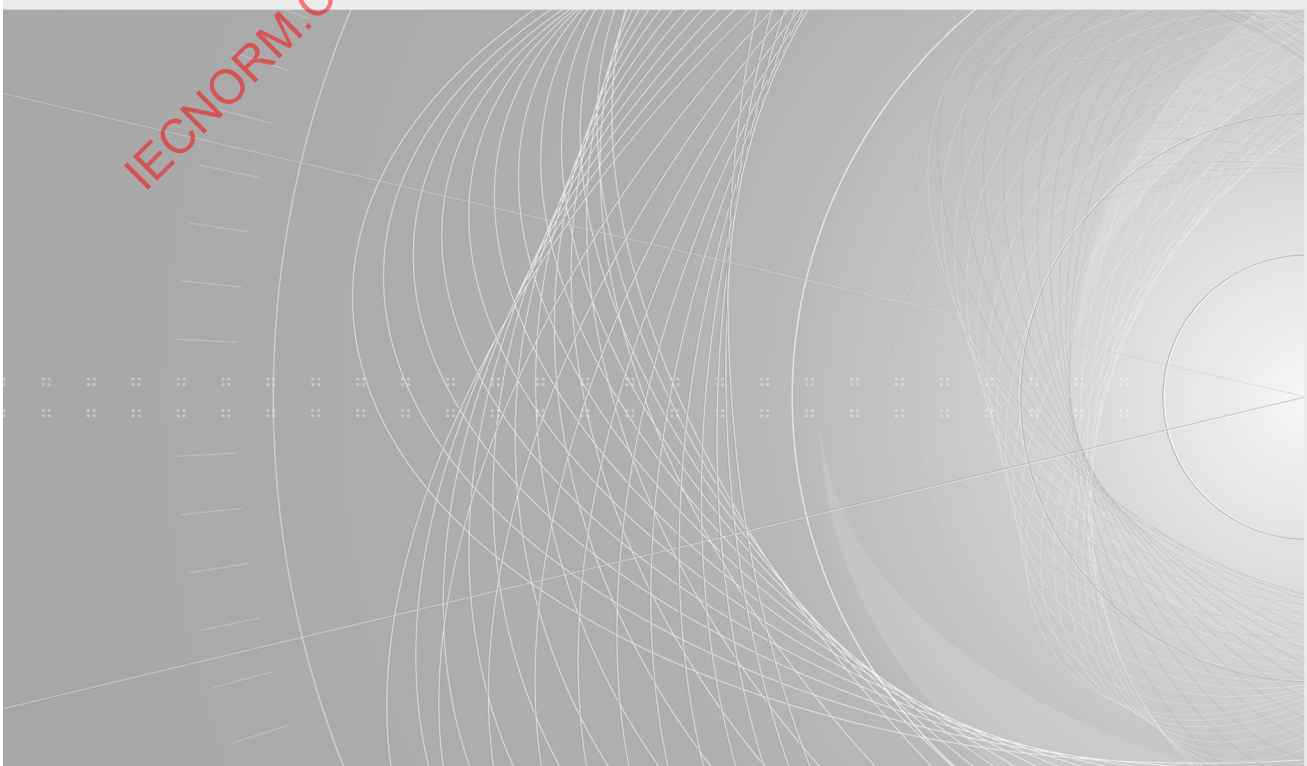
IEC TS 61970-555

Edition 1.0 2016-09

TECHNICAL SPECIFICATION

colour inside

Energy management system application program interface (EMS-API) – Part 555: CIM based efficient model exchange format (CIM/E)



IEC TS 61970-555:2016-09(en)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

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

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

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

IEC Catalogue - webstore.iec.ch/catalogue
The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing 20 000 terms and definitions in English and French, with equivalent terms in 15 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary
65 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

IEC TS 61970-555

Edition 1.0 2016-09

TECHNICAL SPECIFICATION

colour inside

Energy management system application program interface (EMS-API) – Part 555: CIM based efficient model exchange format (CIM/E)

INTERNATIONAL ELECTROTECHNICAL COMMISSION

ICS 33.200

ISBN 978-2-8322-3653-6

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

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

Part 555: CIM based efficient model exchange format (CIM/E)

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.

The main task of IEC technical committees is to prepare International Standards. In exceptional circumstances, a technical committee may propose the publication of a technical specification when

- the required support cannot be obtained for the publication of an International Standard, despite repeated efforts, or
- the subject is still under technical development or where, for any other reason, there is the future but no immediate possibility of an agreement on an International Standard.

Technical specifications are subject to review within three years of publication to decide whether they can be transformed into International Standards.

IEC TS 61970-555, which is a technical specification, has been prepared by IEC technical committee 57: Power systems management and associated information exchange.

The text of this technical specification is based on the following documents:

Enquiry draft	Report on voting
57/1730/DTS	57/1769/RVC

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

This document 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 website under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- transformed into an International standard,
- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

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

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 part of IEC 61970, which is a technical specification, describes ‘CIM-E’, an alternative to IEC 61970-552 (CIM/XML) for serializing CIM data exchanges, which is in use in China. It is a general method of serialization that could be used with any defined exchange, but it was developed specifically as an alternative to IEC 61970-552 for exchanging power system network models and solutions, which are defined in IEC 61970-452 and IEC 61970-456. Following is an explanation as to why the IEC has formally documented this method in addition to IEC 61970-552.

When the IEC first developed its standard for exchanging power system models, the primary model exchange specification, IEC 61970-452, described exchange of a complete model. The companion standard, IEC 61970-552, specified how to serialize model data for exchange, and also described how to serialize ‘incremental updates’ of a model in which only changed parts are exchanged. This latter feature was included because complete models could be very large and require significant processing time, but in most real business situations, model transfers are updates of previous exchanges in which only small parts of the models have changed. Unfortunately, while the IEC understood the potential value of incremental processing, no ‘use cases’ were produced to document the situations in which full or incremental processing would be recommended.

At the time that the IEC was producing these documents, China was following the development of CIM standards and planning to use them. The situations in which they wanted to use CIM model exchange demanded high performance. The solution they chose was CIM-E, a more compact version of serialization than CIM/XML, which their applications could process quickly.

As time passed, the IEC expanded its set of exchanges to include solved cases and continued to envision incremental updates as the approach for optimizing performance. China began to participate actively in WG 13 meetings, and also continued to enhance the CIM-E serialization method, which is a national standard now in China.

In 2011, China proposed that IEC adopt CIM-E as a standard. The IEC concluded that a CIM-E specification was appropriate. China represents a significant part of the world’s electricity infrastructure. It is using CIM. The IEC therefore felt that an IEC specification for CIM-E would be useful, provided that it simply presented a specification for meeting a requirement for CIM-E exchange and did not endorse CIM-E as the recommended method for achieving high performance.

The IEC 61970 series defines an Application Program Interface (API) for an Energy Management System (EMS).

IEC 61970-301 specifies a Common Information Model (CIM): a logical view of the physical aspects of electric utility operations. The CIM is described using the Unified Modeling Language (UML): a language used to specify, visualize, and document systems in an object-oriented manner. UML is an analysis and design language; it is not a programming language. For software programs to use the CIM it needs to be transformed into a schema form that supports a programmable interface.

This part of IEC 61970 describes the translation of the CIM in UML form into a machine readable format as expressed in CIM based Efficient Model Exchange Format(CIM/E) representation of that schema using the CIM/E Schema.

This part of IEC 61970 specifies how the CIM/E schema is used to support power system models or particular application data models exchange requirements, especially for real-time or online applications. Similar to CIM/XML, CIM/E is an efficient serialization format to describe CIM objects. The power system model described by CIM/E or by CIM/XML can be converted bi-directionally with consistent result.

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

Part 555: CIM based efficient model exchange format (CIM/E)

1 Scope

This part of IEC 61970, which is a technical specification, specifies a Component Interface Specification (CIS) for Energy Management Systems Application Program Interfaces. This part specifies the format and rules for exchanging modelling information based upon the CIM. It uses the CIM/E Schema as the meta-model framework for constructing CIM/E documents of power system modelling information. The style of these documents is called CIM/E format. CIM/E is suitable for use in online model exchange of power system applications.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61970-452, *Energy management system application program interface (EMS-API) – Part 452: CIM model exchange specification*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

3.1
Application Program Interface
API
the set of public functions provided by an executable application component for use by other executable application components

3.2
Common Information Model
CIM
abstract model that represents all the major objects in an electric utility enterprise typically contained in an EMS information model

Note 1 to entry: By providing a standard way of representing power system resources as object classes and attributes, along with their relationships, the CIM facilitates the integration of EMS applications developed independently by different vendors. It provides a common information presentation between EMS systems developed independently, or between an EMS system and other systems concerned with different aspects of power system operations, such as generation or distribution management.

3.3
CIM/E Schema
schema specification expressed by using CIM/E format to describe classes and their attributes and associations

3.4
Energy Management System
EMS
Computer system comprising a software platform providing basic support services and a set of applications providing the functionality needed for the effective operation of electrical generation and transmission facilities so as to ensure adequate security of energy supply at minimum cost

3.5
profile
schema that defines the structure and semantics of a model that may be exchanged

Note 1 to entry: A Profile is a restricted subset of the more general CIM.

3.6
Unified Modelling Language
UML
object-oriented modelling language and methodology for specifying, visualizing, constructing, and documenting the artefacts of a system-intensive process

3.7
Uniform Resource Identifier
URI
object-oriented modelling language and methodology for specifying, visualizing, constructing, and documenting the artefacts of a system-intensive process

4 Use case

4.1 General

With the rapid development of the power industry and renewable energy in the world, a large scale of random intermittent wind power and photovoltaic power are integrated to grids and system operators are faced with great challenges. There is a pressing need for the cooperation of multiple control centres to know the real-time situation of the adjacent system and all the power grids. Therefore, the real-time model and graphics of power systems should be shared among all related control centres. Both the 'ENTSO-E Real-time Awareness and Alarm System' project in Europe and the 'Smart Grid Dispatching Platform' project in China attempt to achieve this goal under urgent requirements.

4.2 Real-time sharing on CIM model

Consider an interconnection power grid with two Control Centres, A and B, as shown in Figure 1. Each control centre has an EMS from different vendors. Although A has no direct responsibility for parts of the grid supervised by B, B still has an influence on A's grid. A is, therefore, obliged to maintain a model of these additional parts in at least enough detail to gauge their effect on its own part of the grid.

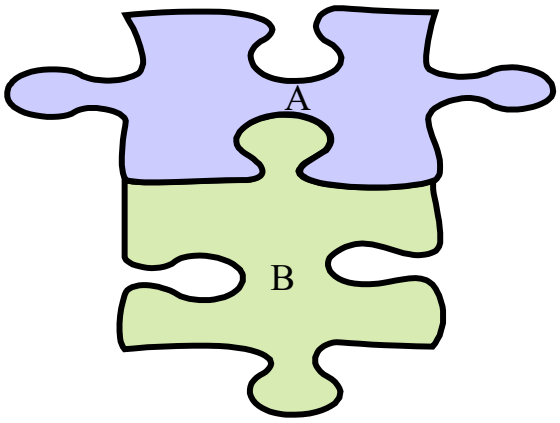


Figure 1 – Interconnection power grid

Each control center maintains an official, detailed model of its own territory, and regularly makes all updates available to its neighbors by CIM/E. Each control center receives its neighbors' models and merges them together into a full detailed interconnection model, as shown in Figure 2.

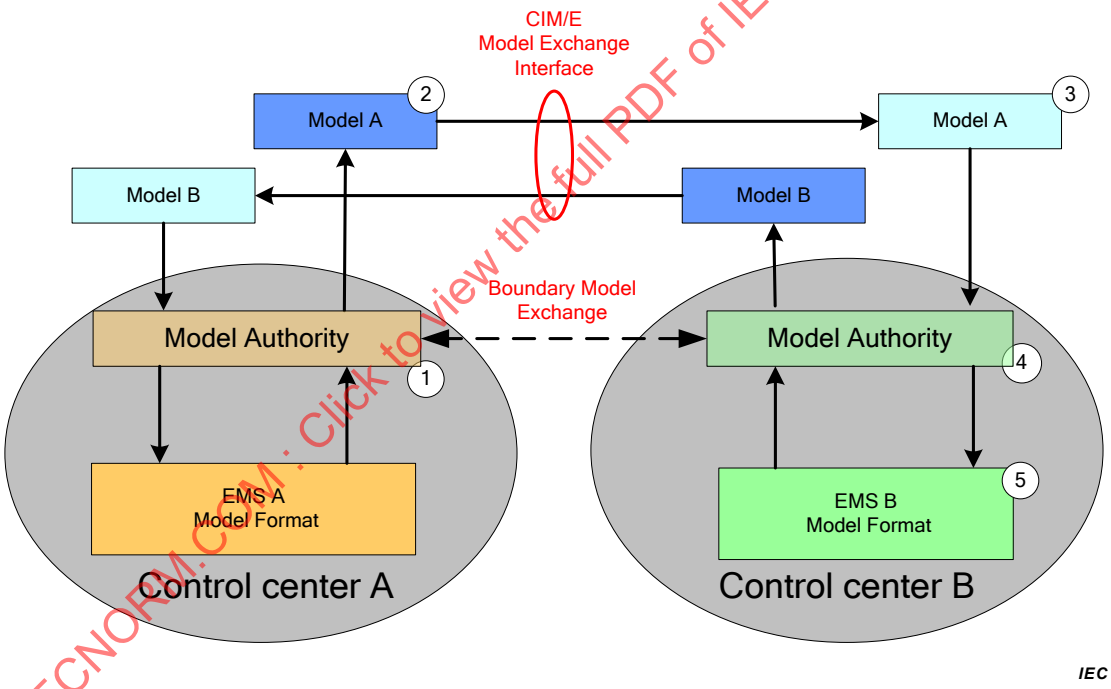


Figure 2 – CIM model exchange and merge

This diagram shows the interaction between these two control centers, A and B. All actions are the same at each control center. Therefore, we can simply follow the steps as A makes a change:

a) A makes a change in its system model using its local EMS modeller.

b) A's model authority releases either its full model or the increment to CIM/E file and sends it to B.

c) B receives and verifies the model.

d) B extracts A's territory from the model according to the boundary and merges them together into a detailed interconnection model (B reduces A if necessary).

e) B imports the resulting model of A into its Model Authority and its EMS model.

4.3 Online Dynamic Stability Analysis (DSA)

The CIM/E solution has been used for an on-line Dynamic Stable Analysis (DSA) project in China, as shown in Figure 3. The DSA is activated by real-time event or runs every 15 minutes. It is also applied on smart grid dispatching platform pilot projects, which is a new generation platform for EMS, DSA, WAMS and MMS. These projects have been put into operation at many control centers and the main objective of sharing real-time models for the whole power system among multiple control centres has been achieved.

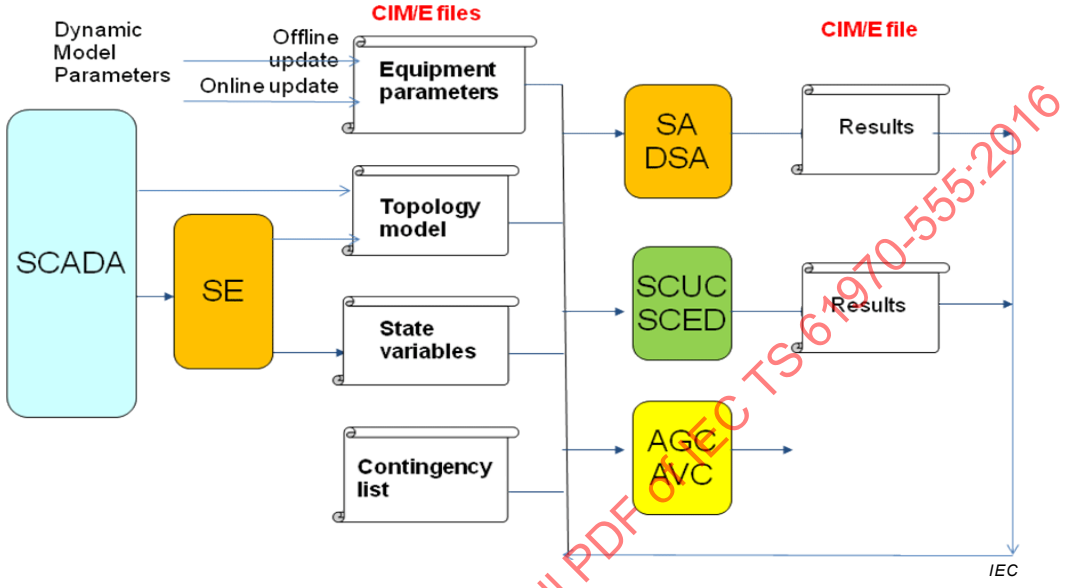


Figure 3 – Online dynamic stability analysis

5 Symbol definitions

The symbols used in this standard are specified in Table 1.

Table 1 – Symbol definitions

Number	Symbol	Definition	Note
1	<>	class start tag	Compliant with XML
2	</ >	class end tag	Compliant with XML
3	<! I>	system declaration	Compliant with XML
4	<@>	Data block header leading for transverse-table	Compliant with XML
5	<@#>	Data block header leading for vertical-table	Compliant with XML
6	<#>	data line start tag	Compliant with XML
7	//	comment start tag	Non-XML
8	:	namespace connector	Compliant with XML
9	::	class and instance entity connector	Non-XML
10	=	assignment connector	Compliant with XML
11	.	connector for names	Compliant with XML
12	*	pointer leading	Non-XML
13	blank	Field separator, may be one or more spaces or tab characters	Compliant with XML
14	’	single quotation for string	Compliant with XML
15	NULL	empty indicator	Compliant with XML
16	-	Unchanged or non-value indicator	Non-XML
17	,	comma separator	Compliant with XML

6 Structure Specification

6.1 General

A power system model can be described as a CIM/E data file according to the CIM/E schema, specified in Figure 1. The CIM/E schema is a RDF schema specification for CIM/E document. The resulting CIM/E data file, which is used as model exchange document, can be parsed and imported into an external system. The model exchange mechanism based on CIM/E is shown in Figure 4.

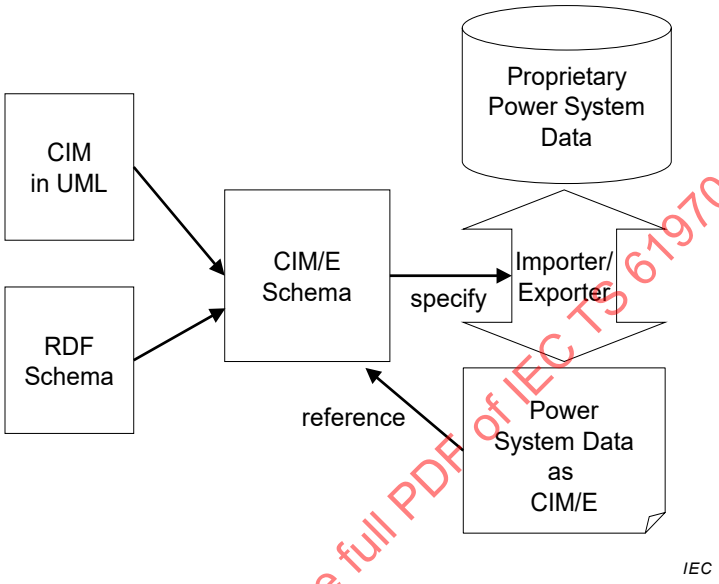


Figure 4 – Model exchange mechanism based on CIM/E

CIM/E is in plain text format. The first two or three characters of each line in a CIM/E format file are used to indicate the meaning of the line for efficient text processing.CIM/E is case-sensitive.

6.2 Basic structure

CIM/E file consists of system declaration, one model header, one or more blocks. There are three kinds of blocks: Data Block, Schema Block and Enumeration Block. The comment line starts with '//' indicating a comment and could appear anywhere in the file. The CIM/E file has two basic table structures; one is transverse-table structure and the other is vertical-table structure. They are specified in Figure 5 and Figure 6 respectively.

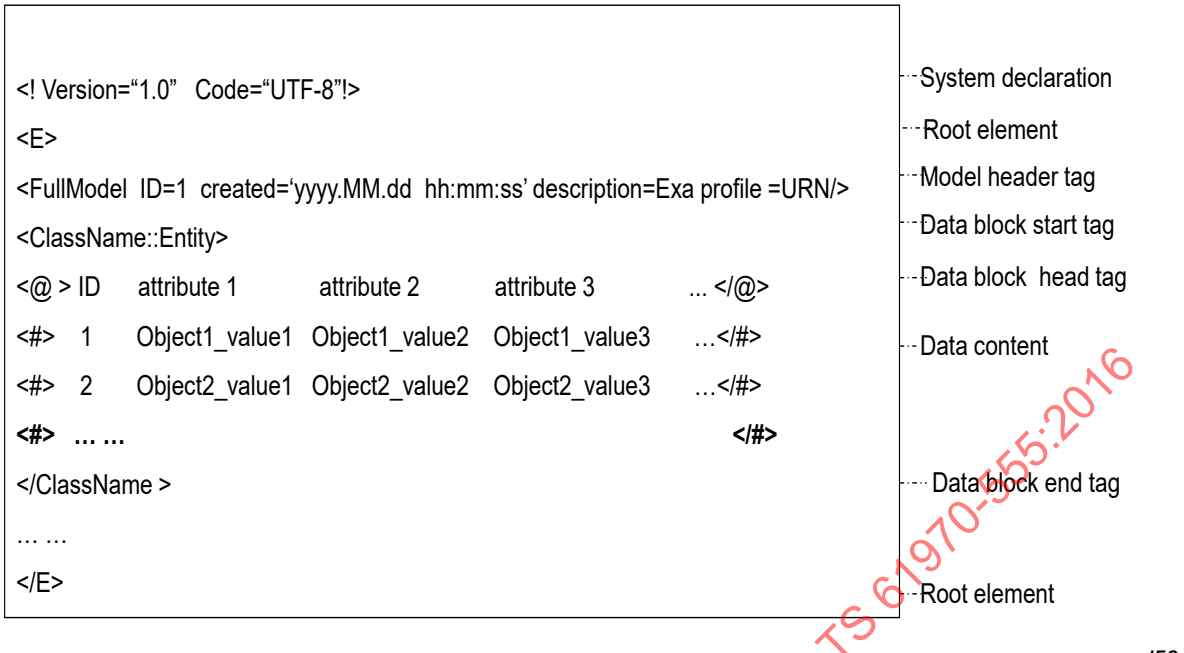


Figure 5 – Transverse-table structure

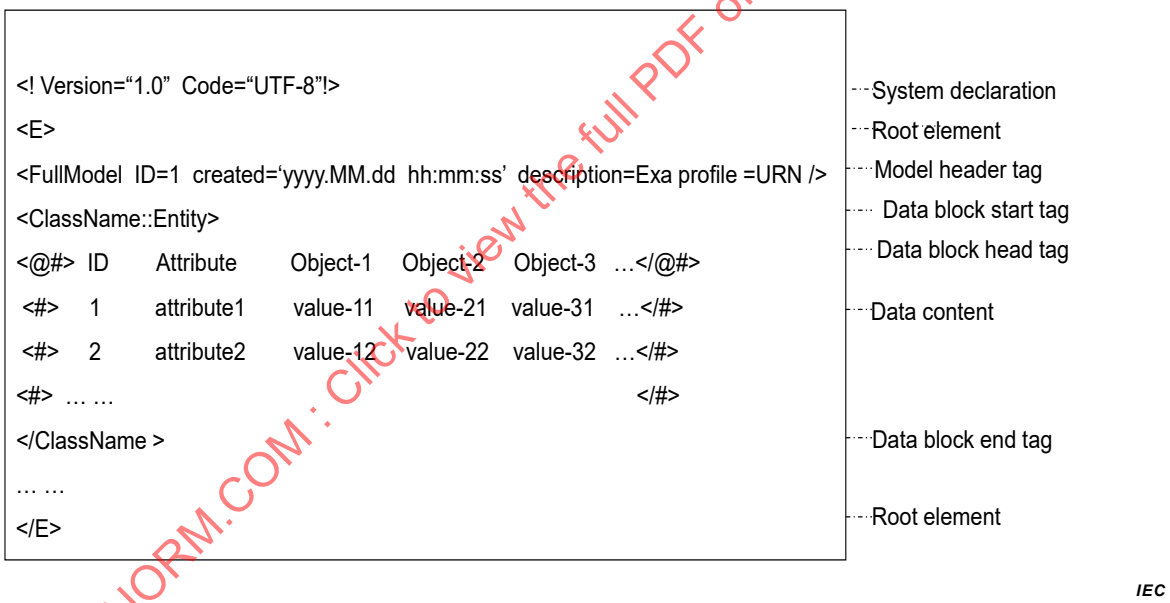


Figure 6 – Vertical-table structure

6.3 Comment

Comment is optional. It starts with double slashes '//'. A comment may appear anywhere in the data file, either following a statement or starting a new line.

6.4 System declaration

The system declaration of CIM/E must be described before the root element. It starts with'<!' and ends with'!>'. The attributes in the system declaration line include: *Version, Code*, others if needed. The details are as follows:

a) *'Version'* is the version of IEC 61970-301 referenced by CIM/E schema.

b) *'Code'* is the character set used in this document.

6.5 Root element

The root element <E> is the parent object of all other model elements in a model block file.

6.6 Model header

This sub-clause references IEC 61970-552. A model header describes the content of the model contained in a document such as the date the model was created and the description of the model. The header may also identify other models and their relationship to the present model. Such information is important when the models are part of a work flow where, for example, the models have relations to each other, e.g. a model succeeds and/or depends on another.

A header of CIM/E must be described in root element. It starts with'<FullModel> or <DifferenceModel>'.The FullModel represents the full model header. The DifferenceModel represents the difference model header. The content in header includes created, description, version, profile and others if needed. The details are as follows:

a) *'created'* is the date when the model was created.

b) *'scenarioTime'* is the date and time that the model represents, e.g. the current time for an operational model, a historical model or a plan model.

c) *'description'* is a description of the model, e.g. the name of person that created the model and the purpose the model is created.

d) *'modelingAuthoritySet'* is a URN describing the equipment model sourcing the data in a CIM/E document, e.g. a model for the whole or a part of a country.

e) *'profile'* is a URN describing the Profiles that governs this model. It uniquely identifies the Profile and its version.

f) *'version'* is a description of the version of the model sourcing the data in a CIM/E document.

g) *'DependentOn'* is a reference to the models that the model described by this document depends on.

h) *'Depending'* is all models depending on this model. This role is not intended to be included in any document exchanging instance data.

i) *'Supersedes'* When a model is updated the resulting model is supersedes the models that were used as basis for the update. Hence this is a reference to CIM/E documents describing the updated models.

j) *'SupersededBy'* All models superseding this model. This role is not intended to be included in any document exchanging instance data.

6.7 Block structure

CIM/E supports three kinds of blocks: Data Block for object data model (full model, incremental model), Schema Block for schema description and Enumeration Block for enumeration description.

a) In Data Block, all objects of the same class are grouped together.It starts with '<ClassName::Entity>', and ends with '</ClassName>'. The ClassName indicates the class name of the data objects, and the Entity indicates the ascription of objects. For example, '<subControlArea::EastChina>' means the Class is 'subControlArea', and data objects belong to East China Grid.

b) Schema Block starts with "<Class Name='class name' >" and ends with '</Class>'. The value of Name is the class name described in the Block.

c) Enumeration Block starts with "<Enum Name='enumname'>" and ends with '</Enum>'. The value of Name is the enumeration name.

6.8 Data block

6.8.1 General

There are two kinds of basic structures in Data Block: transverse-table structure and vertical-table structure. The transverse-table structure is suitable for the model data with a few attributes but many objects (Figure 5), while the vertical-table structure is suitable for the model data with few objects but many attributes (Figure 6).

6.8.2 Data block header

Data Block header is defined as follows.

a) For transverse-table structure, Data Block header starts with '<@>' and ends with '</@>':

<@>*ID AttrName1 AttrName2 AttrName3…*</@>

Where, the ID indicateobject name or ID, can be omitted. AttrNameX are attribute names and they are separated by blanks.

b) For the vertical-table structure, Data Block header starts with '<@#>' and ends with '</@#>':

<@# >*ID AttrName Object1 Object2 Object3…*</@#>

Where, the ID indicate serial number of each attribute, can be omitted, the AttrName is the attribute name, the ObjectX is the object name or URI.

6.8.3 Data block contents

Data Block contents are composed of several data lines, and each line starts with'<#>' and ends with '</#>'.

a) For transverse-table structure (<@>), attribute value in its column of the data line corresponds to the attribute name in its column of the block header one by one. Each data line starts with '<#>' which is followed by blanks and the ID. The ID value could be URI, mRID, object name or number, ID can be omitted. The lists of attribute values are separated by blank.

b) For vertical-table structure (<@#>), '<#>' is followed by blanks and the ID. The ID value is the serial number of the object, can be omitted. After the ID, column (AttrName) is the object attribute name, and the other columns (ObjectX) are the attribute values of ObjectX. The list of object values is separated by blank.

c) If the ID is the serial number of the object, it should start from 1.

d) If the attribute value is null, then it should be assigned by 'NULL'.

6.8.4 Blank separator

Both the space character (' ') and Tab ('\t') can be used as Blank separator. In the value of an attribute, if there is a space character or comma, it should be enclosed by a pair of single quotes (''), like 'my name'. If there is a single quote, it should be enclosed by a pair of double quotes (""), like "it's good". The double quotes and single quotes should not appear in the same value of an attribute.

If an attribute is multiple-valued, its values should be separated by a comma ','.

6.8.5 Assignment connector

The equal sign'=' is used to assign attribute value to its attribute name.

For example:

<FullModel description=' Grid model of test'…/>

6.8.6 Object pointer

An asterisk'*' is used for object pointer. When '*' appears in front of the attribute name in the Data Block header, it means that the attribute value is an object pointer. The URI is contained in the reference. If the pointer refers to many objects, the object identifier will be separated by a comma ',', An example of an object pointer structure is shown in Figure 7.

```
<! Version="1.0" Code="UTF-8"!>
<E>
…
<ClassnameA : : XXX>
<@>  URI        name       pathName    …</@>
<#>  A_obj1     A_nm1      A_path1     …</#>
<#>  A_obj2     A_nm2      A_path2     …</#>
<#>  A_obj3     A_nm3      A_path3     …</#>
… …
</ClassnameA>

<ClassnameB : : YYY>
<@>  URI        name       pathName    *ClassnameA      …</@>
<#>  B_obj1     B_nm1      B_path1     A_obj1           …</#>
<#>  B_obj2     B_nm2      B_path2     A_obj2 , A_obj3  …</#>
<#>  B_obj3     B_nm3      B_path3     A_obj4           …</#>
… …
</ClassnameB>
…
</E>
```

IEC

Figure 7 – Usage example of object pointer

6.8.7 Compound

The attribute usually has a simple data type; this may be compound data type or nested compound type. But in the tables of CIM/E, the compound attributes should be spread into simple type attributes. A composite of primitive, enumeration, CIM Data type or any other compound classes could be used, and could be nested. For a detailed definition of compound type see Clause 8, CIM/E Schema. An example of compound attribute is shown as Figure 8.

```
… …
<Location::huadong>
<@>  mRID  name    s.number    s.name              t.code    t.name …</@>
<#>  1     OPT     120         'No.14 Pek Road'    100109    Beijing  …</#>
<#>  2     OPN     121         'No.16 Pek Road'    100109    Beijing  …</#>
</Location::huadong>
… …
```

IEC

Figure 8 – Usage example of compound type

7 Incremental model

7.1 General

After the initial full model data have been exchanged between A and B (A and B may be different control centers or power plants or substations), then once the model at A is updated, instead of transferring the fresh full model data completely, only the changed part is required to be sent as incremental model.

7.2 Structure of incremental model

Given two base CIM/E models, the incremental model includes the following four groups of statements, each encoded as a sequence of resource description structures.

- Model header is comprised of statements of the incremental model itself.
- Create statements are comprised of statements on the added objects.
- Change statements are comprised of statements on the modified objects.
- Delete statements are comprised of statements on the removed objects.

Two forms of formats can be used to describe the incremental model. The template of the first form is illustrated in Figure 9.

```
<! Version=”1.0” Code=”UTF-8" !>
< E >
<DifferenceModel ID=1 Created=’2013-5-2 12:50:42’ Description= ‘study model’ profile=URN1
Version=15v DependentOn=abce_11_1ba modelingAuthoritySet=’North China’/>
<Create>
<classname::Entity>
<@> ID      attr1    attr2    …   attrX</@>
<#>   1      value1  value2  …  valueX</#>
<#>   2      value1  value2  …  valueX</#>
</classname>
 </Create>
<Change>
<classname::Entity>
<@> ID      attr1    attr2    …   attrX</@>
<#>   1      value1  value2  …  valueX</#>
<#>   2      value1  value2  …  valueX</#>
</classname>
</Change>
<Delete>
<classname::Entity>
<@> ID      attr1    attr2    …   attrX</@>
<#>   1      value1  value2  …  valueX</#>
<#>   2      value1  value2  …  valueX</#>
</classname>
</Delete>
</E>
```

IEC

Figure 9 – The 1st form of incremental model description format

An alternative format to describe the incremental model includes another four groups of statements as follows.

- Model header, comprising of statements on the difference model itself.

- Forward difference statements, comprising of statements found in the new model, not in the old model.
- Reverse difference statements, comprising of statements found in the old model, not in the new model.
- Precondition statements, comprising of statements found in both the initial model and the results model and considered to be dependent to the difference model in an application defined sense.

The template of the second form is illustrated in Figure 10.

```
<! Version=”1.0” Code=”UTF-8" !>
< E >
<DifferenceModel ID=1 Created=’2013-5-2 12:50:42’ Description= ‘study model’ profile=URN1
Version=15v DependentOn=abce_11_1ba modelingAuthoritySet=’North China’/>
<preconditions>
<classname::Entity>
<@> Num    attr1    attr2    …  attrX</@>
<#>  1      value1  value2  …  valueX</#>
<#>  2      value1  value2  …  valueX</#>
</classname>
 </preconditions>
<forwardDifferences>
<classname::Entity>
<@> Num    attr1    attr2    …  attrX</@>
<#>  1      value1  value2  …  valueX</#>
<#>  2      value1  value2  …  valueX</#>
</classname>
</forwardDifferences>
<reverseDifferences>
<classname::Entity>
<@> Num    attr1    attr2    …  attrX</@>
<#>  1      value1  value2  …  valueX</#>
<#>  2      value1  value2  …  valueX</#>
</classname>
</reverseDifferences>
</E>
```

IEC

Figure 10 – The 2nd form of incremental model description format

7.3 Data block of incremental model

The actions to generate the incremental model can be categorized into create, delete and change, and the verbs defined in IEC 61968-100 are used directly in CIM/E.

Created data block starts with <Create> and ends with </Create> in the first form of incremental model while it starts with <forwardDifferences> and ends with </forwardDifferences> in the second form of incremental model. Likely primary key properties should be included in each created object.

Changed data block starts with <Change> and ends with </Change> in the first form of incremental model while it starts with <forwardDifferences> and ends with </forwardDifferences> in the second form of incremental model. Non-null object ID (like mRID) must be included in the data line. If an object ID is changed, it should be described by the deleting of an old object and the creating of a new object. For the changed attribute, the attribute

value is assigned by the changed value; for the unchanged attribute, attribute value is assigned by '-'.Deleted data block starts with <Delete> and ends with </Delete> in the first form of incremental model, while removed data block and pre-modified data block starts with <reverseDifferences> and ends with </reverseDifferences> in the second form of incremental model. Non-null object ID (like mRID) must be included in the data line. If an object is deleted, the cascaded objects should also be deleted. If the reference of the deleted object is from an object that remains, it should also be deleted. For example, if a VoltageLevel is deleted, its cascaded objects should be automatically deleted.

8 CIM/E Schema

8.1 General

In order to efficiently achieve and define the data schema, the CIM/E Schema description is introduced. Similar to the XML Schema, the CIM/E Schema is mainly used to describe the classes and their attributes, including Name, Type, Length, Size, Comment, etc, as shown in Figure 11. The CIM/E Schema description facilitates computer identification and the extension of the attributes. The meta-attributes of schema could be extended if needed. The CIM/E schema information could be imported from CIM RDF schema specified in IEC 61970-501 and vice versa.

```
<! System=* Code=UTF-8 Time='yyyyMMdd_hhmmss' !>
<E Ens:cim='CIM-schema-cim11#'>
...
<Class name='classname1' belongToCategory='Category'Ens='cim'>
<@> ID  Name      Alias     Type    Unit  Size  Inherit          Range  Multiplicty  maxVal  minVal  Comment  ns</@>
<#> 1   mRid      mRid      String  -     64    IdentifiedObject -      -            -       -       -        cim</#>
<#> 2   pathName  pathName  string  -     64    IdentifiedObject -      -            -       -       -        cim</#>
<#> 3   rx        rx        float   -     4     -                -      -            0       1       -        cim</#>
...
</Class>
...
</E>
```

IEC

Figure 11 – CIM/E schema description

A CIM/E model exchange document uses a subset of the CIM to address the model exchange needs of a specific use case (see IEC 61970-45x). A CIM profile defines a portion of the CIM that an importer and exporter of a CIM/E document should be expected. The CIM/E Schema for a profile only contains the classes and properties defined for that profile.

IEC 61970-555 specifies how the CIM/E schema is applied to exchange power system models (referred to as CIM) defined in the IEC 61970-45x series of profile standards, such as the IEC 61970-452 CIM Transmission Network Model Exchange Profile. CIM/E describes all the classes and properties that belong to each profile. The class name and property name in model data are all defined in the CIM/E schema.

8.2 Schema structure

The CIM/E Schema description adopts CIM/E transverse-table structure. It includes System Declaration, Root Element, Class Block, Enumeration Block, etc. In CIM/E Schema description, Root element starts with <E ns:cim='xxx'>, where 'ns' (name space) must be included. The name space is described as "ns:shortname='full name'", for example,<E ns:cim='http://iec.ch/TC57/2003/CIM-schema-cim11#'>. The expression "ns" is necessary for Name Space identification. Root element consists of two kinds of Data Block: Class Block which contains all the classes and their attributes, and Enumeration Block which contains all possible enumeration classes and their values.

8.3 Class description

The CIM/E Schema description starts with:

<Class Name='classname' belongsToCategory='category' ns='shortname'>

Where *'classname'* is the name of class. *'category'* is the category name which the class belongs to. *'shortname'* is the namespace that the class belongs to, which must be declared in Root element. Other attributes can be user-defined if needed.

The meta-attributes of the class are described as follows.

a) *'Name'* indicates name of the attribute. It corresponds to 'rdfs:label' of RDF schema.

b) *'Alias'* indicates alias name of the attribute. Typically, it is local name of class attribute.

c) *'Type'* indicates type of the attribute. It corresponds to 'cims:dataType' of CIM RDF schema, such as Integer, Float, String, Timestamp, etc.

d) *'Size'* indicates attribute length. It is the maximum length of attribute in bytes.

e) *'minVal'* indicates the minimum value for numeric attributes, or reasonable value for other type attributes. For example, year, month, date, hour, minute and second should be within reasonable range.

f) *'maxVal'* indicates the maximum value for numeric attributes, or reasonable value for other type attributes.

g) *'Unit'* indicates the unit of the attribute such as MW, MWh, kV, etc.

h) *'Inherit'* indicates inherited class name of this attribute. It defines the parent class name inherited by current class.

i) *'Range'* indicates the value range of the attribute, it corresponds to 'rdfs:range' of RDF schema.

j) *'Multiplicity'* indicates the allowed cardinality of the association. It corresponds to 'cims:multiplicity' of RDF schema. For example, the association end of Substation with VoltageLevel, see Figure 12.

[Multiplicity from]	[Multiplicity to] name	type	Description
[1..1]	[0..*] VoltageLevels	VoltageLevel	The voltage levels within this substation

IEC

Figure 12 – The multiplicity in class definition

k) *'Comment'* indicates attribute note. It corresponds to 'rdfs:comment' of RDF schema.

l) *'ns'* indicates namespace that the attribute belongs to. It must be declared in the Root element.

The class attributes in CIM/E can be classified into 3 types: basic attribute, relationship attribute and enumeration attribute. For basic attribute, the Type, Size, should not be null. For relationship attribute, the Range and Multiplicity should be non-null. For the enumeration attribute, the Type of enumeration attribute should be the enumeration name defined in Enumeration Block.

8.4 Enumeration block

The Enumeration Block is used for describing enumeration information by transverse-table structure. It includes system declaration, root element and enumeration definition. The Enumeration Block starts with:

<Enum name='enum-name' belongsToCategory='category'>

Where *'enum-name'* is the enumeration name, *'category'* is the name of package which the enumeration belongs to. Other attributes can be extended if needed. Enum name is necessary for Enumeration Block, which corresponds to the RDF Schema 'rdfs:label'.

```
… …
<Enum name='WindingConnection'  belongsToCategory='Wires'/>
<@> Value          Name        Comment</@>
//
<#>  1             Y           Wye</#>
<#>  2             Z           ZigZag</#>
<#>  3             D           Delta</#>
</Enum>
… …
```

IEC

Figure 13 – The example of enumeration definition

The enumeration classes defined in Enumeration Block can be directly referenced in Data Block. For example, the referenced enumeration name is 'WindingConnection.Z', as shown in 2 of Figure 13.

8.5 Compound type

In the CIM/E schema, the compound type include compound class and compound attribute, they are described as follows:

Compound Class: ParentClass.ChildClass

Compound Attribute: ParentAttribute.ChildAttribute

The dot notation is used to connect the parent class and child class, as well as parent attribute and child attribute. The compound type could be nested. The nest compound should be spread into simple attribute .In some cases, if all attribute names are different from each other in the upper class, the parent attribute could be left out. The parent class name included in the attribute name may be full or abbreviation, and abbreviation is preferred. An example of compound attribute is shown as Figure 14. The "s.number" can also be changed to "streetDetail.number". The "s" is the abbreviation of "streetDetail", and the "t" is the abbreviation of "TownDetail".

```
<!  System=*  Code=UTF-8   Time='20140112_000000'  !>
<E  Ens:cim='cim15#'>
 ...
<Class name='Location'  >
<@>  ID   Name       type          inherit                 …</@>
<#>   1   mRID       String        IdentifiedObject        …</#>
<#>   2   name       String        IdentifiedObject        …</#>
<#>   3   s.number   String        streetDetail            …</#>
<#>   4   s.name     String        streetDetail            …</#>
<#>   5   t.code     String        TownDetail              …</#>
<#>   6   t.name     String        TownDetail              …</#>
<#>   …                                                  …</#>
 </Class>
...
</E>
```

IEC

Figure 14 – Example of compound attribute definition