de nouveaux éléments sont ajoutés au modèle de données. C'est dans ce placement et cette maintenance du placement que se situe la plupart de l'investissement en travail que les utilisateurs voudraient conserver. Par conséquent, la présente norme se limite à l'échange de dispositions de diagrammes (c'est-à-dire la disposition des objets CIM dans un espace d'affichage), au lieu d'un échange complet de toutes les caractéristiques d'une présentation graphique.

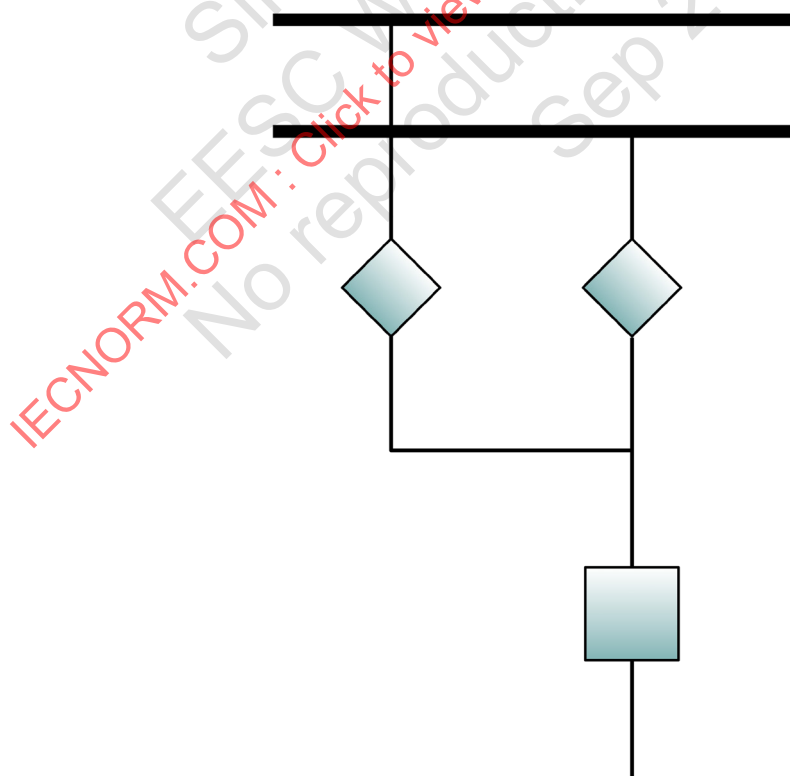
Avec la présente norme proposée, au lieu de maintenir des schémas en doublons pour des applications différentes, les schémas sont exportés par un système et importés par l'autre système. Le profil de disposition du diagramme est une extension de l'IEC 61970-452 relative à l'échange des modèles de système de puissance CIM, et sera orchestré avec l'échange de modèles CIM XML existants et des mises à jour fournies en utilisant le format de fichier CIM XML incrémental existant tel que défini dans l'IEC 61970-552. L'IEC 61970-552 décrit aussi comment les en-têtes de charge utile fournissent des informations et comment les charges utiles s'adaptent entre elles.

Ce processus peut être appliqué pour la construction initiale des schémas ainsi que pour la maintenance continue.

Le système importateur peut créer ses affichages de graphiques à partir des données importées ou bien les données de disposition du diagramme peuvent servir de documentation complémentaire et de moyens de compréhension pour l'échange de données de domaine.

4.2 Exemple de diagramme de Bay simple

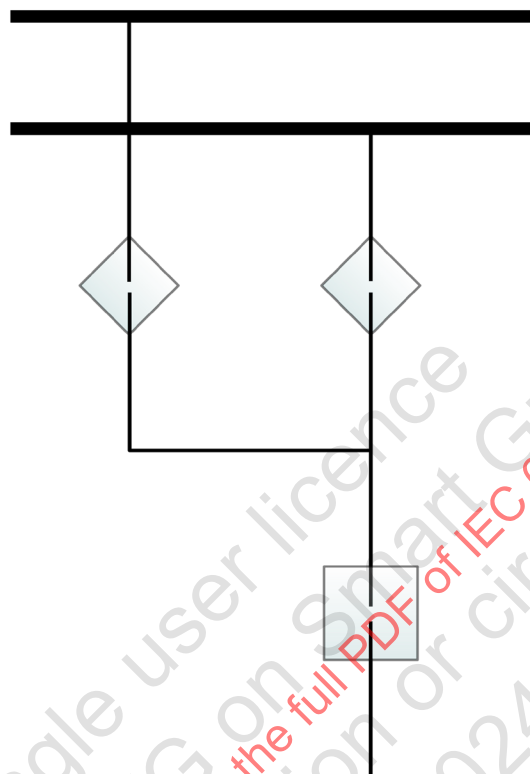
Les diagrammes sont construits selon différentes approches. Le présent paragraphe représente cela en utilisant un simple schéma de bay (c'est-à-dire: cellule). La Figure 2 montre une représentation typique d'une bay.



IEC 0424/14

Figure 2 – Diagramme de Bay tel que rendu

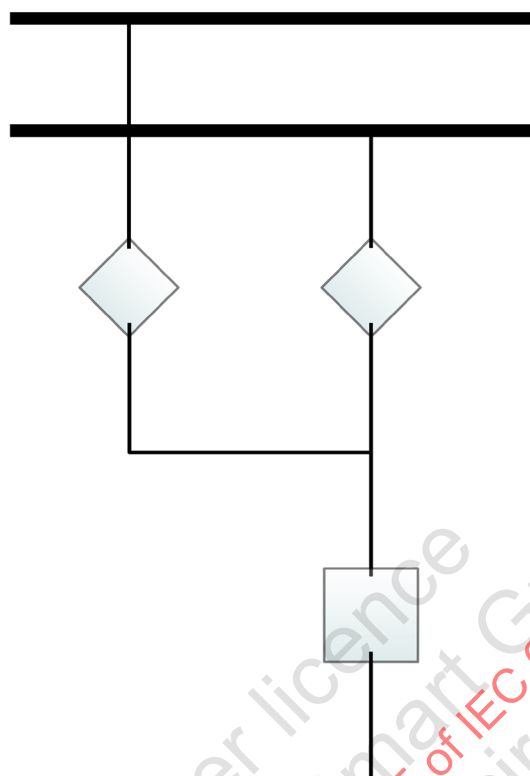
Dans un système GIS, les lignes sont typiquement tracées jusqu'au point central du symbole. Cela est représenté à la Figure 3 (le symbole transparent est utilisé dans un but d'illustration). Cette méthode de construction est indépendante de la taille du symbole, c'est-à-dire que le dessin semble "joli", même si le système importateur a utilisé un symbole de plus petite taille que celui d'origine.



IEC 0425/14

Figure 3 – Diagramme de Bay dessiné dans un style GIS

Dans un système SCADA, les lignes sont typiquement tracées pour s'aligner avec les "Terminals" (bornes) de l'objet représenté par le symbole. Cela est représenté à la Figure 4 (le symbole transparent est utilisé pour le but d'illustration). Cette méthode de construction donne une représentation plus proche de la topologie électrique, mais dépend d'une certaine taille du symbole.



IEC 0426/14

Figure 4 – Diagramme de Bay dessiné dans un style SCADA

5 Profil d'échange de disposition du diagramme

5.1 Généralités

Espace de noms de profil: <http://iec.ch/TC57/61970-453/DiagramLayout/3#>

La présente norme spécifie un profil d'échange pour les schémas avec les caractéristiques suivantes.

- Une méthode générique pour lier l'objet diagramme aux données de domaine. Les données de domaine et les données de disposition du diagramme peuvent être échangées ensemble avec les données de domaine, ou séparément les unes des autres avec l'hypothèse que les données de domaine auront déjà été importées si les données de disposition du diagramme sont importées séparément.
- Elle n'exige ni ne suppose un format spécifique de données de domaine. Par conséquent, elle prend en charge les données de domaine modélisées conformément au modèle d'information commun (CIM) de l'IEC 61970-301 qui sont échangées dans le format conforme à l'IEC 61970-501 (schéma CIM RDF). Les données de disposition du diagramme référencent les données de domaine en conformité avec l'IEC TR 62541-1, *OPC Unified Architecture*).
- L'usage prévu de la présente norme est que la source d'un diagramme utilisera la norme pour coder la disposition de son diagramme tel qu'elle l'a créé. Il incombera alors à tout récepteur de fournir les moyens d'utiliser cette disposition du diagramme dans son système.
 - Dans la situation la plus simple, si la source utilise des placements d'objets Switch, le récepteur crée un modèle de Switch générique dans le système du récepteur qui sera utilisé pour rendre des Switchs (c'est-à-dire qu'il n'y a pas d'échange de modèles de Switch).

- Dans des situations plus complexes, une correspondance biunivoque (1:1) peut ne pas être souhaitable, et les récepteurs peuvent créer des transformations génériques afin d'utiliser des échanges.
- Toutefois, dans les deux situations, il est prévu qu'une fois qu'un investissement est réalisé dans une stratégie de rendu, des importations répétées peuvent être réalisées automatiquement.
- Prend en charge l'échange d'objets de diagramme qui n'ont pas de relation avec les données de domaine, c'est-à-dire, les objets purs d'arrière-plan statique.
- Prend en charge des représentations multiples du même objet de domaine dans les mêmes diagrammes ou des diagrammes différents.
- Permet au diagramme d'être utilisé comme unité d'échange, fournissant une approche simple pour l'échange partiel, ou un échange entre les systèmes qui ont un modèle et un référentiel distincts pour les données de disposition de diagramme.
- Prend en charge l'affectation d'objets de diagramme à des couches ou autres moyens, pour montrer ou cacher des informations en fonction du niveau de zoom et/ou de l'intérêt de l'utilisateur.
- Une méthode générique pour des extensions propriétaires afin de permettre le trajet aller-retour complet (exportation et réimportation dans le même système) sans perte d'informations dans le système et sans casser le format d'échange normalisé.

5.2 Classes de profils de disposition du diagramme

5.2.1 Généralités

Les classes de disposition du diagramme font partie de l'IEC 61970-301 et elles sont pour cela associées aux données de domaine en utilisant la relation d'association UML normalisée. Un nouveau sous-paquetage dans l'IEC 61970-301 contient les classes nécessaires pour exprimer des données schématiques pour des objets de domaine comme illustré dans la Figure 5.



Figure 5 – Modèle d'informations de disposition du diagramme

5.2.2.1 Généralités

Le Diagramme représente le diagramme échangé. Les attributs `initialView` peuvent être utilisés pour spécifier une vue initiale avec les coordonnées `x`, `y` des points diagonaux. Le système de coordonnées est un système normalisé de coordonnées cartésiennes et l'attribut "orientation" définit une orientation positive ou négative. Une orientation positive donne une orientation "du côté droit", alors qu'une orientation négative indique une orientation "du côté gauche". Pour les diagrammes 2D, une orientation positive se traduira par les valeurs `X` croissant de gauche à droite et les valeurs `Y` croissant de bas en haut. Une orientation négative donne une orientation "du côté gauche"; (privilégiée par les affichages graphiques des ordinateurs) avec les valeurs `X` croissant de gauche à droite et les valeurs `Y` croissant de haut en bas.

5.2.2.2 Membres natifs

orientation	0..1	OrientationKind	Le Diagramme représente le diagramme échangé. Les attributs initialView peuvent être utilisés pour spécifier une vue initiale avec les coordonnées x, y des points diagonaux. Le système de coordonnées est un système normalisé de coordonnées cartésiennes et l'attribut "orientation" définit une orientation positive ou négative.
x1InitialView	0..1	Float	Coordonnée X du premier coin de la vue initiale
x2InitialView	0..1	Float	Coordonnée X du deuxième coin de la vue initiale
y1InitialView	0..1	Float	Coordonnée Y du premier coin de la vue initiale
y2InitialView	0..1	Float	Coordonnée Y du deuxième coin de la vue initiale

5.2.2.3 Membres hérités

mRID	0..1	String	voir IdentifiedObject
name	0..1	String	voir IdentifiedObject
aliasName	0..1	String	voir IdentifiedObject
description	0..1	String	voir IdentifiedObject

5.2.3 DiagramObject

5.2.3.1 Généralités

La classe DiagramObject définit un objet qui définit un ou plusieurs points dans un espace donné. Cet objet peut être associé à tout ce qui constitue une sous-classe de la classe *IdentifiedObject* définie dans l'IEC 61970-301.

5.2.3.2 Membres natifs

drawingOrder	0..1	Integer	L'ordre dans le dessin de cet élément. Plus le numéro est grand, plus l'élément est dessiné en dernier dans la séquence. Il est utilisé pour s'assurer que les éléments qui se chevauchent sont rendus dans l'ordre correct.
isPolygon	0..1	Boolean	Définit si les points d'objets de diagramme définissent ou pas les limites d'un polygone ou le cheminement d'une polyligne. Si cette valeur est vraie, il convient qu'une application destinataire prenne en considération le premier point et le dernier point à connecter.
offsetX	0..1	Float	Le décalage dans la direction X. Il est utilisé pour définir le décalage à partir du centre pour rendre une icône (par défaut, un seul point spécifie le centre de l'icône). Le décalage est par unité, avec 0 indiquant qu'il n'y pas de décalage par rapport au centre horizontal de l'icône. -0,5 indique un décalage de 50 % vers la gauche et 0,5 indique un décalage de 50 % vers le côté droit.
offsetY	0..1	Float	Le décalage dans la direction Y. Il est utilisé pour définir le décalage à partir du centre pour rendre une icône (par défaut, un seul point spécifie le centre de l'icône). Le décalage est par unité, avec 0 indiquant qu'il n'y pas de décalage par rapport au centre vertical de l'icône. La direction du décalage dépend de l'orientation du diagramme, avec -0,5 et 0,5 indiquant un déplacement de $\pm 50\%$ sur l'axe vertical.
rotation	0..1	AngleDegrees	Définit l'angle de rotation (en degrés) de l'objet diagramme dans le sens des aiguilles d'une montre à partir de la normale
Diagram	1..1	Diagram	Un objet de diagramme constitue une partie d'un Diagramme
DiagramObjectPoints	0..illimité	DiagramObjectPoint	Un objet diagramme peut avoir 0 ou plus de points afin de refléter la position de sa disposition, le cheminement (pour les polygones) ou les limites (pour les polygones)
DiagramObjectStyle	0..1	DiagramObjectStyle	Un objet diagramme possède un style associé qui fournit une référence pour le style utilisé dans le système d'origine
IdentifiedObject	0..1	IdentifiedObject	L'objet de domaine auquel cet objet diagramme est associé
DiagramStyle	0..1	DiagramStyle	Un Diagramme peut avoir un DiagramStyle.

5.2.3.3 Membres hérités

mRID	0..1	String	voir IdentifiedObject
name	0..1	String	voir IdentifiedObject
aliasName	0..1	String	voir IdentifiedObject
description	0..1	String	voir IdentifiedObject

5.2.4 DiagramObjectGluePoint

Il est utilisé pour regrouper les DiagramObjectPoints à partir de différents DiagramObjects qui sont considérés comme "collés" ensemble dans un diagramme même s'ils ne sont pas exactement aux mêmes coordonnées.

5.2.5 DiagramObjectPoint

5.2.5.1 Généralités

Un DiagramObjectPoint est un point dans un espace donné défini par trois coordonnées et associé à un DiagramObject. Les coordonnées peuvent être positives ou négatives dès lors que l'origine ne doit pas se situer au coin d'un diagramme. L'attribut 'sequence' est utilisé lorsqu'un DiagramObject a plus d'un DiagramObjectPoint, auquel cas il définit l'ordre dans le dessin. Un DiagramObject peut représenter tout objet CIM. Pour les diagrammes unifilaires, ces objets peuvent inclure:

- Des valeurs analogiques de mesure (mesure)
- Breaker (disjoncteur)/disconnecter (sectionneur)
- Power transformer (transformateur de puissance)
- Transmission line (ligne de transport)

5.2.5.2 Membres natifs

sequenceNumber	0..1	Integer	La position de séquence du point, utilisée pour définir l'ordre des points pour les DiagramObjects agissant comme une polygline ou un polygone avec plus d'un point.
xPosition	1..1	Float	La coordonnée X de ce point
yPosition	1..1	Float	La coordonnée Y de ce point
zPosition	0..1	Float	La coordonnée Z de ce point
DiagramObject	1..1	DiagramObject	L'objet diagramme auquel les points sont associés
DiagramObjectGluePoint	0..1	DiagramObjectGluePoint	Le point d'adhérence auquel ce point est associé

5.2.6 DiagramObjectStyle

5.2.6.1 Généralités

DiagramObjectStyle est une référence à un style utilisée par le système d'origine pour un DiagramObject. Un DiagramObjectStyle décrit des informations telles que:

- épaisseur de ligne
- forme, par exemple cercle, rectangle ...
- couleur

5.2.6.2 Membres hérités

mRID	0..1	String	voir IdentifiedObject
name	0..1	String	voir IdentifiedObject
aliasName	0..1	String	voir IdentifiedObject
description	0..1	String	voir IdentifiedObject

5.2.7 TextDiagramObject

5.2.7.1 Généralités

Un TextDiagramObject est un objet diagramme pour placer un texte libre ou un texte provenant d'un objet de domaine associé.

5.2.7.2 Membres natifs

text	0..1	String	Le texte qui est affiché par cet objet diagramme textuel
------	------	--------	--

5.2.7.3 Membres hérités

drawingOrder	0..1	Integer	voir DiagramObject
isPolygon	0..1	Boolean	voir DiagramObject
offsetX	0..1	Float	voir DiagramObject
offsetY	0..1	Float	voir DiagramObject
rotation	0..1	AngleDegrees	voir DiagramObject
Diagram	1..1	Diagram	voir DiagramObject
DiagramObjectPoints	0..illimité	DiagramObjectPoint	voir DiagramObject
DiagramObjectStyle	0..1	DiagramObjectStyle	voir DiagramObject
IdentifiedObject	0..1	IdentifiedObject	voir DiagramObject
mRID	0..1	String	voir IdentifiedObject
name	0..1	String	voir IdentifiedObject
aliasName	0..1	String	voir IdentifiedObject
description	0..1	String	voir IdentifiedObject

5.2.8 VisibilityLayer

5.2.8.1 Généralités

Les couches (layers) sont typiquement utilisées pour regrouper des objets de diagramme selon des thèmes et des échelles. Les thèmes sont utilisés pour afficher ou cacher certaines informations (par exemple: lacs, frontières), alors que les échelles sont utilisées pour cacher ou afficher des informations selon le niveau de zoom courant (cacher le texte lorsqu'il est trop petit à lire ou lorsqu'il dépasse la taille de l'écran). Cela s'appelle aussi «désencombrement» ou «de-cluttering».

Le profil de disposition du diagramme prendra en charge une relation m:n entre les objets de diagramme et les couches. Il incombera au système importateur la tâche de convertir un cas m:n en une représentation 1:n appropriée si le système importateur ne prend pas en charge de relation m:n.

5.2.8.2 Membres natifs

drawingOrder	0..1	Integer	L'ordre dans le dessin pour cette couche. En ce qui concerne le drawingOrder pour les objets diagrammes, plus le numéro est grand, plus la couche et les objets y contenus sont rendus en dernier.
VisibleObjects	1..illimité	DiagramObject	Une VisibilityLayer (couche de visibilité) peut contenir un ou plusieurs objets de diagramme

5.2.8.3 Membres hérités

mRID	0..1	String	voir IdentifiedObject
name	0..1	String	voir IdentifiedObject
aliasName	0..1	String	voir IdentifiedObject
description	0..1	String	voir IdentifiedObject

5.2.9 Classes abstraites - IdentifiedObject

5.2.9.1 Généralités

Il s'agit d'une classe racine (root) dans le but de fournir une identification commune pour toutes les classes nécessitant des attributs d'identification et de dénomination.

5.2.9.2 Membres natifs

nom	mult	type	description
aliasName	0..1	String	L'aliasName est un nom de l'objet en texte libre lisible par l'utilisateur en remplacement de IdentifiedObject.name. Il peut ne pas être unique et peut ne pas se corréler à une hiérarchie de dénomination. L'attribut aliasName a été conservé pour des raisons de rétrocompatibilité avec les précédentes versions du CIM. Il est cependant recommandé de remplacer aliasName par la classe Name puisqu'il est prévu de supprimer aliasName à l'avenir.
description	0..1	String	La description est un texte libre lisible par l'utilisateur décrivant ou nommant l'objet. Il peut ne pas être unique et peut ne pas se corréler à une hiérarchie de dénomination.
mRID	0..1	String	Identificateur Master resource publié par une autorité de modèle. Le mRID (identificateur Master resource) est globalement unique dans un contexte d'échange. Il est facile d'obtenir une unicité globale en utilisant un UUID, tel que spécifié dans le RFC 4122, pour le mRID. L'utilisation d'un UUID est fortement recommandée. Pour des fichiers de données CIMXML en syntaxe RDF, en conformité avec l'IEC 61970-552 Edition 1, le mRID est cartographié en attributs rdf:ID ou rdf:about qui identifient les éléments d'objet du CIM.
name	0..1	String	Le nom est un texte, librement lisible par l'utilisateur et éventuellement non unique, qui nomme l'objet.

5.2.10 Énumérations – OrientationKind

negative (négatif)
positive (positif)

5.2.11 DiagramStyle

5.2.11.1 Généralités

Le style de diagramme fait référence à un style utilisé dans le système d'origine pour un diagramme. Un style de diagramme décrit des informations schématiques, géographiques, liées à la topologie nodale, etc.

5.2.11.2 Membres natifs

Diagram	0..*	Diagram	Un DiagramStyle peut être utilisé par de nombreux Diagrams.
---------	------	---------	---

5.2.11.3 Membres hérités

aliasName	0..1	String	voir IdentifiedObject
description	0..1	String	voir IdentifiedObject
mRID	0..1	String	voir IdentifiedObject
name	0..1	String	voir IdentifiedObject

6 Rendu graphique

6.1 Généralités

L'Article 6 illustre l'utilisation du modèle d'objet décrit à l'Article 5. Il utilise les conventions montrées à la Figure 6 pour représenter des instances d'objet et les relations.

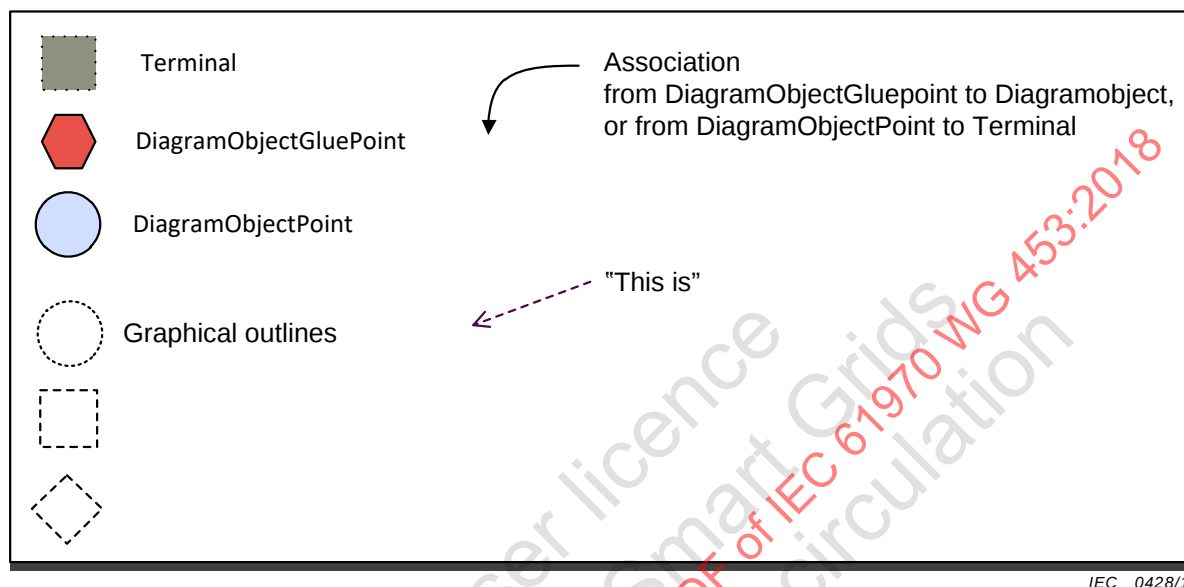
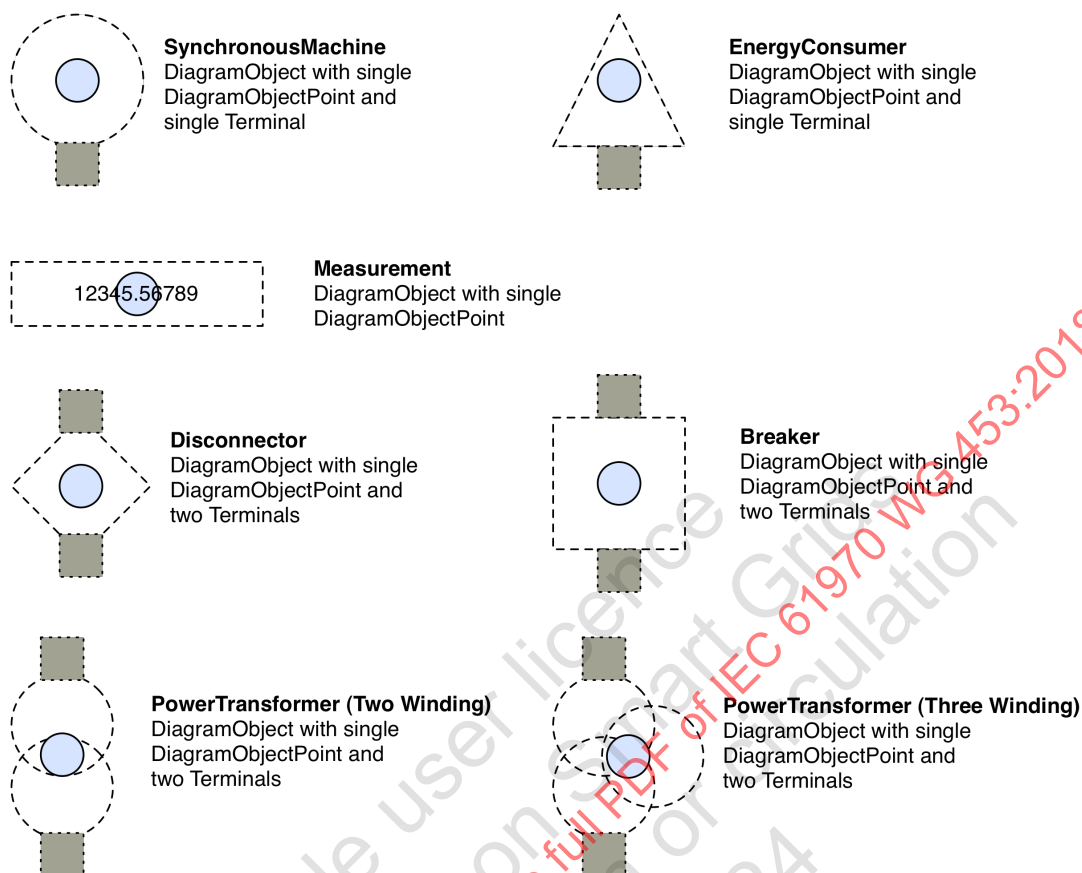


Figure 6 – Conventions utilisées pour représenter des instances d'objet et les associations

6.2 Objets à point unique

Les objets de diagramme les plus simples possèdent un point unique sous la forme d'un DiagramObjectPoint. Ils sont utilisés pour spécifier le point central d'une icône et peuvent représenter des appareils de réseau ou des emplacements de mesure (mesure) sur un diagramme. Cela est montré à la Figure 7.



IEC 0429/14

Légende

Anglais	Français
Terminal	Borne
SynchronousMachine DiagramObject with single DiagramObjectPoint and single Terminal	Machine synchrone DiagramObject avec un seul DiagramObjectPoint et un seul Terminal
EnergyConsumer DiagramObject with single DiagramObjectPoint and single Terminal	Consommateur d'énergie DiagramObject avec un seul DiagramObjectPoint et un seul Terminal
Measurement DiagramObject with single DiagramObjectPoint	Mesure DiagramObject avec un seul DiagramObjectPoint
Disconnecter DiagramObject with single diagramObjectPoint and two Terminals	Sectionneur DiagramObject avec un seul DiagramObjectPoint et deux Terminals
Breaker DiagramObject with single diagramObjectPoint and two Terminals	Disjoncteur DiagramObject avec un seul DiagramObjectPoint et deux Terminals
PowerTransformer (Two Winding) DiagramObject with single diagramObjectPoint and two Terminals	Transformateur de puissance (deux enroulements) DiagramObject avec un seul DiagramObjectPoint et deux Terminals
PowerTransformer (Three Winding) DiagramObject with single diagramObjectPoint and two Terminals	Transformateur de puissance (trois enroulements) DiagramObject avec un seul DiagramObjectPoint et deux Terminals

Figure 7 – Objets de diagramme à point unique