

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Energy management system application program interface (EMS-API) –
Part 552: CIMXML Model exchange format**

**Interface de programmation d'application pour système de gestion d'énergie
(EMS-API) –
Partie 552: Format d'échange de modèle CIMXML**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2013 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester.

If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de la CEI ou du Comité national de la CEI du pays du demandeur.

Si vous avez des questions sur le copyright de la CEI ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de la CEI de votre pays de résidence.

IEC Central Office
3, rue de Varembe
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.

Useful links:

IEC publications search - www.iec.ch/searchpub

The advanced search enables you 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It also gives information on projects, replaced and withdrawn publications.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 on-line and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary (IEV) on-line.

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.

A propos de la CEI

La Commission Electrotechnique Internationale (CEI) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications CEI

Le contenu technique des publications de la CEI est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Liens utiles:

Recherche de publications CEI - www.iec.ch/searchpub

La recherche avancée vous permet de trouver des publications CEI en utilisant différents critères (numéro de référence, texte, comité d'études,...).

Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

Just Published CEI - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications de la CEI. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne au monde de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans les langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (VEI) en ligne.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Energy management system application program interface (EMS-API) –
Part 552: CIMXML Model exchange format**

**Interface de programmation d'application pour système de gestion d'énergie
(EMS-API) –
Partie 552: Format d'échange de modèle CIMXML**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX



ICS 33.200

ISBN 978-2-8322-1180-9

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	3
INTRODUCTION.....	5
1 Scope.....	6
2 Normative references.....	6
3 Terms and definitions.....	7
4 Model exchange header.....	9
4.1 General.....	9
4.2 CIMXML documents and headers.....	9
4.3 Model and header data description.....	10
4.4 Work flow.....	12
5 Object identification.....	13
5.1 URIs as identifiers.....	13
5.2 About rdf:ID and rdf:about.....	14
5.3 CIMXML element identification.....	14
6 CIMXML format rules and conventions.....	15
6.1 General.....	15
6.2 Simplified RDF syntax.....	16
6.2.1 General.....	16
6.2.2 Notation.....	16
6.2.3 Syntax definition.....	16
6.2.4 Syntax extension for difference model.....	21
6.3 CIMXML format style guide.....	26
6.4 Representing new, deleted and changed objects as CIMXML elements.....	27
6.5 CIM RDF schema generation with CIM profile.....	27
6.6 CIM extensions.....	28
6.7 RDF simplified syntax design rationale.....	28
Bibliography.....	30
Figure 1 – Model with header.....	10
Figure 2 – Example work flow events.....	12
Figure 3 – Example work flow events with more dependencies.....	13
Figure 4 – CIMXML-based power system model exchange mechanism.....	15
Figure 5 – Relations between UML, profile and CIMXML tools.....	28
Table 1 – Header attributes.....	11

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**ENERGY MANAGEMENT SYSTEM APPLICATION
PROGRAM INTERFACE (EMS-API) –****Part 552: CIMXML Model exchange format****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.

International Standard IEC 61970-552 has been prepared by IEC technical committee 57: Power systems management and associated information exchange.

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

FDIS	Report on voting
57/1386/FDIS	57/1402/RVD

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

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

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.

IECNORM.COM : Click to view the full PDF of IEC 61970-552:2013

INTRODUCTION

This International standard is part of the IEC 61970 series that define 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 an electric utility operations. The CIM is described using the Unified Modelling 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. In order for software programs to use the CIM, it must be transformed into a schema form that supports a programmable interface.

IEC 61970-501 describes the translation of the CIM in UML form into a machine readable format as expressed in the Extensible Markup Language (XML) representation of that schema using the Resource Description Framework (RDF) Schema specification language.

IEC 61970-552 specifies how the CIM RDF schema specified in IEC 61970-501 is used to exchange power system models using XML (referred to as CIMXML) defined in the 61970-45x series of profile standards, such as the CIM Transmission Network Model Exchange Profile described in IEC 61970-452.

Withdwn.com : Click to view the full PDF of IEC 61970-552:2013

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

Part 552: CIMXML Model exchange format

1 Scope

This International Standard 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 RDF Schema presented in IEC 61970-501 as the meta-model framework for constructing XML documents of power system modelling information. The style of these documents is called CIMXML format.

Model exchange by file transfer serves many useful purposes. Profile documents such as IEC 61970-452 and other profiles in the 61970-45x series of standards explain the requirements and use cases that set the context for this work. Though the format can be used for general CIM-based information exchange, specific profiles (or subsets) of the CIM are identified in order to address particular exchange requirements. The initial requirement driving the solidification of this specification is the exchange of transmission network modelling information for power system security coordination.

This standard supports a mechanism for software from independent suppliers to produce and consume CIM described modelling information based on a common format. The proposed solution:

- is both machine readable and human readable, although primarily intended for programmatic access,
- can be accessed using any tool that supports the Document Object Model (DOM) and other standard XML application program interfaces,
- is self-describing,
- takes advantage of current World Wide Web Consortium (W3C) recommendations.

This document is the Level 2 Component Interface Specification document that describes in narrative terms (with text and examples based on the CIM) the detailed definition of the CIMXML format.

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 series, *International Electrotechnical Vocabulary*

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

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

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*

W3C: RDF/XML Syntax Specification

W3C: Extensible Markup Language (XML) 1.0

W3C: XSL Transformations (XSLT)

W3C: Document Object Model (DOM)

3 Terms and definitions

For the purposes of this International Standard, the terms and definitions contained in IEC 60050 (for general glossary) and IEC 61970-2 (for EMS-API glossary definitions), as well as the following apply.

3.1

Application Program Interface

API

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, between entire 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

CIMXML

serialisation format for exchange of XML data as defined in this document

3.4

Document Object Model

DOM

platform- and language-neutral interface defined by the World Wide Web Consortium (W3C) that allows programs and scripts to dynamically access and exchange the content, structure and style of documents

3.5

Document Type Definition

DTD

standard for describing the vocabulary and syntax associated with an XML document

Note 1 to entry: XML Schema and RDF are other forms that can be used.

3.6

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 assure adequate security of energy supply at minimum cost

3.7

Hypertext Markup Language

HTML

mark-up language used to format and present information on the Web

3.8

model

collection of data describing objects or entities real or computed

Note 1 to entry: In the context of CIM the semantics of the data is defined by profiles; refer to 3.9.

Note 2 to entry: In power system analysis, a model is a set of static data describing the power system. Examples of Models include the Static Network Model, the Topology Solution, and the Network Solution produced by a power flow or state estimator application.

3.9

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.

3.10

profile document

collection of profiles intended to be used together for a particular business purpose

3.11

Resource Description Framework

RDF

language recommended by the W3C for expressing metadata that machines can process simply

Note 1 to entry: RDF uses XML as its encoding syntax.

3.12

RDF Schema

schema specification language expressed using RDF to describe resources and their properties, including how resources are related to other resources, which is used to specify an application-specific schema

3.13

Real-World Object

objects that belong to the real world problem domain as distinguished from interface objects and controller objects within the implementation

Note 1 to entry: The real-world objects for the EMS domain are defined as classes in IEC 61970-301 Common Information Model.

Note 2 to entry: Classes and objects model what is in a power system that needs to be represented in a common way to EMS applications. A class is a description of an object found in the real world, such as a PowerTransformer, GeneratingUnit, or Load that needs to be represented as part of the overall power system model in an EMS. Other types of objects include things such as schedules and measurements that EMS applications also need to process, analyze, and store. Such objects need a common representation to achieve the purposes of the EMS-API standard for plug-compatibility and interoperability. A particular object in a power system with a unique identity is modeled as an instance of the class to which it belongs.

3.14**Standard Generalized Markup Language****SGML**

international standard for the definition of device-independent, system-independent methods of representing texts in electronic form

Note 1 to entry: HTML and XML are derived from SGML.

3.15**Unified Modeling Language****UML**

object-oriented modeling language and methodology for specifying, visualizing, constructing, and documenting the artifacts of a system-intensive process

3.16**Uniform Resource Identifier****URI**

Web standard syntax and semantic for identifying (referencing) resources (things, such as files, documents, images).

3.17**eXtensible Markup Language****XML**

subset of Standard Generalized Markup Language (SGML), ISO 8879, for putting structured data in a text file

Note 1 to entry: This is an endorsed recommendation from the W3C. It is license-free, platform-independent and well-supported by many readily available software tools.

3.18**eXtensible Stylesheet Language****XSL**

language for expressing stylesheets for XML documents

4 Model exchange header**4.1 General**

Model exchange typically involves the exchange of a collection of documents, each of which contains instance data (referred to as a model) and a header. The structure and semantics of each model are described by a profile, which is not included in the exchanged data. The overall exchange is governed by a collection of profiles in a Profile Document.

A header describes the content of the model contained in a document e.g. the date the model was created, description etc. 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.

Subclauses 4.2 to 4.4 define the model with header data and work flow it is designed to support.

4.2 CIMXML documents and headers

A CIMXML document is described by a single header. Multiple headers in a CIMXML document are not allowed. Hence instance data in a CIMXML document adhere to a single profile.

In case multiple and possibly related CIMXML documents need to be kept together they shall be collected in an archive, e.g. zip.

4.3 Model and header data description

A description of a model is attached as header data to the model. Figure 1 describes the model with header information.

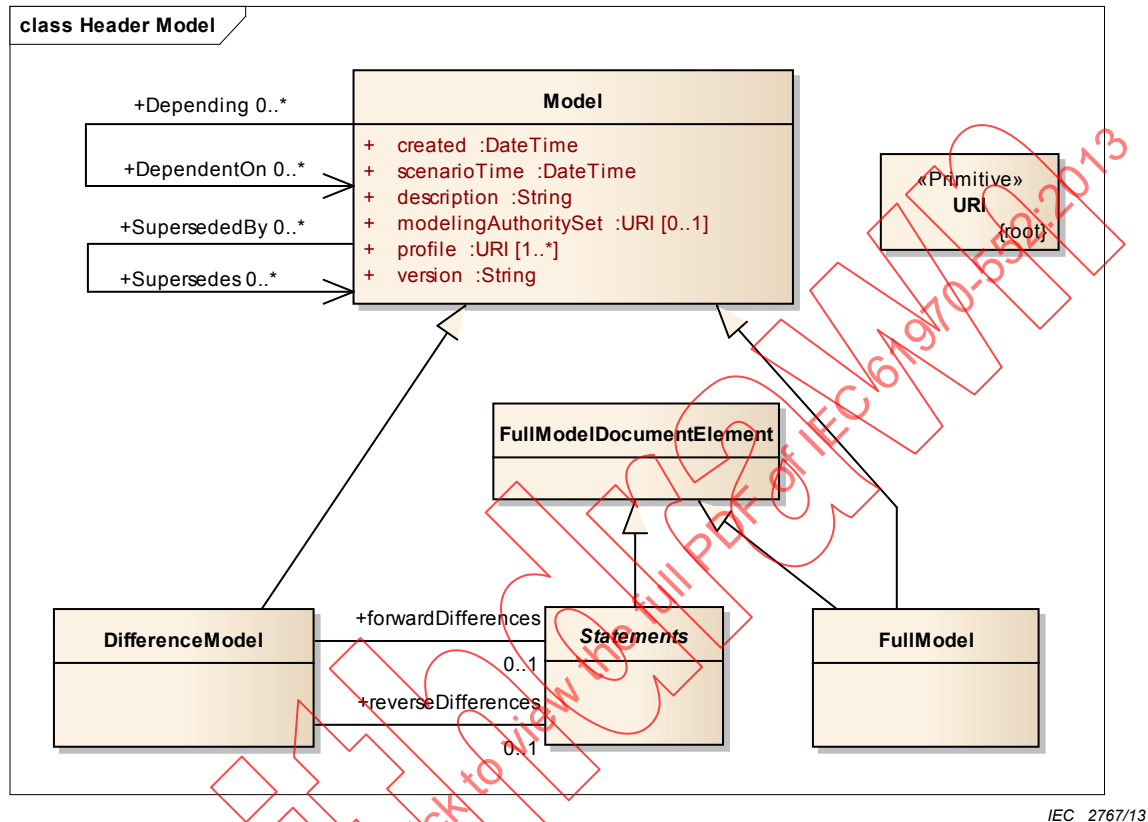


Figure 1 – Model with header

In Figure 1 the classes FullModel, DifferenceModel and Statements describe the model data while the header is described by the classes Model and Description. The following is a bottom up description of these classes:

- The FullModelDocumentElement class represents any of the elements that may appear in a full model document. It has the two sub types Statements or FullModel, both are further described below. A full model document typically contains one FullModel element and one set of Definition elements.
- The Statements class represent a set of Definition (refer to 6.2.3.5) and/or Description (refer to 6.2.3.6) elements.
- The FullModel (refer to 6.2.3.4) class represent the full model header and its contents is described by the Model class.
- The DifferenceModel (refer to 6.2.4.6) class represents the difference model header. The content is described by the Model class, the association role forwardDifferences and association role reverseDifferences. Both association roles may have one set of Statements.
- The Model class describes the header content that is the same for the FullModel and the DifferenceModel. A Model is identified by an rdf:about attribute. The rdf:about attribute uniquely describe the model and not the document where the header exists. Hence multiple documents created from the same unchanged data model will have the same rdf:about. This also means that a model change result in a new rdf:about next time a document is created.

The Model class attributes are described in Table 1.

Table 1 – Header attributes

Class	Attribute	Description
Model	created	The date when the model was created (note this is typically not when the CIMXML document was created which is after this time).
Model	scenarioTime	The date and time that the model represents, e.g. the current time for an operational model, a historical model or a future planned model.
Model	description	A description of the model, .e.g. the name of person that created the model and for what purpose.
Model	modelingAuthoritySet	A URN describing the equipment model sourcing the data in a CIMXML document, e.g. a model for the whole or a part of a country.
Model	profile	A URN describing the Profiles that governs this model. It uniquely identifies the Profile and its version.
Model	version	A description of the version of the model sourcing the data in a CIMXML document. Examples are – Variations of the equipment model for the ModelingAuthoritySet – Different study cases resulting in different solutions. The version attribute is a custom string that is changed in synchronisation with the rdf:about identifier, refer to description of the Model class above.
Model	DependentOn	A reference to the models that the model described by this document depends on, e.g. – A load flow solution depends on the topology model it was computed from – A topology model computed by a topology processor depends on the network model it was computed from.
Model	Depending	All models depending on this model. This role is not intended to be included in any document exchanging instance data.
Model	Supersedes	When a model is updated the resulting model supersedes the models that were used as basis for the update. Hence this is a reference to CIMXML documents describing the updated models.
Model	SupersededBy	All models superseding this model. This role is not intended to be included in any document exchanging instance data.

The profile attribute is a URI having the following format:

<http://iec.ch/<committee>/<year>/<standard>-<part>/<profile>/<version>>

e.g. <http://iec.ch/TC57/2011/61970-452/Equipment/2>

The UML in Figure 1 translates into CIMXML elements as follows:

- 1) A leaf class in Figure 1 (DifferenceModel, Statements and FullModel) appears as class elements under the document element (6.2.3.3).
- 2) Statement elements appear as Definition (6.2.3.5) or Description elements (6.2.3.6).
- 3) Literal attributes, e.g. Model.created, appears as literal property elements (6.2.3.8).
- 4) Roles appear, e.g. Model.Supersedes, as resource property elements (6.2.3.10).
- 5) Inherited attributes and roles appear directly as elements under the leaf class following the rules 3, 4 and 5 above.
- 6) A CIMXML model document is identified by a Model rdf:about attribute (implicit in the UML). Hence the roles DependentOn and Supersedes are references to the Model rdf:about attribute.

- 7) A full model document may be regenerated multiple times from the same source data. Full model documents regenerated from unchanged source data keep the model identification (Model rdf:about) unchanged from the original full model document.
- 8) When generating a full model document superseding a differential the new full model document will have the same model identification (Model rdf:about) as the differential if the model is unchanged since the differential was created. Hence it is an alternate to the differential.

4.4 Work flow

A work flow is described by a sequence of exchange events. The model description in 4.3 supports work flow events related in time with the Model.Supersedes attribute and events related to profiles with the Model.DependentOn attribute. An example of this is shown in Figure 2.

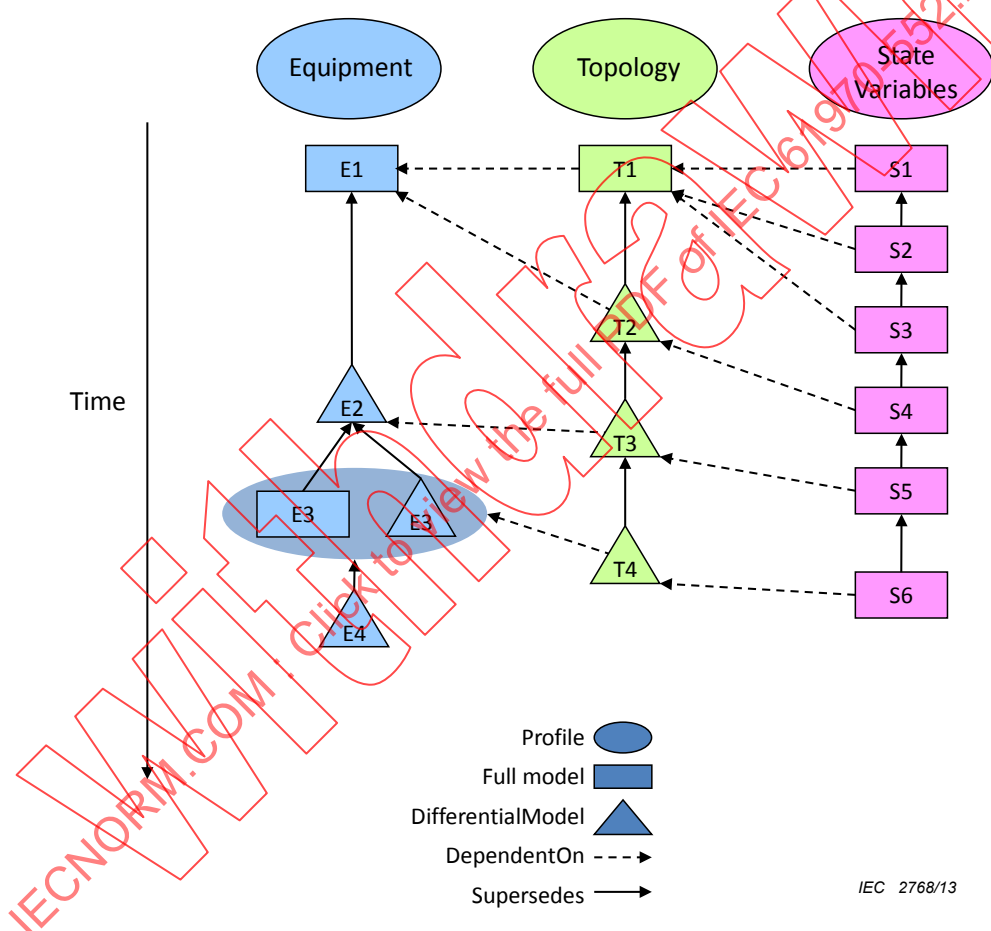


Figure 2 – Example work flow events

In this example, a solved network model is exchanged as a collection of models governed by a Profile Document comprising Equipment, Topology, and State Variables documents. The left time line in Figure 2 represents how the Equipment model document is exchanged over time. The center time line shows how new Topology results are exchanged over time and the Equipment models on which each depends. The right most time line shows how multiple State Variable documents are exchanged and the Topology documents on which they depend. Also note that the equipment model E3 is represented both by a full and an incremental document. The situation in Figure 2 represents a simple case. A more complex situation is shown in Figure 3.

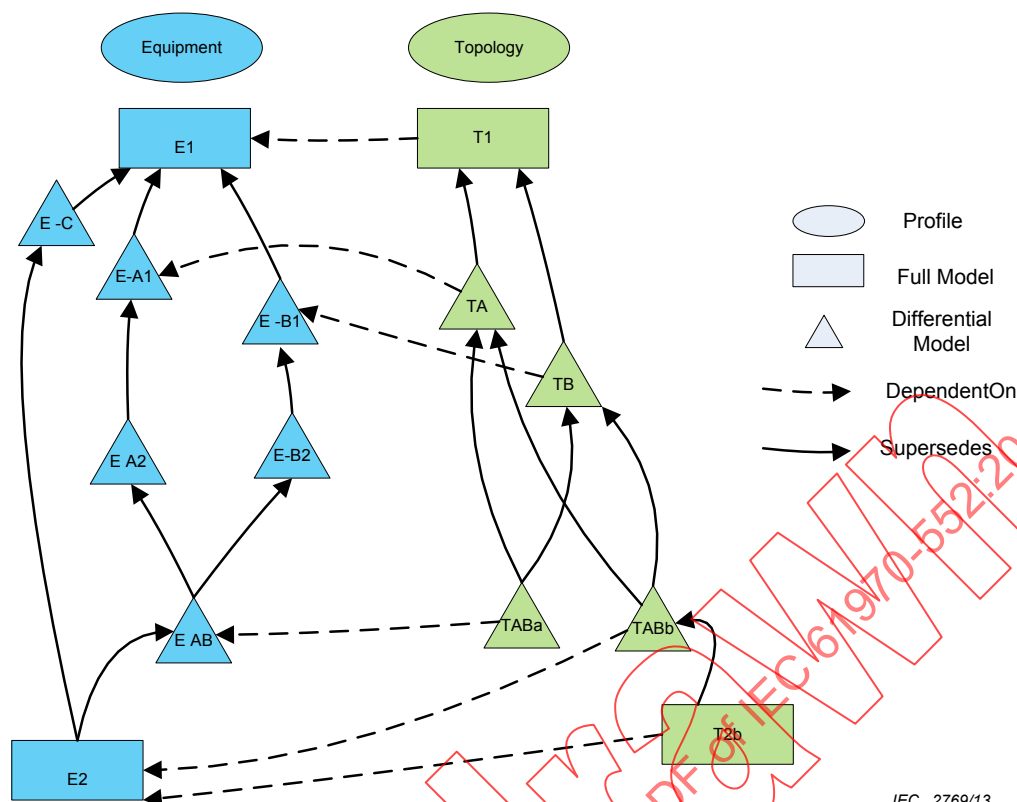


Figure 3 – Example work flow events with more dependencies

The CIMXML documents in Figure 3 may be created from a data modeller environment where multiple change tracks of a model appear in parallel, e.g. the equipment model has three tracks E-Ax, E-Bx and E-C that eventually merge into the full model E2 superseding the equipment model tracks.

A receiver of the CIMXML documents may use any of the topology documents TA, TB, TABa or T2b with the equipment model from E2. As the sender (the data modeller in this example) only verified T2b with E2 this is the only combination that is supposed to fit together. Concerning T2b the receiver may choose to apply TB and TABb to T1 instead of using T2b.

5 Object identification

5.1 URIs as identifiers

UUIDs (Universally Unique Identifier), also known as GUIDs (Globally Unique Identifier) can be used to identify resources in such a way that the

- identifiers can be independently and uniquely allocated by different authorities. This is a big advantage with the UUID.
- identifiers are stable over time and across documents.

If, in addition, the UUID is embedded in a Uniform Resource Name (URN) then the document can be simplified by the elimination of XML base namespace declarations (xml:base attributes). The URN is a concise, fixed-length, absolute URI.

The standard for an URN containing a UUID is defined by the Internet Engineering Task Force RFC 4122.

RFC 4122 specifies the syntax of the URN and how the UUID portion following the last colon is allocated. The algorithm is aligned with, and technically compatible with, IEC 9834-8:2004 Information Technology, "Procedures for the operation of OSI Registration Authorities: Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 Object Identifier components" ITU-T Rec. X.667, 2004.

CIMXML elements are identified by a URI. A URI can have two forms:

- URL
- URN

The URL and URN forms have fundamentally different structures, i.e.:

- URL form: *protocol://authority/path?query#fragment* where the *protocol* in CIMXML is http
- URN form: *urn:namespace:specification* where the *namespace* in CIMXML is uuid.

The URN *specification* format is summarized below

- 8 character hex number
- a dash “-”
- 4 character hex number
- a dash “-”
- 4 character hex number
- a dash “-”
- 4 character hex number
- a dash “-”
- 12 character hex number

where letters are lower case.

An example of the URN form is shown below

- "urn:uuid:26cc8d71-3b7e-4cf8-8c93-8d9d557a4846".

5.2 About rdf:ID and rdf:about

A CIMXML element can be identified by two different RDF constructs:

- rdf:ID
- rdf:about

The use of rdf:ID and rdf:about has a specific meaning that does not align with their definition in RDF. The meanings are:

- an rdf:ID specifies the life of object globally, i.e. its creation or deletion.
- an rdf:about is a reference to an existing object.

5.3 CIMXML element identification

Object identification is so central in RDF that all elements representing objects are identified with a rdf:ID or rdf:about XML attribute. All classes in CIM that inherit IdentifiedObject have the UML object identification attribute IdentifiedObject.mRID. The attribute is implicitly mapped to the rdf:ID/rdf:about XML attribute.

A CIMXML document may only use the URN form (see 5.1) as further described below.

CIMXML files contain XML elements describing CIM objects (ACLineSegments, Substations etc.). The CIM has lots of association roles that show up as references in the XML elements (typically as `rdf:resource` or `rdf:about` attributes). CIM data is exchanged in different CIMXML documents that depend on each other as described in Clause 4. Some references then cross CIMXML document boundaries. A consequence of this is that the identification of a CIM object must be stable during its life time. Otherwise referencing objects across document boundaries will break.

A common practice in object oriented systems is to assume all objects have an identifier that is unique in space and time which means:

- Different objects are assigned different identifiers.
- Identifiers once assigned are never reused even if the original object having it is gone.

The URN form as described in 5.1 is used as CIMXML element identification with the following differences

- the prefix “urn:uuid:” is replaced by an underscore “_”. The underscore avoids a numeric starting character for the non-base part of the identifier. Starting the non-base part of the identifier with a numeric character is invalid RDF. The underscore is added in all cases to simplify parsers, even if the UUID starts with a non-numeric character.
- the prefix is defined as an `xml:base=“urn:uuid:”`

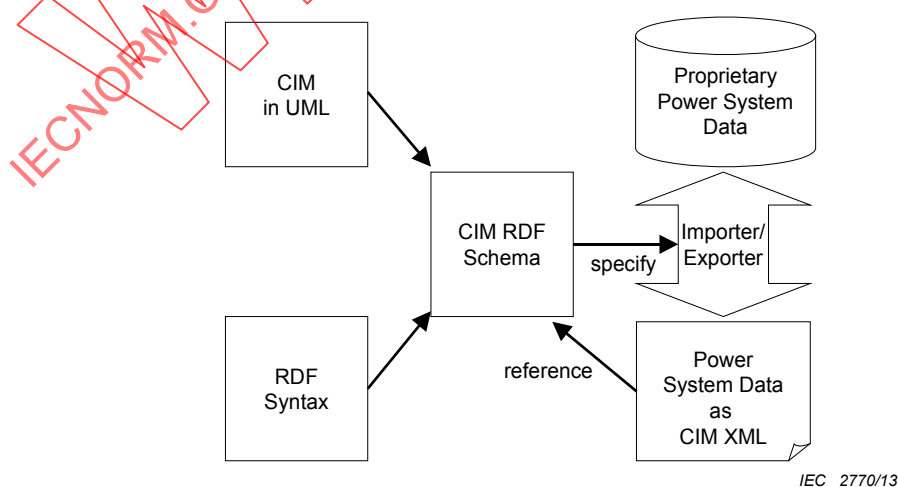
Some examples:

- `rdf:ID=“_26cc8d71-3b7e-4cf8-8c93-8d9d557a4846”`
- `rdf:about=“#_26cc8d71-3b7e-4cf8-8c93-8d9d557a4846”`.

6 CIMXML format rules and conventions

6.1 General

Given the CIM RDF Schema described in IEC 61970-501, a power system model can be converted for export as an XML document (see Figure 3). This document is referred to as a CIMXML document. All of the tags (resource descriptions) used in the CIMXML document are supplied by the CIM RDF schema. The resulting CIMXML model exchange document can be parsed and the information imported into a foreign system.



IEC 2770/13

Figure 4 – CIMXML-based power system model exchange mechanism

6.2 Simplified RDF syntax

6.2.1 General

RDF syntax provides many ways to represent the same set of data. For example, an association between two resources can be written with a resource attribute or by nesting one element within another. This could make it difficult to use some XML tools, such as XSL processors, with the CIMXML document.

Therefore, only a subset of the RDF Syntax is to be applied in creating CIMXML documents. This syntax simplifies the work of implementers to construct model serialization and de-serialization software, as well as to improve the effectiveness of general XML tools when used with CIMXML documents. The reduced syntax is a proper subset of the standard RDF syntax; thus, it can be read by available RDF de-serialization software.

The following subsections define a subset of the RDF Syntax. This simplified syntax is for exchanging power system models between utilities. The aim of the specification is to make it easier for implementers to construct de-serialization software for RDF data, to simplify their choices when serializing RDF data, and to improve the effectiveness of general XML tools such as XSLT processors when used with the serialized RDF data.

The reduced syntax is a proper subset of the standard RDF syntax. Thus, it can be read by RDF de-serialization software such as SirPAC [8]1. In this, it differs from other proposals for a simplified syntax, such as [9], [10].

The reduced syntax does not sacrifice any of the power of the RDF data model. That is, any RDF data can be exchanged using this syntax. Moreover, features of RDF such as the ability to extend a model defined in one document with statements in second document are preserved.

6.2.2 Notation

The simplified syntax is defined in the following section. Each kind of element is defined in a subsection beginning with a model of the element, followed by some defining text, and a reference to the RDF grammar. The semantics of the element are not detailed (refer to the RDF recommendation [3] for that information). The notation for the element model is as follows:

- a) A symbol in italics in the position of an element type, attribute name or attribute value indicates the type of name or value required. The symbol will be defined in the text.
- b) The symbol *rdf* stands for whatever namespace prefix is chosen by the implementation for the RDF namespace. Similarly the symbol *cim* stands for the chosen CIM namespace prefix.
- c) A comment within the element model indicates the allowed content. A symbol in italics stands for a kind of element or other content defined in the text. A construction (a | b) indicates that a and b are alternatives. A construction a* indicates zero or more repetitions of a.
- d) All other text in the model is literal.

6.2.3 Syntax definition

6.2.3.1 General

The syntax definition is enriched with examples. The examples should help to get a better understanding of the formal syntax definition. The same example is used for several syntax definitions. The syntax focused in the example is indicated in bold.

¹ Numbers in square brackets refer to the bibliography.

6.2.3.2 Name space URIs defined in this specification

The following name spaces are defined in this specification:

- `cim-model-description_uri` described by `xmlns:md`
- `difference-model-namespace-uri` described by `xmlns:dm`

Their values are defined as:

- `xmlns:md`="http://iec.ch/TC57/61970-552/ModelDescription/1#"
- `xmlns:dm`="http://iec.ch/TC57/61970-552/DifferenceModel/1#"

6.2.3.3 Document element

```
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:cim="cim-namespace-uri"
  xmlns:md="cim-model-description-uri">
  <!-- Content: full-model (definition|description)* -->
</rdf:RDF>

<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:cim="cim-namespace-uri"
  xmlns:md="cim-model-description-uri"
  xml:base="urn:uuid:">
  <!-- Content: full-model (definition|description)* -->
</rdf:RDF>
```

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cim="http://iec.ch/TC57/2004/CIM-schema-cim10#"
  xmlns:md="http://iec.ch/TC57/61970-552/ModelDescription/1#"
  xml:base="urn:uuid:">
</rdf:RDF>
```

- The element type is `rdf:RDF`.
- The RDF namespace must be declared as `http://www.w3.org/1999/02/22-rdf-syntax-ns#`
- The CIM namespace must be declared. With newer versions of the CIM schema the version needs to be adjusted in the CIM name space. Parties exchanging documents have to agree on the used version.
- Other namespaces may be declared.
- The `xml:base` attribute shall always be present, refer to 5.3.

The RDF [3] grammar clause: 6.1.

6.2.3.4 FullModel element

```
<md:FullModel rdf:about=model-uri>
  <!-- Content: (literal-property|resource-property
  | compound-property)* -->
</md:FullModel >
```

Example:

```
<md:FullModel rdf:about="urn:uuid:26cc8d71-...">
  <md:Model.created>2008-12-24</md:Model.created>
  <md:Model.Supersedes rdf:resource="urn:uuid:26cc8d71-a002-4c2b-bcf4-
  7bc97430bf87"/>
```

```
<md:Model.DependentOn rdf:resource=#_26cc8d71-a002-4c2b-bcf4-7bc97430bf88"/>
<md:Model.version>V32</md:Model.version>
<md:Model.modelingAuthoritySet>http://polarenergy.com/2008/NorthPoleTSO</md:Model.modelingAuthoritySet>
<md:Model.description>Santa Claus made a study case peak load summer base topology solution</md:Model.description>
<md:Model.profile>http://iec.ch/TC57/61970-456/StateVariables/1</md:Model.profile>
</md:FullModel>
```

- 1) The full model element introduces a new model.
- 2) The value of the about attribute, model-uri, is a name chosen by the implementation. The model-uri uniquely identifies a document and is the name referenced by other documents, e.g. by Supersedes or DependentOn, as indicated in Figure 2.

6.2.3.5 Definition element

```
<classname rdf:ID=identity>
  <!-- Content:
    (literal-property|resource-property|compound-property) *
  -->
</classname>

<classname rdf:about=resource-uri>
  <!-- Content:
    (literal-property|resource-property|compound-property) *
  -->
</classname>
```

Example:

```
<cim:SynchronousMachine rdf:about="#_31dcf429-6bfb-4e2e-b2996-42491b3abc1">
  <cim:IdentifiedObject.name>IN-2</cim:IdentifiedObject.name>
  <cim:SynchronousMachine.minimumMVar>-9999</cim:SynchronousMachine.minimumMVar>
  <cim:SynchronousMachine.operatingMode rdf:resource="http://iec.ch/TC57/2001/CIM-schema-cim10#SynchronousMachineOperatingMode.generator"/>
  <cim:RegulatingCondEq.RegulationSchedule rdf:resource="#_ca32746f-a002-4c2b-bcf4-7bc97430bf87"/>
  <cim:Equipment.EquipmentContainer rdf:resource="#_6cb8701a-12f1-4de9-9e68-125d95073a75"/>
</cim:SynchronousMachine>
```

- a) The definition element introduces a new resource and gives its type. There are two forms: the first as an rdf:ID attribute and the second an rdf:about attribute.
- b) The element type, *classname*, is the XML qualified name of a class from the CIM schema or other schema declared as a namespace in the document element.
- c) The value of the id attribute, *identity*, is chosen by the implementation. It must be unique in the document. (It is not necessarily related to the power system resource name.)

6.2.3.6 Description element

```
<rdf:Description rdf:about=resource-uri>
  <!-- Content:
    (literal-property|resource-property|compound-property) *
  -->
</rdf:Description >
```

Example:

```
<rdf:Description rdf:about="#_26cc8d71-a002-4c2b-bcf4-7bc97430bf87">
  <cim:IdentifiedObject.name>TROY</cim:IdentifiedObject.name>
```

</rdf:Description>

- a) The description element adds information about a resource introduced elsewhere in this or another document.
- b) The *resource-uri* is a URN-reference that identifies the subject resource.
- c) The Description element is used only in difference models (refer to 6.2.4). It is never used in full models.

6.2.3.7 Compound element

```
<classname>
  <!-- Content:
    (literal-property|resource-property|compound-property)*
  -->
</classname>
```

Example:

```
<cim:DateTimeInterval>
  <cim:DateTimeInterval.start>2013-02-28</cim:DateTimeInterval.start>
  <cim:DateTimeInterval.end>2013-02-29</cim:DateTimeInterval.end>
</cim:DateTimeInterval>
```

- 1) The compound element introduces a structured value. The value does not represent a resource nor have any *identity*. It can only appear as the object of a property.
- 2) The element type, *classname*, is the XML qualified name of a compound class.
- 3) A compound element is treated as an indivisible unit. Hence a compound element is not supposed to be split in multiple elements having different sets of members. Refer also to paragraph 6.2.4.7.4.

6.2.3.8 Literal-Property element

```
<propname>
  <!-- Content: text -->
</propname>
```

Example:

```
<cim:SynchronousMachine rdf:ID=" 31dcf429-6Bfb-4e2e-b2996-42491b3abc1">
  <cim:IdentifiedObject.name>IN-2</cim:IdentifiedObject.name>
  <cim:SynchronousMachine.minimumMVar>-9999</cim:SynchronousMachine.minimumMVar>
  <cim:SynchronousMachine.operatingMode rdf:resource="http://iec.ch/TC57/2001/CIM-schema-cim10#SynchronousMachineOperatingMode.generator"/>
  <cim:RegulatingCondEq.RegulationSchedule rdf:resource="#_ca32746f-a002-4c2b-bcf4-7bc97430bf87"/>
  <cim:Equipment.EquipmentContainer rdf:resource="#_6cb8701a-12f1-4de9-9e68-125d95073a75"/>
</cim:SynchronousMachine>
```

- a) The literal-property element introduces a property and a literal value applying to the enclosing resource.
- b) The element type, *propname*, is the XML qualified name of a property from the CIM schema or other schema declared as a namespace in the document element.
- c) The content *text* is any XML text with <, >, and & escaped representing the value of the property.
- d) Floating point numbers may slightly change due to rounding effects when imported and re-exported again. This is allowed and need to be managed by applications, e.g. by use of a dead band in case the values are compared.

6.2.3.9 Compound-Property element

```
<propname>
  <!-- Content: (compound) -->
</propname>
```

Example:

```
<cim:TimeSchedule>
  <cim:TimeSchedule.scheduleInterval>
    <cim:DateTimeInterval>
      <cim:DateTimeInterval.start>2013-02-28</cim:DateTimeInterval.start>
      <cim:DateTimeInterval.end>2013-02-29</cim:DateTimeInterval.end>
    </cim:DateTimeInterval>
  </cim:TimeSchedule.scheduleInterval>
</cim:TimeSchedule>
```

6.2.3.10 Resource-Property element

```
<propname rdf:resource=resource-uri/>
```

- The resource-property element introduces a property and a resource as its value applying to the enclosing resource.
- The element type, *propname*, is the XML qualified name of a property from the CIM schema or other schema declared as a namespace in the document element.
- The *resource-uri* is an URN-reference that identifies a resource.
- For relations with roles having cardinality greater than one the resource property element shall be repeated as many times as there are references

Example 1 - URN-Reference:

The example contains two references one for a RegulationSchedule and the other to the parent represented as EquipmentContainer.

```
<cim:SynchronousMachine rdf:ID="_31dcf429-6bfb-4e2e-b299-642491b3abc1">
  <cim:IdentifiedObject.name>IN-2</cim:IdentifiedObject.name>
  <cim:SynchronousMachine.minimumMVar>-9999</cim:SynchronousMachine.minimumMVar>
  <cim:SynchronousMachine.operatingMode rdf:resource="http://iec.ch/TC57/2001/CIM-
schema-cim10#SynchronousMachineOperatingMode.generator"/>
  <cim:RegulatingCondEq.RegulationSchedule rdf:resource="#_cd32746f-a002-4c2b-
bcf4-7bc97430bf87"/>
  <cim:Equipment.EquipmentContainer rdf:resource="#_6cb8701a-12f1-4de9-9e68-
125d95073a75"/>
</cim:SynchronousMachine>
```

Example 2 - Enumeration:

The example defines the attribute value of SynchronousMachine.operatingMode as "generator". The operatingMode is specified in the CIM schema as the enumeration SynchronousMachineOperatingMode.

```
<cim:SynchronousMachine rdf:ID="_31dcf429-6bfb-4e2e-b2996-42491b3abc1" >
  <cim:IdentifiedObject.name>IN-2</cim:IdentifiedObject.name>
  <cim:SynchronousMachine.minimumMVar>-9999</cim:SynchronousMachine.minimumMVar>
  <cim:SynchronousMachine.operatingMode rdf:resource="http://iec.ch/TC57/2001/CIM-
schema-cim10#SynchronousMachineOperatingMode.generator"/>
  <cim:RegulatingCondEq.RegulationSchedule rdf:resource="#_cd32746f-a002-4c2b-
bcf4-7bc97430bf87"/>
  <cim:Equipment.EquipmentContainer rdf:resource="#_6cb8701a-12f1-4de9-9e68-
125d95073a75"/>
</cim:SynchronousMachine>
```


The example defines the attribute value of `SynchronousMachine.operatingMode` as “generator”. The `operatingMode` is specified in the CIM schema as the enumeration `SynchronousMachineOperatingMode`.

Example 3 – Role with cardinality greater than one:

```
<cim:SynchronousMachine rdf:ID="_31dcf429-6bfb-4e2e-b299-642491b3abc1">
  <cim:IdentifiedObject.name>IN-2</cim:IdentifiedObject.name>
  <cim:SynchronousMachine.minimumMVar>-9999</cim:SynchronousMachine.minimumMVar>
  <cim:SynchronousMachine.operatingMode rdf:resource="http://iec.ch/TC57/2001/CIM-schema-cim10#SynchronousMachineOperatingMode.generator"/>
  <cim:RegulatingCondEq.RegulationSchedule rdf:resource="#_cd32746f-a002-4c2b-bcf4-7bc97430bf87"/>
  <cim:Equipment.EquipmentContainer rdf:resource="#_6cb8701a-12f1-4de9-9e68-125d95073a75"/>
  <cim:Equipment.ReactiveCapabilityCurves rdf:resource="#_6cb8701a-12f1-4de9-9e68-125d95073a76"/>
  <cim:Equipment.ReactiveCapabilityCurves rdf:resource="#_6cb8701a-12f1-4de9-9e68-125d95073a77"/>
  <cim:Equipment.ReactiveCapabilityCurves rdf:resource="#_6cb8701a-12f1-4de9-9e68-125d95073a78"/>
</cim:SynchronousMachine>
```

6.2.4 Syntax extension for difference model

6.2.4.1 General

The general syntax definition in the first part of this clause is used for partial and full model data exchange. Once the initial complete set of model data is exchanged, only updates are required to maintain the model as changes occur. In general, those changes can be specified as a set of differences between two models. The difference document is itself an RDF model (a collection of RDF statements) and therefore can be processed by an RDF infrastructure.

6.2.4.2 Example use case

To illustrate the difference document approach to handling incremental model updates, an example use case is provided. In this example, the participants are Regional Energy Co. and Network Power Co.:

- Each participant has a copy of a power system model, B1.
- Regional Energy Co. updates B1, to reflect forthcoming power system modifications, producing B2.
- Regional Energy Co. sends the differences between B1 and B2 to Network Power Co. as a difference model.
- Network Power Co. reviews and validates the difference model.
- Network Power Co. merges the difference model with its copy of model B1, to produce B2.

An alternative would have been for Regional Energy Co. to simply send Network Power Co. a copy of B2. However, B2 is a very large model and it is not feasible to validate it in any reasonable period of time. Validation is not entirely automated, but involves analysis by experts. Indeed, the best validation strategy for B2 may be to compare it to the previously validated B1. This brings us back to the need for a difference model.

A more complicated use case would involve more than two participants. Several peers of Regional Energy Co. would contribute difference models to Network Power Co. This use case would introduce issues of parallel model changes and concurrency conflict.

6.2.4.3 Requirements

Given two RDF models, B1 and B2, called base models, the requirement is for a difference model that:

- Represents the differences between the two base models.
- Is itself an RDF model (a collection of RDF statements) and therefore can be processed by RDF infrastructure.
- Efficiently represents a small difference between two large base models.
- When an object is deleted, the system applying the differences is responsible for performing the “cascading deletions” ,i.e, finding and deleting all other contained objects. Associations with deleted objects should also be deleted.
- Remove operations are not reversible (at least, not from the information in the difference model).
- May contain information about itself such as authorship, purpose and date.
- May contain information to protect against conflicts arising when two difference models are created concurrently from the same base model.

The requirement to treat each difference document as a database commit operation is outside the scope of this service (i.e., a roll back functionality, if desired, is the responsibility of the receiving application, not the sending application). This is in recognition of the fact that the sending application may not be aware of changes made in the B2 model documents by other agents since the last update to B1.

6.2.4.4 Structure of difference document

Given two base RDF models, B1 and B2, the difference model is made up of four groups of statements, each encoded as a sequence of resource description structures:

- Header statements, comprising statements about the difference model itself.
- Forward difference statements, comprising statements found in B2, but not in B1.
- Reverse difference statements, comprising statements found in B1, but not in B2.
- Precondition statements, comprising statements found in both B1 and B2 and considered to be dependencies of the difference model in an application defined sense.

Any or all of the four groups can be empty.

The difference model itself is represented by a resource of type `dm:DifferenceModel`. It is conventional to use the URN of the model itself for this resource.

The following properties apply to the difference model resource:

- `dm:forwardDifferences` is a property of the difference model whose value is a collection of statements (i.e., resources of type `rdf:Statement`) representing the forward difference statements.
- `dm:reverseDifferences` is a property of the difference model whose value is the collection of reverse difference statements.
- `dm:preconditions` is a property of the difference model whose value is the collection of precondition statements.

Header properties also apply to the difference model resource. These may indicate authorship, date and purpose. These properties can be drawn from the Dublin Core vocabulary or any other convenient schema.

The namespace for the difference model vocabulary, represented by the prefix `dm:` in the foregoing, is: <http://iec.ch/TC57/61970-552/DifferenceModel/1#>.

6.2.4.5 Preconditions and concurrency

The precondition statements are a subset of both B1 and B2 and carry no difference information. In simple, sequential model revision scenarios they can be omitted.

For a large shared model, sequential revision is not always feasible. Revisions are likely to be constructed concurrently by different participants, without reference to each other. Concurrency issues must be handled, but the conventional database-oriented approach of using locks to detect incompatible concurrent transactions is not feasible on a web-scale.

The precondition statements are an alternative to locks. Informally, they represent the information that would have been read-locked in an equivalent database transaction. Software agents that process difference models can check that the preconditions hold and, if not, warn of incompatible model revisions.

The choice of statements to include as preconditions is application-specific (as is the choice of which information to lock in a database transaction). Preconditions should include statements that would affect decisions of the agent that produced the model revision.

6.2.4.6 Difference model template

The following is a template for the conventional syntax of a difference model.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cim="cim-namespace-uri"
  xmlns:md="cim-model-description-uri"
  xmlns:dm="difference-model-namespace-uri"
  xml:base="urn:uuid:">
  <dm:DifferenceModel rdf:about=model-uri>
    <!-- Content: (literal-property|resource-property|compound-property) * -->
    -->
    <dm:preconditions parseType="Statements"
    ">
    <!-- Content: (definition|description) * -->
    </dm:preconditions>
    <dm:forwardDifferences parseType="Statements">
    ">
    <!-- Content: (definition|description) * -->
    </dm:forwardDifferences>
    <dm:reverseDifferences parseType="Statements">
    ">
    <!-- Content: (definition|description) * -->
    </dm:reverseDifferences>
  </dm:DifferenceModel>
</rdf:RDF>
```

Simply for clarification with the namespace "dm" new statements are introduced that are valid extensions to the standard RDF syntax through the new property `rdf:parseType`, which is called Statements.

```
<property parseType="Statements">
  <!-- Content: (definition|description) * -->
</property>
```

The content model of an element with `rdf:parseType="Statements"` is the same as the content model of the `rdf:RDF` element.

The content generates the same RDF statements as if it appeared in an `rdf:RDF` element.

6.2.4.7 Difference model usage

6.2.4.7.1 General

The following cases explain the usage of the difference model.

6.2.4.7.2 Add resource

The difference model contains for a given resource only a forward difference statement if the particular resource is added.

EXAMPLE:

The following example adds two new ACLineSegments each with its adjacent Terminals. The Terminals are linked to new ConnectivityNodes. Those ConnectivityNodes are assigned to a new VoltageLevel in an existing Substation.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cim="cim-namespace-uri"
xmlns:md="cim-model-description-uri"
xmlns:dm="difference-model-namespace-uri"
xml:base="urn:uuid:">
  <dm:DifferenceModel rdf:about="#_26cc8d71-3b7e-4cf8-8c93-8d9d557a4846">
    <md:Model.created>2008-12-24</md:Model.created>
    <md:Model.Supersedes rdf:resource="#_26cc8d71-3b7e-4cf8-8c93-8d9d557a4847"/>
    <md:Model.DependentOn rdf:resource="#_26cc8d71-3b7e-4cf8-8c93-8d9d557a4848"/>
    <md:Model.version>V32</md:Model.version>

    <md:Model.modelingAuthoritySet>http://polarenergy.com/2008/NorthPoleTSO</md:Model.modelingAuthoritySet>
    <md:Model.description>Santa Claus made a study case peak load summer base topology solution</md:Model.description>
    <md:Model.profile>http://iec.ch/TC57/61970-452/EquipmentModel/1</md:Model.profile>
    <dm:forwardDifferences rdf:parseType="Statements">
      <!-- Add ACLineSegment ACLine_New1 -->
      <cim:ACLineSegment rdf:ID="_26cc8d71-12f1-4de9-9e68-125d95073a75">
        <cim:IdentifiedObject.name>New 1</cim:IdentifiedObject.name>
        <cim:Conductor.r>0.0646</cim:Conductor.r>
        <cim:Conductor.x>0.5961</cim:Conductor.x>
        <cim:Conductor.bch>0.4066</cim:Conductor.bch>
      </cim:ACLineSegment>
      <cim:Terminal rdf:ID="_26cc8d71-12f1-4de9-9e68-125d95073a75">
        <cim:IdentifiedObject.name>T1</cim:IdentifiedObject.name>
        <cim:Terminal.ConnectivityNode rdf:resource="#_26cc8d71-12f1-4de9-9e68-125d95073a75"/>
        <cim:Terminal.ConductingEquipment rdf:resource="#_26cc8d71-12f1-4de9-9e68-125d95073a75"/>
      </cim:Terminal>
      <cim:Terminal rdf:ID="_26cc8d71-12f1-4de9-9e68-125d95073a76">
        <cim:IdentifiedObject.name>T2</cim:IdentifiedObject.name>
        <cim:Terminal.ConnectivityNode rdf:resource="#_26cc8d71-12f1-4de9-9e68-125d95073a75"/>
        <cim:Terminal.ConductingEquipment rdf:resource="#_26cc8d71-12f1-4de9-9e68-125d95073a75"/>
      </cim:Terminal>
      <!-- Add ACLineSegment ACLine_New2 -->
      <cim:ACLineSegment rdf:ID="_26cc8d71-12f1-4de9-9e68-125d95073a75">
        <cim:IdentifiedObject.name>New 2</cim:IdentifiedObject.name>
        <cim:Conductor.r>0.0646</cim:Conductor.r>
        <cim:Conductor.x>0.5961</cim:Conductor.x>
        <cim:Conductor.bch>0.4066</cim:Conductor.bch>
      </cim:ACLineSegment>
      <cim:Terminal rdf:ID="_26cc8d71-12f1-4de9-9e68-125d95073a75">
        <cim:IdentifiedObject.name>T1</cim:IdentifiedObject.name>
        <cim:Terminal.ConnectivityNode rdf:resource="#_26cc8d71-12f1-4de9-9e68-125d95073a75"/>
        <cim:Terminal.ConductingEquipment rdf:resource="#_26cc8d71-12f1-4de9-9e68-125d95073a75"/>
      </cim:Terminal>
      <cim:ConnectivityNode rdf:ID="_26cc8d71-12f1-4de9-9e68-125d95073a75">
        <cim:IdentifiedObject.name>ND New1</cim:IdentifiedObject.name>
        <cim:ConnectivityNode.EquipmentContainer rdf:resource="#_26cc8d71-12f1-4de9-9e68-125d95073a75"/>
        <cim:ConnectivityNode.Terminals rdf:resource="#_26cc8d71-12f1-4de9-9e68-125d95073a75"/>
      </cim:ConnectivityNode>
      <cim:Terminal rdf:ID="_26cc8d71-12f1-4de9-9e68-125d95073a75">
        <cim:IdentifiedObject.name>T2</cim:IdentifiedObject.name>
```

```

    <cim:Terminal.ConnectivityNode rdf:resource="#_26cc8d71-..."/>
    <cim:Terminal.ConductingEquipment rdf:resource="#_26cc8d71-12f1-4de9-
9e68-125d95073a75"/>
  </cim:Terminal>
  <cim:ConnectivityNode rdf:ID="_26cc8d71-12f1-4de9-9e68-125d95073a75">
    <cim:IdentifiedObject.name>ND New2</cim:IdentifiedObject.name>
    <cim:ConnectivityNode.EquipmentContainer rdf:resource="#_26cc8d71-
12f1-4de9-9e68-125d95073a75"/>
    <cim:ConnectivityNode.Terminals rdf:resource="#_26cc8d71-12f1-4de9-
9e68-125d95073a75"/>
  </cim:ConnectivityNode>
  <cim:VoltageLevel rdf:ID="_26cc8d71-12f1-4de9-9e68-125d95073a75">
    <cim:IdentifiedObject.name>230K</cim:IdentifiedObject.name>
    <cim:VoltageLevel.Substation rdf:resource="#_26cc8d71-12f1-4de9-9e68-
125d95073a75"/>
    <cim:VoltageLevel.BaseVoltage rdf:resource="#_26cc8d71-12f1-4de9-
9e68-125d95073a75"/>
  </cim:VoltageLevel>
</dm:forwardDifferences>
</dm:DifferenceModel>
</rdf:RDF>

```

6.2.4.7.3 Delete resource

The difference model contains for a given resource only a reverse difference statement if the particular resource is deleted.

Cascading deletes are deletes where an object and its child objects (if any) are deleted. In a cascading delete it would be possible to just include the root or parent object in a CIMXML document. The receiver then has to figure out what child objects to delete. To make clear what objects are included in a cascading delete the creator of the CIMXML document shall include all objects as elements in the cascade. Including only the root or parent object is not allowed.

The EquipmentContainer-Equipment relation is a parent-child relation where deletion of an EquipmentContainer shall also result in a deletion of its child Equipment. Other examples of such parent child relations are

- EquipmentContainers also has a parent child relation, e.g. Station-VoltageLevel
- PowerTransformer and its TransformerEnds
- ConductingEquipment and its Terminals

The CIM does not currently specify the containment relations. As this information is missing it is up to an implementer to decide which relation is regarded a containment relation. This spoils interoperability. This is the reason to include all objects in a cascaded delete to indicate the sending systems interpretation of containment.

Associations not considered a containment relation are cut from objects that that are in a cascading delete, e.g. if a ConnectivityNode is not affected by a delete but a ConductingEquipment connected to it is, then the association Terminal-ConductingEquipment is cut. This means that if the reference to a cut object is from an object that stays the reference from that objects shall be removed which means an update of the object that stays.

Delete elements shall have all its property elements included. The reason is that this enables reversing the delete operation and recreates the object.

EXAMPLE:

The example below contains the deletion of a PowerTransformer with all resources that are hierarchically subordinated.

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cim="cim-namespace-uri"
xmlns:dm="difference-model-namespace-uri"
xml:base="urn:uuid:">
  <dm:DifferenceModel rdf:about="#_26cc8d71-12f1-4de9-9e68-125d95073a75">

```

```
<!-- header properties omitted for brevity -->
<!-- Delete Transformer -->
<dm:reverseDifferences rdf:parseType="Statements" >
  <cim:PowerTransformer rdf:ID="_41bb4445-6756-43fa-9e5a-48B6cd71790e">
    ...all properties of the transformer follows here...
  </cim:PowerTransformer>
  ...all parts of the transformer follows here...
</dm:reverseDifferences>
</dm:DifferenceModel>
</rdf:RDF>
```

6.2.4.7.4 Update resource

The difference model contains for a given resource forward difference and reverse difference statements if the resource is changed.

EXAMPLE:

The example below defines the move of the EnergyConsumer from 115k to 230k through the changed link from its Terminal to a different ConnectivityNode.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cim="cim-namespace-uri"
  xmlns:dm="difference-model-namespace-uri"
  xml:base="urn:uuid:">
  <dm:DifferenceModel rdf:about="#_26cc8d71-12f1-4de9-9e68-125d95073a75">
    <!-- header properties omitted for brevity -->

    <!-- Move EnergyConsumer load from 115K to 230K -->
    <dm:forwardDifferences rdf:parseType="Statements" >
      <rdf:Description rdf:about="#_33e4e305-1c70-4dcc-a423-45e4812dcd07">
        <cim:Terminal.ConnectivityNode rdf:resource="#_612fa147-902c-4f88-
be3f-0302b3750b18"/>
      </rdf:Description>
    </dm:forwardDifferences>
    <dm:reverseDifferences rdf:parseType="Statements" >
      <rdf:Description rdf:about="#_33e4e305-1c70-4dcc-a423-45e4812dcd07">
        <cim:Terminal.ConnectivityNode rdf:resource="#_5d74fc6a-b518-4a3e-
9e72-4827efd197cf"/>
      </rdf:Description>
    </dm:reverseDifferences>
  </dm:DifferenceModel>
</rdf:RDF>
```

For change of compound elements (6.2.3.7) the complete compound is replaced, i.e. the old element and all its members are removed by a reverse difference statement and added back with a forward difference statement.

6.3 CIMXML format style guide

A useful feature of RDF syntax is that it allows an arbitrary subset of a power system model to be serialized in a document. This is a two edged sword, however. A document produced by one party may not be usable by a second party if it does not contain all the properties expected. Moreover, a document containing a partial model may not be usable if the resource URN's do not agree with other documents.

The following guidelines apply to the content of a CIMXML document and help maximize the range of applications that can use it.

- Include the likely primary key properties of each resource at the point it is introduced. For example, the *cim:IdentifiedObject.name* and *cim:Equipment.EquipmentContainer* properties are likely to be required properties.
- Reason: a large class of applications will want to load a database with the model data. Many database schemas will require primary key values on insertion.
- Include single-valued properties rather than their many-valued inverse. For example, use *cim:Equipment.EquipmentContainer* and not *cim:EquipmentContainer.Contains_Equipments*.

- d) Reason: Because these properties are inverses, a statement predicated on one implies the converse statement predicated on the other. It is less error prone to include only one side and makes editing or transforming the document easier.
- e) When encountering many to many relationships, there is usually a primary direction of reference. Include the primary reference rather than their many-valued inverse. For example, use `cim:SynchronousMachine.MVArCapabilityCurves` and not `cim:MVArCapabilityCurve.SynchronousMachines`, since the primary relationship is from `SynchronousMachine` to `MVArCapabilityCurve`.
- f) Reason: Same reasons as for item c) above.
- g) When encountering a single-valued relationship with a single value inverse, include either one, but not both. Importing software needs to be designed to handle either direction of reference and infer the inverse.
- h) Reason: Because these properties are inverses, a statement predicated on one implies the converse statement predicated on the other. This is less error prone, and arguably, makes editing or transforming the document easier.
- i) Many valued properties, if used, appear as repeated property elements having the same property name.

6.4 Representing new, deleted and changed objects as CIMXML elements

The following cases exist for identification of elements and how they appear in full or differential models

- New objects are represented by the definition element (refer to 6.2.3.5) identified by a `rdf:ID` attribute in full or differential models.
- Deleted objects are represented by the definition element (refer to 6.2.3.5) identified by a `rdf:ID` attribute in differential models.
- Changed objects are represented by the description element (refer to 6.2.3.6) identified by a `rdf:about` attribute in differential models.
- An added property (e.g. internally a null value is changed to a valid value) is a change that appears only in the forward section of a difference model.
- A removed property (e.g. internally a valid value is changed to a null value) is a change that appears only in the backwards section of a difference model.

6.5 CIM RDF schema generation with CIM profile

IEC 61970-501 discusses the generation of CIM RDF Schema. A CIMXML model exchange document uses a subset of the CIM to address the model exchange needs of a specific use case; see Part 400 series profile documents. A CIM profile defines that portion of the CIM that an importer and exporter of a CIMXML document should be expected to handle. The RDF Schema for a profile then contains only the classes and properties defined for that profile.

A RDF Schema file can be generated from the CIM UML model by an application having a user interface where the subset of the CIM UML model is interactively specified. The RDF Schema file can be used by an application to validate a CIMXML document, refer to Figure 5.

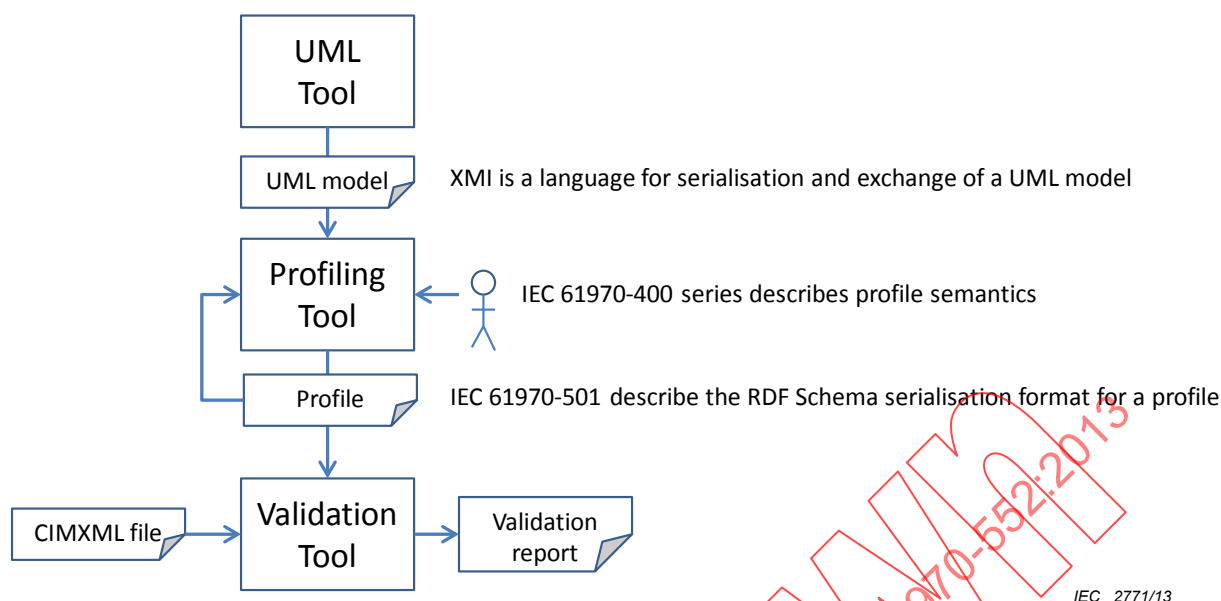


Figure 5 – Relations between UML, profile and CIMXML tools

6.6 CIM extensions

The CIM RDF schema can be extended with new classes and attributes by providing a separate namespace. Because a separate namespace is used, the customized CIMXML documents clearly delineate what is CIM standard and what is custom. Several different custom extensions can exist and be clearly identified within the same XML document. When these customized documents are imported to information systems that know nothing about the extensions, the elements with the unknown tags can be simply ignored. The following declaration identifies an extended namespace, *bpa*.

```
xmlns:bpa="http://www.bpa.gov/schema/cim_extension/2001may"
```

For example, a non-CIM attribute, *OriginalPO*, can be added to the *breaker* class, as shown below. These customized tags for BPA can be simply ignored if a system import program is not interested in such extensions.

```
<cim:SynchronousMachine rdf:ID="31dcf429-6Bfb-4e2e-b2996-42491b3abc1">
  <cim:IdentifiedObject.name>IN-2</cim:IdentifiedObject.name>
  <cim:SynchronousMachine.minimumMVar>-9999</cim:SynchronousMachine.minimumMVar>
  <cim:SynchronousMachine.operatingMode rdf:resource="http://iec.ch/TC57/2001/CIM-schema-cim10#SynchronousMachineOperatingMode.generator"/>
  <bpa:OriginalPO>PO1234378</bpa:OriginalPO>
  <cim:RegulatingCondEq.RegulationSchedule rdf:resource="#_ca32746
fa0024c2bbcf47bc97430bf87"/>
  <cim:Equipment.EquipmentContainer rdf:resource="#_6cb8701a-12f1-4de9-9e68-
125d95073a75"/>
</cim:SynchronousMachine>
```

The RDF schema corresponding to this extension can be added to a separate RDF schema document thereby keeping the CIM RDF schema clearly separate and allowing each to evolve independently.

6.7 RDF simplified syntax design rationale

The following points explain some of the choices made in the simplified syntax.

- 1) The literal properties could be represented by property attributes (RDF [3] grammar clause 6.10). This would be more compact. However, property elements were chosen

because they are easier to deal with in XSLT expressions. (For example, they can be sorted.) They also make it easier to represent multi-line text.

- 2) The syntax is flat, with a two-level resource/property structure. More deeply nested structures might be more compact. Moreover, a well-chosen nested structure might permit common queries to be more easily encoded in XSLT expressions. On the other hand, the flat structure was chosen because it is the simplest structure possible and is easy to produce and interpret. By avoiding any application dependency on the details of a nesting structure it should be a more portable syntax.
- 3) All resources are given a type at the time they are introduced (by the definition element). However, the RDF model allows a resource to be un-typed. In the present application, un-typed resources are not required. However the difference model uses un-typed resources as described in 6.2.3.6.

IECNORM.COM : Click to view the full PDF of IEC 61970-552:2013

Bibliography

1. *XML for CIM Model Exchange*, IEEE PES PICA 2001 Conference Paper, May 2001, A. deVos, S. Widergren, J. Zhu
2. *Simplified RDF Syntax for Power System Model Exchange*, CIMXML Interoperability Group Paper, 16 November 2000, A. deVos
3. *Resource Description Framework (RDF) Model and Syntax Specification*, W3C Recommendation 22 February 1999 <http://www.w3.org/TR/REC-rdf-syntax>, Ora Lassila, Ralph R. Swick
4. *Resource Description Framework (RDF) Schema Specification*, W3C Proposed Recommendation 03 March 1999 <http://www.w3.org/TR/PR-rdf-schema>, Dan Brickley, R.V. Guha, Netscape
5. *Uniform Resource Identifiers (URI): Generic Syntax*, Berners-Lee, Fielding, Masinter, Internet Draft Standard August, 1998; [RFC2396](http://www.rfc-editor.org/rfc/rfc2396.txt)
6. *Namespaces in XML*, Bray, Hollander, Layman eds, W3C Recommendation; <http://www.w3.org/TR/1999/REC-xml-names-19990114>
7. *Extensible Markup Language 1.0 (Second Edition)*, W3C Recommendation 6 October 2000, <http://www.w3.org/TR/REC-xml>, Bray, Paoli Sperberg-McQueen, Maler
8. "SiRPAC – Simple RDF Parser & Compiler", <http://www.w3.org/RDF/Implementations/SiRPAC>
9. "A Strawman Unstriped Syntax for RDF in XML", Tim Berners-Lee, November 1999, <http://www.w3.org/DesignIssues/Syntax>
10. "Simplified Syntax for RDF", Sergey Melnik, November 1999, <http://www-db.stanford.edu/~melnik/rdf/syntax.html>

IECNORM.COM : Click to view the full PDF of IEC 61970-552:2013

SOMMAIRE

AVANT-PROPOS	33
INTRODUCTION.....	35
1 Domaine d'application.....	36
2 Références normatives	36
3 Termes et définitions	37
4 En-tête d'échange de modèles	40
4.1 Généralités	40
4.2 Documents CIMXML et en-têtes	40
4.3 Description des modèles et des données d'en-tête	40
4.4 Flux de travail	42
5 Identification des objets	44
5.1 URI comme identificateurs	44
5.2 Informations relatives à rdf:ID et à rdf:about	45
5.3 Identification des éléments CIMXML	46
6 Règles et conventions relatives au format CIMXML	46
6.1 Généralités	46
6.2 Syntaxe RDF simplifiée	47
6.2.1 Généralités	47
6.2.2 Notation.....	48
6.2.3 Définition de la syntaxe	48
6.2.4 Extension de la syntaxe pour le modèle de différence.....	53
6.3 Guide de style du format CIMXML	58
6.4 Représentation des objets nouveaux, supprimés et modifiés en éléments CIMXML	59
6.5 Génération du schéma RDF du CIM avec le profil CIM.....	59
6.6 Extensions du CIM	60
6.7 Justification de conception de la syntaxe simplifiée de RDF	61
Bibliographie	62
Figure 1 – Modèle avec en-tête	41
Figure 2 – Exemple d'événements de flux de travail.....	43
Figure 3 – Exemple d'événements de flux de travail avec plus de dépendances	44
Figure 4 – Mécanisme d'échange de modèles de réseau électrique basé sur le langage CIMXML	47
Figure 5 – Relations entre UML, profil et outils CIMXML	60
Tableau 1 – Attributs d'en-tête.....	41

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

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

Partie 552: Format d'échange de modèle CIMXML

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (CEI) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI 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, la CEI – 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 la CEI"). 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 la CEI, participent également aux travaux. La CEI 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 la CEI concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de la CEI intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI 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 la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de la CEI dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) La CEI 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 la CEI. La CEI 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 la CEI, 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 la CEI, 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 la CEI ou de toute autre Publication de la CEI, 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 la CEI peuvent faire l'objet de droits de brevet. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale CEI 61970-552 a été établie par le comité d'études 57 de la CEI: Gestion des systèmes de puissance et échanges d'informations associés.

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
57/1386/FDIS	57/1402/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

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

Une liste de toutes les parties de la série CEI 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 la CEI.

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 la CEI 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 internationale est l'une des différentes parties de la série de normes CEI 61970 qui définissent une interface de programmation d'application (API¹) pour un système de gestion d'énergie (EMS²).

La CEI 61970-301 spécifie un Modèle d'Information Commun (CIM³): une vue logique des aspects physiques des opérations d'un service public de distribution d'électricité. Le CIM est décrit à l'aide du Langage de Modélisation Unifié (UML⁴), un langage utilisé pour spécifier, visualiser et documenter les systèmes de façon orientée objet. Le langage UML est un langage d'analyse et de conception; et non un langage de programmation. Pour que les logiciels utilisent le CIM, ce dernier doit être converti en un schéma prenant en charge une interface programmable.

La CEI 61970-501 décrit la traduction du CIM au format UML en un format lisible par une machine comme exprimé dans la représentation du Langage de balisage extensible (XML⁵) de ce schéma à l'aide du langage de spécification du Schéma du Cadre de Description des Ressources (RDF⁶).

La CEI 61970-552 spécifie la manière dont le schéma RDF du CIM spécifié dans la CEI 61970-501 est utilisé pour échanger des modèles de réseau électrique à l'aide du langage XML (appelé CIMXML) défini dans la série de normes de profils 61970-45x, telles que le Profil d'échange du modèle de réseau de transport du CIM décrit dans la CEI 61970-452.

1 API = *Application Program Interface*.

2 EMS = *Energy Management System*.

3 CIM = *Common Information Model*.

4 UML = *Unified Modelling Language*.

5 XML = *Extensible Markup Language*.

6 RDF = *Resource Description Framework*.

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

Partie 552: Format d'échange de modèle CIMXML

1 Domaine d'application

La présente Norme Internationale spécifie une Spécification d'Interface de Composants (Component Interface Specification (CIS)) pour les Interfaces de Programmation d'Application des Systèmes de Gestion d'Énergie. Cette partie spécifie le format et les règles pour échanger les informations de modélisation basées sur le CIM. Elle utilise le Schéma RDF du CIM présenté dans la CEI 61970-501 comme le cadre de métamodèle pour générer les documents XML contenant les informations relatives à la modélisation des réseaux électriques. Le style de ces documents est appelé format CIMXML.

L'échange de modèles par transfert de fichiers répond à plusieurs objectifs utiles. Les documents de profil tels que la CEI 61970-452 et d'autres profils dans la série de normes 61970-45x exposent les exigences et les cas d'utilisation posant le contexte pour cette tâche. Bien que le format puisse être utilisé pour l'échange d'informations basé sur le CIM, les profils (ou sous-ensembles) spécifiques du CIM sont identifiés afin d'aborder les exigences d'échange particulières. L'exigence initiale contrôlant la consolidation de la présente spécification est l'échange des informations de modélisation du réseau de transmission pour la coordination de la sécurité des réseaux électriques.

La présente norme prend en charge un mécanisme pour les logiciels provenant de fournisseurs indépendants afin de produire et d'utiliser les informations de modélisation décrites dans le CIM dans un format commun. La solution proposée:

- est à la fois lisible par l'homme et par la machine, bien qu'elle soit essentiellement destinée à un accès programmatique,
- peut être accessible à l'aide de tout outil prenant en charge le Document Object Model (DOM) et d'autres interfaces de programmation d'application de XML normalisé,
- est autodescriptive,
- met à profit les recommandations actuelles du World Wide Web Consortium (W3C).

Ce document est le document de Spécification d'Interface de Composants de Niveau 2 qui décrit sous forme narrative (avec du texte et des exemples basés sur le CIM) la définition détaillée du format CIMXML.

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).

CEI 60050 (toute la série), *Vocabulaire Electrotechnique International*

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

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

CEI 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)

W3C: RDF/XML Syntax Specification (Spécification de la Syntaxe RDF/XML)

W3C: Extensible Markup Language (Langage de Balisage Extensible) (XML) 1.0

W3C: XLS Transformations (Transformations XSL) (XSLT)

W3C: Document Object Model (Modèle Objet de Document) (DOM)

3 Termes et définitions

Pour les besoins de la présente Norme Internationale, les termes et définitions donnés dans la CEI 60050 (pour les définitions de glossaire général) et la CEI 61970-2 (pour les définitions de glossaire de l'EMS-API), ainsi que les suivants s'appliquent.

3.1

Interface de Programmation d'Application API

ensemble des fonctions publiques qu'offre un composant exécutable d'application pour être utilisées par d'autres composants exécutables d'application

3.2

Modèle d'Information Commun CIM

modèle abstrait représentant tous les principaux objets dans une entreprise de service public de distribution d'électricité généralement contenus dans un modèle d'information EMS

Note 1 à l'article: En fournissant une façon normalisée de représenter des ressources de réseau électrique comme des classes et des attributs d'objets ainsi que leurs relations, le CIM facilite l'intégration des applications EMS développées de façon indépendante par différents fournisseurs, entre des systèmes EMS complets développés de façon indépendante ou entre un système EMS et d'autres systèmes concernés par différents aspects de l'exploitation d'un réseau électrique tels que la gestion de la production ou de la distribution.

3.3

CIMXML

format de sérialisation pour l'échange de données XML comme défini dans le présent document

3.4

Modèle Objet de Document DOM (Document Object Model)

interface de plate-forme et de langage neutre définie par le World Wide Web Consortium (W3C) permettant aux programmes et aux scripts d'accéder dynamiquement au contenu, à la structure et au style des documents et de les échanger

3.5

Définition de Type de Document

DTD (Document Type Definition)

norme pour décrire le vocabulaire et la syntaxe associés à un document XML

Note 1 à l'article: Le Schéma XML et RDF sont d'autres formes pouvant être utilisées.

3.6

Système de Gestion d'Énergie

EMS

système informatique comprenant une plate-forme logicielle offrant des services de soutien de base et un ensemble d'applications offrant les fonctionnalités requises pour le bon fonctionnement des installations de production et de transport d'électricité afin d'assurer la sécurité adéquate d'approvisionnement énergétique à un coût minimal

3.7

Langage de Balisage Hypertexte

HTML⁷

langage de balisage utilisé pour mettre en forme et présenter les informations sur le Web

3.8

modèle

ensemble de données décrivant les objets ou les entités réel(le)s ou calculé(e)s

Note 1 à l'article: Dans le contexte du CIM, la sémantique des données est définie par des profils; voir 3.9.

Note 2 à l'article: Dans une analyse de réseau électrique, un modèle désigne un ensemble de données statiques décrivant le réseau électrique. Des exemples de Modèles incluent le Static Network Model (Modèle de Réseau Statique), la Topology Solution (Solution de Topologie) et la Network Solution (Solution de Réseau) produits par une application de calcul de répartition ou d'estimateur d'état.

3.9

profil

schéma qui définit la structure et la sémantique d'un modèle pouvant être échangé

Note 1 à l'article: Un Profil est un sous-ensemble restreint du CIM plus général.

3.10

document de profil

ensemble de profils destinés à être utilisés ensemble dans un but commercial particulier

3.11

Cadre de Description des Ressources

RDF

langage recommandé par le W3C pour exprimer des métadonnées que les ordinateurs peuvent traiter simplement

Note 1 à l'article: RDF utilise XML comme syntaxe d'encodage.

3.12

schéma RDF

langage de spécification de schéma exprimé à l'aide de RDF pour décrire les ressources et leurs propriétés, y compris la manière dont les ressources sont liées les unes aux autres, utilisé pour spécifier un schéma spécifique à l'application

⁷ L'abréviation "HTML" est dérivée du terme anglais développé correspondant "Hypertext Markup Language".

3.13

objets réels

objets appartenant au domaine du problème du monde réel différents des objets d'interface et des objets de contrôleur au sein de la mise en œuvre

Note 1 à l'article: Les objets réels pour le domaine de l'EMS sont définis comme des classes dans la CEI 61970-301 Modèle d'Information Commun.

Note 2 à l'article: Les classes et les objets modélisent ce qui se trouve dans un réseau électrique et qui nécessite d'être représenté de la même façon que pour les applications EMS. Une classe est une description d'un objet réel, telle que PowerTransformer, GeneratingUnit ou Load qui nécessite d'être représentée comme faisant partie du modèle intégral de réseau électrique dans un EMS. Parmi les autres types d'objets figurent des éléments tels que les programmes et les mesures que des applications EMS nécessitent également de traiter, d'analyser et d'enregistrer. De tels objets nécessitent une représentation commune pour atteindre les objectifs de compatibilité de connexion et d'interopérabilité de la norme EMS-API. Un objet particulier dans un réseau électrique ayant une identité unique est modélisé comme une instance de la classe à laquelle il appartient.

3.14

Langage Normalisé de Balisage Généralisé

SGML⁸

norme internationale pour la définition de méthodes indépendantes de l'appareil et du système pour représenter les textes au format électronique

Note 1 à l'article: HTML et XML sont issus de SGML.

3.15

Langage de Modélisation Unifié

UML⁹

langage de modélisation orienté objet et méthodologie pour spécifier, visualiser, élaborer et documenter les artefacts d'un processus intensif du système

3.16

Identificateur de Ressource Uniforme

URI¹⁰

syntaxe et sémantique normalisées de Web pour identifier (référer) les ressources (éléments tels que des fichiers, des documents, des images)

3.17

Langage de Balisage Extensible

XML¹¹

sous-ensemble du Langage Normalisé de Balisage Généralisé (SGML), ISO 8879, pour insérer des données structurées dans un fichier texte

Note 1 à l'article: Il s'agit d'une recommandation approuvée du W3C. Elle est gratuite, indépendante de la plateforme et adaptée à de nombreux outils logiciels facilement disponibles.

3.18

Langage Extensible de Feuille de Style

XSL¹²

langage pour exprimer les feuilles de style des documents XML

⁸ L'abréviation "SGML" est dérivée du terme anglais développé correspondant "Standard Generalized Markup Language".

⁹ L'abréviation "UML" est dérivée du terme anglais développé correspondant "Unified Modelling Language".

¹⁰ L'abréviation "URI" est dérivée du terme anglais développé correspondant "Uniform Resource Identifier".

¹¹ L'abréviation "XML" est dérivée du terme anglais développé correspondant "eXtensible Markup Language".

¹² L'abréviation "XSL" est dérivée du terme anglais développé correspondant "eXtensible Stylesheet Language".

4 En-tête d'échange de modèles

4.1 Généralités

L'échange de modèles implique généralement l'échange d'un ensemble de documents, chacun contenant des données d'instance (désignées comme modèle) et un en-tête. La structure et la sémantique de chaque modèle sont décrites par un profil, qui n'est pas inclus dans les données échangées. L'échange global est régi par un ensemble de profils appelé Groupe de Profils.

Un en-tête décrit le contenu du modèle contenu dans un document, par exemple la date de création du modèle, la description, etc. L'en-tête peut également identifier d'autres modèles ainsi que leur relation avec le modèle actuel. Ces informations s'avèrent importantes lorsque les modèles font partie d'un flux de travail comme lorsque les modèles sont liés les uns aux autres comme par exemple lorsqu'un modèle succède à et/ou dépend d'un autre.

Les Paragraphes 4.2 à 4.4 définissent le modèle avec des données d'en-tête et le flux de travail qu'elles sont destinées à prendre en charge.

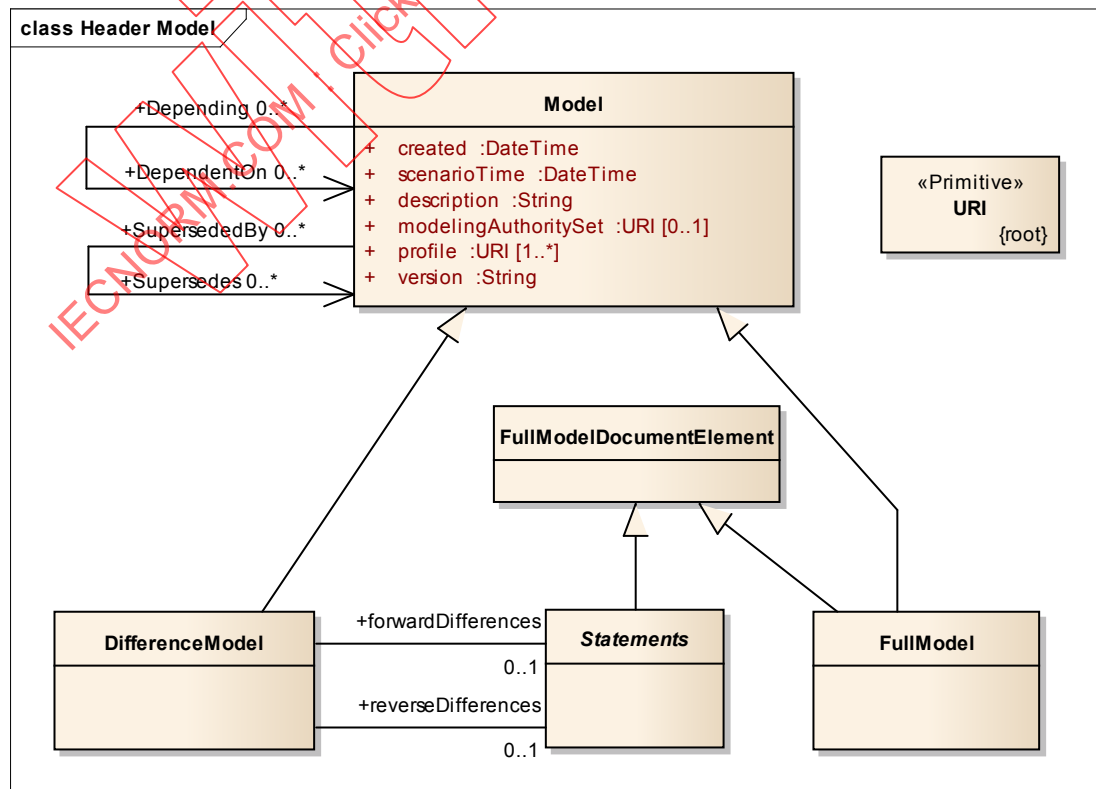
4.2 Documents CIMXML et en-têtes

Un document CIMXML est décrit par un seul en-tête. Il n'est pas permis d'avoir plusieurs en-têtes dans un document CIMXML. Par conséquent, les données d'instance dans un document CIMXML n'adhèrent qu'à un seul profil.

Lorsque plusieurs documents CIMXML possiblement associés nécessitent d'être maintenus ensemble, ils doivent être recueillis dans une archive, par exemple zip.

4.3 Description des modèles et des données d'en-tête

Une description d'un modèle est attachée comme données d'en-tête au modèle. La Figure 1 décrit le modèle avec des informations d'en-tête.



Légende

Anglais	Français
Class Header Model	Class Header Model (Modèle d'en-tête)

Figure 1 – Modèle avec en-tête

Dans la Figure 1, les classes FullModel, DifferenceModel et Statements décrivent les données du modèle alors que l'en-tête est décrit par les classes Model et Description. Ci-dessous est donnée une description de bas en haut de ces classes:

- La classe FullModelDocumentElement représente les éléments pouvant apparaître dans un document de modèle intégral. Elle possède les deux sous-types Statements ou FullModel, les deux étant décrits ci-dessous ultérieurement. Un document de modèle intégral contient généralement un élément FullModel et un ensemble d'éléments Definition.
- La classe Statement représente un ensemble d'éléments Definition (voir 6.2.3.5) et/ou Description (voir 6.2.3.6).
- La classe FullModel (voir 6.2.3.4) représente l'en-tête du modèle intégral et son contenu est décrit par la classe Model.
- La classe DifferenceModel (voir 6.2.4.6) représente l'en-tête du modèle de différence. Le contenu est décrit par la classe Model et les rôles d'association forwardDifferences et reverseDifferences. Les deux rôles d'association peuvent posséder un ensemble de Statements.
- La classe Model décrit le contenu de l'en-tête identique à FullModel et à DifferenceModel. Un Modèle est identifié par un attribut rdf:about. L'attribut rdf:about décrit de façon unique le modèle et non le document dans lequel l'en-tête existe. Par conséquent, plusieurs documents créés à partir du même modèle de données inchangé posséderont le même rdf:about. Ceci signifie également qu'une modification de modèle donne lieu à un nouvel attribut rdf:about lors de la création suivante d'un document.

Les attributs de la classe Model sont décrits dans le Tableau 1.

Tableau 1 – Attributs d'en-tête

Classe	Attribut	Description
Model	created	Date de création du modèle (noter qu'il ne s'agit généralement pas de la date à laquelle le document CIMXML a été créé, qui est postérieure).
Model	scenarioTime	Date et heure que le modèle représente, par exemple l'heure actuelle pour un modèle opérationnel, un modèle historique ou un futur modèle planifié.
Model	description	Une description du modèle, par exemple le nom de la personne ayant créé le modèle ainsi que le but de cette création.
Model	modelingAuthoritySet	Un URN décrivant le modèle d'équipement qui est la source des données dans un document CIMXML, par exemple un modèle pour le pays entier ou une partie du pays.
Model	profile	Un URN décrivant les Profils régissant ce modèle. Il identifie de façon unique le Profil et sa version.
Model	version	Une description de la version du modèle à l'origine des données dans un document CIMXML. Exemples: – Variations du modèle d'équipement pour le ModelingAuthoritySet – Différents cas d'étude amenant différentes solutions. L'attribut de version est une chaîne personnalisée modifiée en synchronisation avec l'identificateur rdf:about, se référer à la description de la classe Model ci-dessus

Classe	Attribut	Description
Model	DependentOn	Une référence aux modèles dont dépend le modèle décrit par ce document, par exemple – Une solution de calcul de répartition dépend du modèle de topologie à partir duquel elle a été calculée – Un modèle de topologie calculé par un processeur de topologie dépend du modèle de réseau à partir duquel il a été calculé.
Model	Depending	Tous les modèles dépendant de ce modèle. Ce rôle n'est pas destiné à être inclus dans un document échangeant des données d'instance.
Model	Supersedes	Lorsqu'un modèle est mis à jour, le modèle résultant remplace les modèles de base utilisés pour la mise à jour. Par conséquent, il s'agit d'une référence aux documents CIMXML décrivant les modèles mis à jour.
Model	SupersededBy	Tous les modèles remplaçant ce modèle. Ce rôle n'est pas destiné à être inclus dans un document échangeant des données d'instance.

L'attribut de profil est un URI au format suivant:

<http://iec.ch/<committee>/<year>/<standard>-<part>/<profile>/<version>>

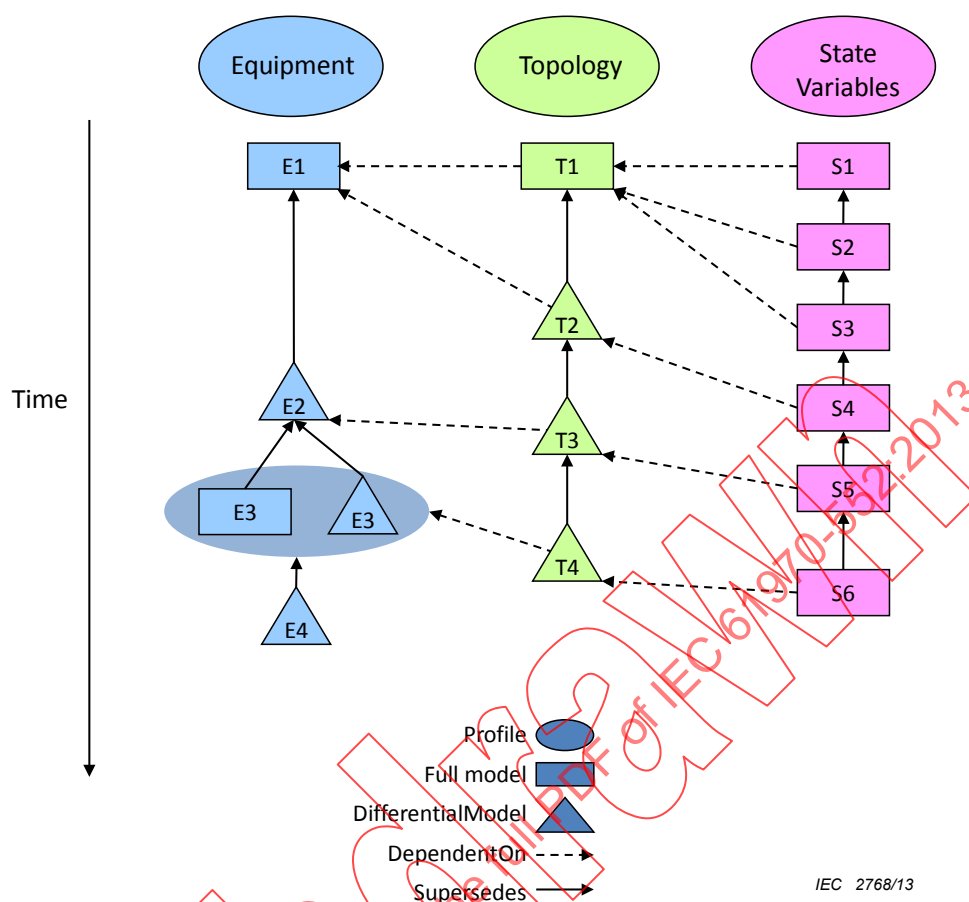
par exemple <http://iec.ch/TC57/2011/61970-452/Equipment/2>

L'UML à la Figure 1 est converti en éléments CIMXML comme suit:

- 1) Une classe feuille à la Figure 1 (DifferenceModel, Statements et FullModel) apparaît comme des éléments de classe sous l'élément document (6.2.3.3).
- 2) Les éléments Statement apparaissent comme des éléments Definition (6.2.3.5) ou Description (6.2.3.6).
- 3) Les attributs littéraux, par exemple Model.created, apparaissent comme des éléments literal property (6.2.3.8).
- 4) Les rôles apparaissent, par exemple Model.Supersedes, comme des éléments resource property (6.2.3.10).
- 5) Les attributs et les rôles hérités apparaissent directement comme des éléments de la classe feuille répondant aux règles 3, 4 et 5 ci-dessus.
- 6) Un document de modèle CIMXML est identifié par un attribut rdf:about de Model (implicite dans l'UML). Par conséquent, les rôles DependentOn et Supersedes sont des références à l'attribut rdf:about de Model.
- 7) Un document de modèle intégral peut être régénéré à plusieurs reprises à partir des mêmes données sources. Les documents de modèle intégral régénérés à partir de données sources inchangées conservent l'identification du modèle (rdf:about de Model) inchangée du document de modèle intégral d'origine.
- 8) Lors de la génération d'un document de modèle intégral remplaçant un différentiel, le nouveau document de modèle intégral possédera la même identification de modèle (rdf:about de Model) que le différentiel si le modèle n'a pas été modifié depuis la création du différentiel. Par conséquent, il s'agit d'une alternative au différentiel.

4.4 Flux de travail

Un flux de travail est décrit par une séquence d'événements d'échanges. La description du modèle en 4.3 prend en charge les événements de flux de travail en fonction du temps avec l'attribut Model.Supersedes et les événements liés aux profils avec l'attribut Model.DependentOn. Un exemple est présenté à la Figure 2.

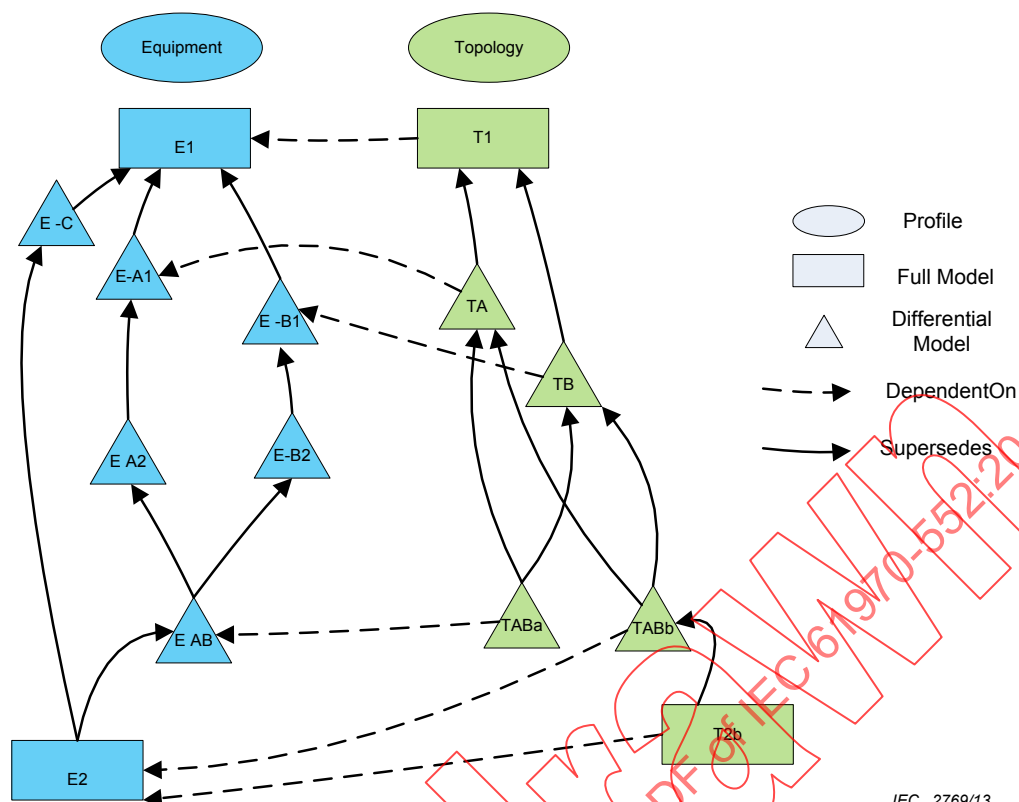


Légende

Anglais	Français
Time	Temps
Equipment	Équipement
Topology	Topologie
State Variables	Variables d'état
Profile	Profil
Full model	Modèle intégral
DifferentialModel	Modèle différentiel
DependentOn	Dépend de
Supersedes	Remplace

Figure 2 – Exemple d'événements de flux de travail

Dans cet exemple, un modèle de réseau résolu est échangé comme un ensemble de modèles régi par un Profile Document (Groupe de Profils) comportant les documents d'Equipment, de Topology et de State Variables. La ligne des temps à gauche de la Figure 2 représente la manière dont le document du modèle Equipment est échangé au cours du temps. La ligne des temps au centre montre la manière dont les nouveaux résultats de Topology sont échangés au cours du temps ainsi que les modèles Equipment dont chacun dépend. La ligne des temps à droite montre la manière dont plusieurs documents de State Variables sont échangés ainsi que les documents Topology dont ils dépendent. Noter également que le modèle Equipment E2 est représenté à la fois par un document intégral et un document incrémentiel. La situation de la Figure 2 représente un cas simple. Une situation plus complexe est présentée à la Figure 3.



IEC 2769/13

Légende

Anglais	Français
Equipment	Équipement
Topology	Topologie
Profile	Profil
Full model	Modèle intégral
Differential Model	Modèle différentiel
DependentOn	Dépend de
Supersedes	Remplace

Figure 3 – Exemple d'événements de flux de travail avec plus de dépendances

Les documents CIMXML de la Figure 3 peuvent être créés à partir d'un environnement modélisateur de données comportant plusieurs voies de changement d'un modèle en parallèle, par exemple, le modèle équipement comporte trois voies E-Ax, E-Bx et A-C qui fusionnent dans le modèle intégral E2 remplaçant les voies du modèle équipement.

Un récepteur des documents CIMXML peut utiliser n'importe quel document de topologie TA, TB, TABa ou T2b avec le modèle équipement de E2. Étant donné que l'émetteur (ici le modélisateur de données) n'a vérifié que T2b avec E2, il s'agit de la seule combinaison supposée fonctionner. Concernant T2b, le récepteur peut choisir d'appliquer TB et TABb à T1 à la place d'utiliser T2b.

5 Identification des objets

5.1 URI comme identificateurs

Les UUID (Universally Unique Identifier) (Identificateur Universel Unique), aussi connus sous le nom de GUID (Globally Unique Identifier) (Identificateur Globalement Unique) peuvent être utilisés pour identifier des ressources de manière à ce que

- les identificateurs puissent être attribués indépendamment et de manière unique par différentes autorités. Ceci représente un énorme avantage avec l'UUID.
- les identificateurs soient stables dans le temps et dans les documents.

Si, de plus, l'UUID est incorporé dans un Nom Uniforme de Ressources (URN), alors le document peut être simplifié en éliminant les déclarations des espaces de noms de base XML (attributs `xml:base`). L'URN est un URI concis, absolu, de longueur fixe.

La norme pour un URN contenant un UUID est définie par l'Internet Engineering Task Force RFC 4122.

RFC 4122 spécifie la syntaxe de l'URN et la manière dont la partie de l'UUID suivant les derniers deux-points est attribuée. L'algorithme est aligné et techniquement compatible avec la CEI 9834-8:2004 ou Recommandation UIT-T. X.667, 2004 "Technologies de l'information - Interconnexion des systèmes ouverts - Procédures opérationnelles des organismes d'enregistrement de l'OSI: génération et enregistrement des identificateurs universels uniques (UUID) et utilisation de ces identificateurs comme composants d'identificateurs d'objets ASN.1".

Les éléments CIMXML sont identifiés par un URI. Un URI peut prendre deux formes:

- URL
- URN

Les formes URL et URN ont des structures fondamentalement différentes, c'est-à-dire:

- Forme URL; *protocol://authority/path?query#fragment* où le *protocol* en CIMXML est `http`
- Forme URN: *urn:namespace:specification* où *namespace* en CIMXML est `uuid`.

Le format URN *specification* est résumé ci-dessous

- Numération hexadécimale à 8 caractères
- Un tiret "-"
- Numération hexadécimale à 4 caractères
- Un tiret "-"
- Numération hexadécimale à 4 caractères
- Un tiret "-"
- Numération hexadécimale à 4 caractères
- Un tiret "-"
- Numération hexadécimale à 12 caractères

où les lettres sont en minuscule.

Un exemple de forme URN est donné ci-dessous

- "urn:uuid:26cc8d71-3b7e-4cf8-8c93-8d9d557a4846".

5.2 Informations relatives à `rdf:ID` et à `rdf:about`

Un élément CIMXML peut être identifié par deux constructions de RDF différentes:

- `rdf:ID`
- `rdf:about`

L'utilisation de `rdf:ID` et de `rdf:about` possède une signification spécifique qui n'est pas conforme à leur définition dans RDF. Les significations sont:

- un `rdf:ID` spécifie la vie de l'objet globalement, c'est-à-dire sa création ou sa suppression.
- un `rdf:about` est une référence à un objet existant.

5.3 Identification des éléments CIMXML

L'identification des objets est tellement essentielle dans RDF que tous les éléments représentant des objets sont identifiés par un attribut XML `rdf:ID` ou `rdf:about`. Toutes les classes dans le CIM qui héritent de `IdentifiedObject` ont l'attribut d'identification d'objet UML `IdentifiedObject.mRID`. L'attribut est mis en correspondance de manière implicite avec l'attribut XML `rdf:ID/rdf:about`.

Un document CIMXML ne peut utiliser que la forme URN (voir 5.1) comme décrit plus en détail ci-après.

Les fichiers CIMXML contiennent des éléments XML décrivant les objets du CIM (`ACLineSegments`, `Substations`, etc.). Le CIM comporte de nombreux rôles d'association qui apparaissent comme des références dans les éléments XML (généralement comme attributs `rdf:resource` ou `rdf:about`). Les données du CIM sont échangées dans différents documents CIMXML qui dépendent les uns des autres comme décrit à l'Article 4. Certaines références traversent ensuite les frontières des documents CIMXML. L'une des conséquences est que l'identification d'un objet du CIM doit être stable lors de sa durée de vie. Si tel n'est pas le cas, les objets de référencement au niveau des frontières des documents se briseront.

Une pratique courante dans les systèmes orientés objet consiste à supposer que tous les objets possèdent un identificateur unique dans l'espace et dans le temps ce qui signifie que:

- Différents identificateurs sont attribués à différents objets.
- Les identificateurs une fois attribués ne sont jamais réutilisés même si l'objet d'origine les possédant n'est plus présent.

La forme URN décrite en 5.1 est utilisée comme identification d'élément CIMXML avec les différences suivantes

- Le préfixe "`urn:uuid:`" est remplacé par un trait bas "`_`". Le trait bas évite d'utiliser un caractère numérique de début pour la partie non de base de l'identificateur. Commencer la partie non de base de l'identificateur avec un caractère numérique est un RDF non valide. Dans tous les cas, le trait bas est ajouté pour simplifier les analyseurs, même si l'UUID commence avec un caractère non numérique.
- Le préfixe est défini comme un `xml:base="urn:uuid:"`

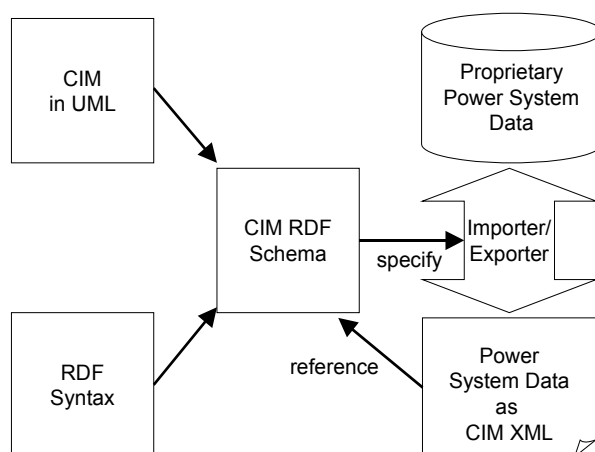
Quelques exemples:

- `rdf:ID="_26cc8d71-3b7e-4cf8-8c93-8d9d557a4846"`.
- `rdf:about="#_26cc8d71-3b7e-4cf8-8c93-8d9d557a4846"`.

6 Règles et conventions relatives au format CIMXML

6.1 Généralités

En tenant compte du Schéma RDF du CIM décrit dans la CEI 61970-501, un modèle de réseau électrique peut être converti pour l'exportation sous forme d'un document XML (voir Figure 3). Il est fait référence à ce document comme un document CIMXML. Toutes les balises (descriptions de ressources) utilisées dans le document CIMXML sont fournies par le schéma RDF du CIM. Le document d'échange de modèle CIMXML résultant peut être analysé et les informations importées dans un système étranger.



IEC 2770/13

Légende

Anglais	Français
CIM in UML	CIM en UML
CIM RDF Schema	Schéma RDF du CIM
RDF Syntax	Syntaxe RDF
Proprietary Power System Data	Données du réseau électrique propriétaire
Importer/Exporter	Importateur/Exportateur
Power System Data as CIM XML	Données du réseau électrique au format CIM XML
specify	spécifier
reference	faire référence à

Figure 4 – Mécanisme d'échange de modèles de réseau électrique basé sur le langage CIMXML

6.2 Syntaxe RDF simplifiée

6.2.1 Généralités

La syntaxe RDF fournit de nombreuses façons de représenter le même ensemble de données. Par exemple, une association entre deux ressources peut être écrite avec un attribut ressource ou en imbriquant un élément dans un autre. Cela est susceptible de rendre difficile l'utilisation de certains outils XML, tels que les processeurs XSL, avec le document CIMXML.

Par conséquent, seul un sous-ensemble de la Syntaxe RDF est appliqué lors de la création des documents CIMXML. Cette syntaxe simplifie le travail des personnes chargées de la mise en œuvre pour élaborer un logiciel de sérialisation et de désérialisation du modèle, ainsi que pour améliorer l'efficacité des outils XML généraux lorsqu'ils sont utilisés avec des documents CIMXML. La syntaxe réduite est un propre sous-ensemble de la syntaxe RDF normalisée; par conséquent, elle peut être lue par le logiciel de désérialisation de RDF disponible.

Les sous-sections suivantes définissent un sous-ensemble de la Syntaxe RDF. Cette syntaxe simplifiée est destinée à l'échange des modèles du réseau électrique entre les entreprises de service public. Le but de la spécification est de faciliter la tâche des personnes chargées de la mise en œuvre pour élaborer le logiciel de désérialisation pour les données RDF, pour simplifier leur choix lors de la sérialisation des données RDF et pour améliorer l'efficacité des outils XML généraux tels que les processeurs XSLT lorsqu'ils sont utilisés avec les données RDF sérialisées.

La syntaxe réduite est un propre sous-ensemble de la syntaxe RDF normalisée. Par conséquent, elle peut être lue par un logiciel de désérialisation RDF comme SirPAC [8]¹³. Sur ce point, elle se distingue des autres propositions par une syntaxe simplifiée, telle que [9], [10].

La syntaxe réduite n'altère pas la puissance du modèle de données RDF. Cela signifie que n'importe quelle donnée RDF peut être échangée en utilisant cette syntaxe. De plus, les fonctionnalités de RDF sont préservées telles que l'aptitude à étendre un modèle défini dans un document avec déclarations dans un second document.

6.2.2 Notation

La syntaxe simplifiée est définie dans la section suivante. Chaque type d'élément est défini dans une sous-section commençant par un modèle de l'élément, suivi de texte de définition et d'une référence à la grammaire RDF. La sémantique de l'élément n'est pas détaillée (voir la recommandation RDF [3] pour cette information). La notation pour le modèle de l'élément est comme suit:

- Un symbole en italique au lieu d'un type d'élément, le nom de l'attribut ou la valeur de l'attribut indique le type de nom requis ou la valeur requise. Le symbole sera défini dans le texte.
- Le symbole *rdf* indique que le préfixe de l'espace de noms est choisi par la mise en œuvre pour l'espace de noms RDF. De la même manière, le symbole *cim* indique le préfixe d'espace de noms du CIM choisi.
- Un commentaire au sein du modèle d'élément indique le contenu autorisé. Un symbole en italique indique un type d'élément ou autre contenu défini dans le texte. Une construction (a | b) indique que a et b sont possibles. Une construction a* indique zéro répétition de a ou plus.
- Tout le reste du texte dans le modèle est littéral.

6.2.3 Définition de la syntaxe

6.2.3.1 Généralités

La définition de la syntaxe est enrichie d'exemples. Il convient que les exemples aident à mieux comprendre la définition de la syntaxe formelle. Le même exemple est utilisé pour plusieurs définitions de syntaxe. La syntaxe présentée dans l'exemple est indiquée en gras.

6.2.3.2 URI des espaces de noms définis dans la présente spécification

Les espaces de noms suivants sont définis dans la présente spécification:

- `cim-model-description_uri` décrit par `xmlns:md`
- `difference-model-namespace-uri` décrit par `xmlns:dm`

Leurs valeurs sont définies comme:

- `xmlns:md="http://iec.ch/TC57/61970-552/ModelDescription/1#"`
- `xmlns:dm="http://iec.ch/TC57/61970-552/DifferenceModel/1#"`

6.2.3.3 Élément Document

```
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:cim="cim-namespace-uri"
  xmlns:md="cim-model-description_uri">
  <!-- Content: full-model (definition|description)* -->
</rdf:RDF>
```

¹³ Les chiffres entre crochets se réfèrent à la bibliographie.

```
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:cim="cim-namespace-uri"
  xmlns:md="cim-model-description-uri"
  xml:base="urn:uuid:">
  <!-- Content: full-model (definition|description)* -->
</rdf:RDF>
```

Exemple:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cim="http://iec.ch/TC57/2004/CIM-schema-cim10#"
  xmlns:md="http://iec.ch/TC57/61970-552/ModelDescription/1#"
  xml:base="urn:uuid:">
</rdf:RDF>
```

- Le type d'élément est rdf:RDF.
- L'espace de noms RDF doit être déclaré comme `http://www.w3.org/1999/02/22-rdf-syntax-ns#`
- L'espace de noms du CIM doit être déclaré. Avec les versions les plus récentes du schéma du CIM, la version nécessite d'être ajustée dans l'espace de noms du CIM. Il est nécessaire que les parties échangeant les documents s'accordent sur la version utilisée.
- D'autres espaces de noms peuvent être déclarés.
- L'attribut `xml:base` doit toujours être présent, voir 5.3.

Article relatif à la grammaire RDF: 6.1.

6.2.3.4 Élément FullModel

```
<md:FullModel rdf:about=model-uri>
  <!-- Content: (literal-property|resource-property
  |compound-property)* -->
</md:FullModel>
```

Exemple:

```
<md:FullModel rdf:about="#_26cc8d71-...">
  <md:Model.created>2008-12-24</md:Model.created>
  <md:Model.Supersedes rdf:resource="#_26cc8d71-a002-4c2b-bcf4-7bc97430bf87"/>
  <md:Model.DependentOn rdf:resource="#_26cc8d71-a002-4c2b-bcf4-7bc97430bf88"/>
  <md:Model.version>V32</md:Model.version>
  <md:Model.modelingAuthoritySet>http://polarenergy.com/2008/NorthPoleTSO</md:Model.modelingAuthoritySet>
  <md:Model.description>Santa Claus made a study case peak load summer base topology solution</md:Model.description>
  <md:Model.profile>http://iec.ch/TC57/61970-456/StateVariables/1</md:Model.profile>
</md:FullModel>
```

- L'élément full model introduit un nouveau modèle.
- La valeur de l'attribut `about`, `model-uri`, est un nom choisi par la mise en œuvre. `model-uri` identifie de façon unique un document et constitue le nom référencé par d'autres documents, par exemple `Supersedes` ou `DependentOn`, comme indiqué à la Figure 2.

6.2.3.5 Élément Definition

```
<classname rdf:ID=identity>
  <!-- Content:
  (literal-property|resource-property|compound-property)*
  -->
```



```
</classname>
```

```
<classname rdf:about=resource-uri>
  <!-- Content:
    (literal-property|resource-property|compound-property) *
  -->
</classname>
```

Exemple:

```
<cim:SynchronousMachine rdf:about="#_31dcf429-6bfb-4e2e-b2996-42491b3abc1">
  <cim:IdentifiedObject.name>IN-2</cim:IdentifiedObject.name>
  <cim:SynchronousMachine.minimumMVar>-9999</cim:SynchronousMachine.minimumMVar>
  <cim:SynchronousMachine.operatingMode rdf:resource="http://iec.ch/TC57/2001/CIM-
schema-cim10#SynchronousMachineOperatingMode.generator"/>
  <cim:RegulatingCondEq.RegulationSchedule rdf:resource="#_ca32746f-a002-4c2b-bcf4-
7bc97430bf87"/>
  <cim:Equipment.EquipmentContainer rdf:resource="#_6cb8701a-12f1-4de9-9e68-
125d95073a75"/>
</cim:SynchronousMachine>
```

- L'élément définition introduit une nouvelle ressource et indique son type. Il existe deux formes: la première en tant qu'attribut `rdf:ID` et la deuxième en tant qu'attribut `rdf:about`.
- Le type d'élément, *classname*, est le nom qualifié en XML d'une classe du schéma CIM ou d'un autre schéma déclaré comme un espace de noms dans l'élément document.
- La valeur de l'attribut `id`, *identity*, est choisie par la mise en œuvre. Elle doit être unique dans le document. (Elle n'est pas nécessairement liée au nom de la ressource du réseau électrique.)

6.2.3.6 Élément Description

```
<rdf:Description rdf:about=resource-uri>
  <!-- Content:
    (literal-property|resource-property|compound-property) *
  -->
</rdf:Description>
```

Exemple:

```
<rdf:Description rdf:about="#_26cc8d71-a002-4c2b-bcf4-7bc97430bf87">
  <cim:IdentifiedObject.name>TROY</cim:IdentifiedObject.name>
</rdf:Description>
```

- L'élément description ajoute des informations relatives à une ressource introduite n'importe où dans ce document ou dans un autre document.
- La *resource-uri* est une référence URN qui identifie la ressource sujet.
- L'élément Description est utilisé uniquement dans les modèles de différence (voir 6.2.4). Il n'est jamais utilisé dans les modèles intégraux.

6.2.3.7 Élément Compound

```
<classname>
  <!-- Content:
    (literal-property|resource-property|compound-property) *
  -->
</classname>
```

Exemple:

```
<cim:DateTimeInterval>
  <cim:DateTimeInterval.start>2013-02-28</cim:DateTimeInterval.start>
```