

# **CIM XML 模型交换格式**

## **CIM XML Model Exchange Format**

**2005年8月**

# 目 录

1. 范围.....	1
2. 参考的标准.....	1
3. CIM XML 格式和约定 .....	1
3.1 概述.....	1
3.2 简化的 RDF 语法.....	3
3.2.1 概述.....	3
3.2.2 注记符号.....	3
3.2.3 语法定义.....	4
3.2.3 部分模型的传输.....	8
3.3 CIM XML 格式风格导则 .....	9
3.4 从 CIM 视图到 CIM RDF 子集生成.....	10
3.5 CIM 扩展.....	10
附录 A: RDF 简化语法设计原理.....	12
附录 B: CIM XML 文档示例 .....	13
B.1 例子.....	13
B.2 CIM XML 文档引用例子 .....	13
参考文献.....	15

## 1. 范围

国际标准 IEC61970 为能量管理系统应用程序接口定义了一套组件接口规范（CIS）。本部分规定了基于 CIM 的模型信息交换的格式和规则。它使用第 501 部分中描述的 CIM RDF 模式作为元模型框架来构建电力系统信息模型的 XML 文档。这些文档的风格被称为 CIM XML 格式。

这个标准支持独立供应商提供的软件可以生成和使用基于公用格式的描述 CIM 的模型信息。提出的解决方案为：

- 该信息模型既可以适合机器读取也可以适合人工读取，但主要的目的是程序可读。
- 可以被支持文档对象模型(DOM)的任何工具读取或者被其它标准 XML 应用程序接口读取。
- 自描述
- 利用当前万维网联盟（W3C）建议的成果。

该标准是接口规范的第二层次，以描述的方式（用基于 CIM 的文本和例子来）来详细定义 CIM XML 格式。

## 2. 参考的标准

IEC 61970-1, EMSAPI – 部分 1: 导则和基本要求

IEC 61970-2, EMSAPI – 部分2: 词汇

IEC 61970-301, EMSAPI – Part 301: 公共信息模型（CIM）基础

IEC 61970-302, EMSAPI – Part 302: 公共信息模型（CIM） 财务、能量计划

IEC 61970-303, EMSAPI – Part 303: 公共信息模型（CIM） SCADA

## 3. CIM XML 格式和约定

### 3.1 概述

根据 IEC61970 501 标准所描述的 CIM RDF 模式，一个 EMS 电力系统模型

能被转换导出为一个 XML 文档，参看图 1。这个文档称为 CIM XML 文档。CIM RDF 模式提供了 CIM XML 文档所使用的资源描述格式。最终的 CIM XML 模型交换文档能被解析，其中的信息将被导入到一个外部系统。

约定使用由 <http://groups.yahoo.com/group/cimxml/> 下载的 [cim10\\_030501\\_rdf.rdf](#) 作为符合 IEC61970 501 标准的 CIM RDF 模式。或由 <http://groups.yahoo.com/group/cimxml/> 下载的 [cim10\\_030501.mdl](#) 转换的 [cim10\\_030501.xml](#) 作为符合 IEC61970 501 标准的 CIM RDF 模式。

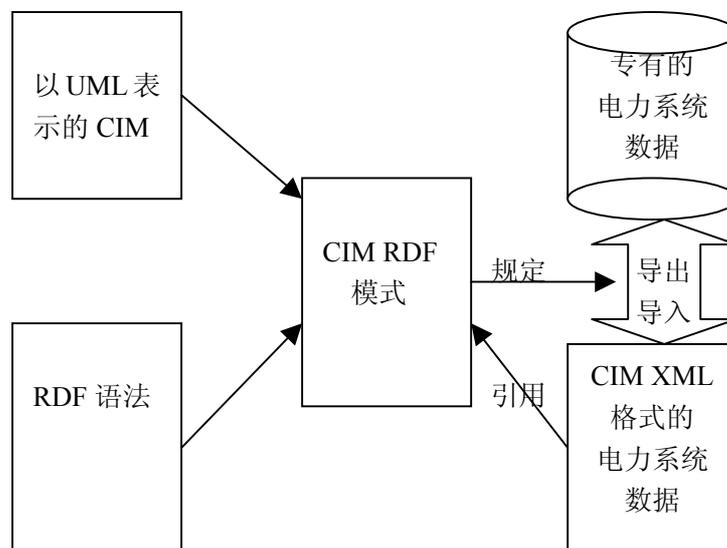


图 1 基于 XML 的 EMS 数据的交换机制

类似于使用任何编程语言，创建一个 CIM XML 文档可以有很多种方法可供选择。RDF 语法本身可以有多种用法而生成相同的基本结果。一种处理 CIM RDF 模式的方法在生成一个 CIM XML 文档的时候可以变化出不同的形式。下面的子节将要讨论生成 CIM XML 文档的风格规则。在生成这些 CIM XML 文档时，这些规则对于各个互操作的厂家进行沟通和理解是很重要的，因为它们简化和帮助编写一个清晰的解释模型信息的软件。

## 3.2 简化的 RDF 语法

### 3.2.1 概述

RDF 语法提供了许多种方法来表达同样的一组数据。例如，两个资源之间的关联可以写成为一个资源属性，或者将一个元素嵌入到另一中，这样会使得使用一些 XML 工具(如 XSL 处理程序)处理 CIM XML 文档出现困难。

因此,仅使用一个 RDF 语法的子集来创建 CIM XML 文档。这个语法简化了实现者构建模型序列化和序列软件，同时也提高了处理 CIM XML 文档时常规工具的效率。简化了的语法是标准 RDF 语法的一个合理的子集；这样，文档能够被现有的 RDF 序列软件读取。

下面的子节定义了一个 RDF 语法的子集。这个简化了的语法是为各个企业之间交换电力系统模型信息用的。其目标是使实现者可以容易的创建 RDF 数据的反序列化软件，减少序列化 RDF 数据时的选择，提高在序列化 RDF 数据时常规的 XML 生成工具(如 XSLT 处理器)的效率。

简化的语法并没有牺牲任何 RDF 数据模型的功能。也就是说，任何 RDF 数据可以使用这种语法来进行交换。此外，它保留了诸如一个模型可以在一个文档里定义而在另外一个文档里被保持和使用的 RDF 的特征。

### 3.2.2 注记符号

下面的若干节定义了简化语法。每一类元素都在一个子节中定义，以该元素的模型开始，接着是一些定义文本和 RDF 语法的引用。元素的语义并没有细化，可以参考 RDF 建议[3]获取这些信息。关于语法模型的助记符号如下：

- 1.在元素类型、属性名字或者属性值的位置以斜体符号表示,以指示所需要的名字或值的类型。符号将以文本形式进行定义。

- 2.*rdf* 符号作为前缀代表了实现时所选择 RDF 名字空间。类似地 *cim* 符号作为前缀代表选择了 EPRI CIM 名字空间。

- 3.元素模型中的注释指出了允许的内容。斜体符号代表了一类元素或者其它以文本方式定义的内容。一个 (a | b) 的结构指示了 a 和 b 是可选的。一个 a\* 指示了 0 个或者多个 a 的复制。

4.模型中其它文本均是文字的。

### 3.2.3 语法定义

#### 3.2.3.1 概述及XML文档结构的简介

在语法定义中用例子作进一步说明。这些例子将帮助较好的理解形式语法的定义。同一个例子可以用于几个语法定义。在例子中关注的语法将以粗体指示。例子以 xml 助记符号来表达。

下面对 XML 文档的结构作一个简单的介绍：

1. 一个 XML 文档是一组容器。容器可以包含其它的容器以及内容。良构的 XML 文档要求由两块组成，一是序言，一是根元素。序言包含指出 XML 标准的版本号以及使用的编码的声明。CIM XML 文件的序言为：

```
<?xml version="1.0" encoding="UTF-8"?>
```

2. 根元素包含了 XML 文档的实际内容，这些内容包含在起始和结束标记之中。CIM XML 文档的根元素为：

```
<rdf:RDF>
```

```
.  
:
```

```
</rdf:RDF>
```

3. 根元素也包含对名字空间的引用。名字空间为文档中使用的其它元素定义了一个上下文环境。元素的名称的前面均需要前缀一个名字空间的缩写并紧跟在冒号之后。例如：RDF语法的名字空间以及CIM模式的名字空间声明为：

```
xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
```

```
xmlns:cim="http://iec.ch/TC57/2003/CIM-schema-cim10#">
```

在文档中对名字空间的引用使用的语法是：

```
<cim:Company rdf:ID="_C70BB64F21C747E9A2B04CC2529751B0">
```

4. 注释可以出现在 XML 文档的任何地方，用<!-- and -->来界定。

5. 元素： 一个元素是在起始和结束标记之间的一个容器。起始标记定义了元素的名字以及该元素包含的任意的属性。在以下的例子中 **Company** 是元素的名字，而 **ID** 是它的属性。

```
<cim:Company rdf:ID="_C70BB64F21C747E9A2B04CC2529751B0">
```

而结束标记则是</cim:Company>。

6. 属性： 对于 Company 元素的 ID 属性可以被 XML 文档中的其它元素所引用。具有 ID="\_C70BB64F21C747E9A2B04CC2529751B0"的元素可以被其它元素所引用，但引用时需要使用#字符加在 ID 属性值的前面如：“#\_C70BB64F21C747E9A2B04CC2529751B0”。

### 3.2.3.2 文档元素

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

例如：

```
<?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/TC57/2001/CIM-schema-cim10#">
```

</ rdf:RDF >

1.该元素类型是rdf:RDF，用于定义CIM XML文档。

2.RDF名字空间必须被声明为http://www.w3.org/1999/02/22-rdf-syntax-ns#

3.CIM名字空间必须被声明，包括其版本信息。格式为：  
http://iec.ch/TC57/2001/CIM-schema-*<version>*#。当前的版本是cim10。

4.可以声明其它的名字空间。

可参考RDF语法条款:6.1。

### 3.2.3.3 定义元素

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

例子：

```
<cim:SynchronousMachine rdf:ID="_31DFC4296BFB4E2EB299642491B3ABC1">
<cim:Naming.name>IN-2</cim:Naming.name>
<cim:SynchronousMachine.minimumMVAr>-9999</cim:SynchronousMachine.minimumMVAr>
  <cim:SynchronousMachine.operatingMode
```

```

rdf:resource="http://iec.ch/TC57/2001/CIM-schema-cim10#SynchronousMachineOperating
Mode.generator"/>
    <cim:RegulatingCondEq.RegulationSchedule
rdf:resource="#_CA32746FA0024C2BBCF47BC97430BF87"/>
    <cim:Equipment.MemberOf_EquipmentContainer
rdf:resource="#_6CB8701A12F14DE99E68125D95073A75"/>
</cim:SynchronousMachine>
  
```

1. 定义元素引入一个新的资源并给出它的类型。可以用于定义CIM 模式中类的实例。

2. 该元素类型为：*classname*，是一个CIM模式中类的合法的XML名字，或是在文档元素中作为名字空间所声明的其它模式中类的合法的XML名字符号。

3. id 属性的值为：*identity*，是一个被实现所选择的XML名字符号。它在文档中必须是唯一的。（并不要求它和电力系统资源名有关系。）

可参考RDF语法条款：6.13。

### 3.2.3.4 描述元素

```

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

例子：

```

<rdf:Description rdf:about="#Substation-2">
  <cim:Naming.name>TROY</cim:Naming.name>
</rdf:Description>
  
```

1. 描述元素增加了在本文档或其它文档引入的资源的信息。

2. *Resource-uri*是一个URI引用，标识了主题资源。（可以被简化为本文档中标识资源的片段（fragment）部分。）

可参考RDF语法条款：6.13。

### 3.2.3.5 文字-属性元素

```

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

例子：

```

<cim:SynchronousMachine rdf:ID="_31DFC4296BFB4E2EB299642491B3ABC1">
  
```

```

    <cim:Naming.name>IN-2</cim:Naming.name>
  <cim:SynchronousMachine.minimumMVAr>-9999</cim:SynchronousMachine.minimumMVAr>
    <cim:SynchronousMachine.operatingMode
rdf:resource="http://iec.ch/TC57/2001/CIM-schema-cim10#SynchronousMachineOperating
Mode.generator"/>
    <cim:RegulatingCondEq.RegulationSchedule
rdf:resource="#_CA32746FA0024C2BBCF47BC97430BF87"/>
    <cim:Equipment.MemberOf_EquipmentContainer
rdf:resource="#_6CB8701A12F14DE99E68125D95073A75"/>
  </cim:SynchronousMachine>

```

1. 文字-属性元素引入了一个属性和一个应用于本资源的一个文字值。用于定义CIM 模式中类的实例的属性。

2. 该项类型 *propname*，是一个来源于EPRI CIM模式中属性的合法的XML名字，或在文档项中所声明的名字空间下的其它模式中属性的合法的XML名字。

3. 内容 *text* 是任何不带 <、> 和 & 符号的XML文本，表达了属性的值。

可参考RDF语法条款:6.12。

### 3.2.3.6 资源-属性元素

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

1. 资源属性项引入了一个属性和作为当前资源的属性值的一个资源。用于定义 CIM 模式的实例中的枚举列表以及与其它实例间的关联、聚集等关系。

2. 该项类型 *propname*，是一个来源于 EPRI CIM 模式中的合法的属性名字，或在文档项中所声明的名字空间下的其它模式中的合法的属性名字。

3. Resource-uri 是一个 URI 引用，它标识了一个资源。

可参考 RDF 语法条款： 6.12。

例1 URI-引用：

这个例子包含了两个引用,一个是对RegulationSchedule的引用,另外一个是对以EquipmentContainer表示的父资源的引用。

```

<cim:SynchronousMachine rdf:ID="_31DFC4296BFB4E2EB299642491B3ABC1">
  <cim:Naming.name>IN-2</cim:Naming.name>

  <cim:SynchronousMachine.minimumMVAr>-9999</cim:SynchronousMachine.minimumM

```

```

VAr>
  <cim:SynchronousMachine.operatingMode
rdf:resource="http://iec.ch/TC57/2001/CIM-schema-cim10#SynchronousMachineOperating
Mode.generator"/>
  <cim:RegulatingCondEq.RegulationSchedule
rdf:resource="#_CA32746FA0024C2BBCF47BC97430BF87"/>
  <cim:Equipment.MemberOf_EquipmentContainer
rdf:resource="#_6CB8701A12F14DE99E68125D95073A75"/>
  </cim:SynchronousMachine>
  
```

## 例2 枚举列表

这个例子定义了同步机运行模式（SynchronousMachine.operatingMode）的属性值为发电机（generator）。在CIM模式中，运行方式（operatingMode）属性是作为SynchronousMachineOperatingMode的枚举数列表进行规定说明的。因此用

“<http://iec.ch/TC57/2001/CIM-schema-cim10#SynchronousMachineOperatingMode.generator>”的形式来表达。其中

<http://iec.ch/TC57/2001/CIM-schema-cim10>是说明CIM模式的，而

SynchronousMachineOperatingMode.generator是枚举列表

SynchronousMachineOperatingMode中的一个枚举值generator。

```

<cim:SynchronousMachine rdf:ID="_31DFC4296BFB4E2EB299642491B3ABC1">
  <cim:Naming.name>IN-2</cim:Naming.name>
<cim:SynchronousMachine.minimumMVar>-9999</cim:SynchronousMachine.minimumM
VAr>
  <cim:SynchronousMachine.operatingMode
rdf:resource="http://iec.ch/TC57/2001/CIM-schema-cim10#SynchronousMachineOper
atingMode.generator"/>
  <cim:RegulatingCondEq.RegulationSchedule
rdf:resource="#_CA32746FA0024C2BBCF47BC97430BF87"/>
  <cim:Equipment.MemberOf_EquipmentContainer
rdf:resource="#_6CB8701A12F14DE99E68125D95073A75"/>
  </cim:SynchronousMachine>
  
```

### 3.2.3 部分模型的传输

部分模型的传输可以不需要任何附加的语法定义。仅只要求：所有的链接到基本模型文件中的类的实例的新的关联（associations）必须包含在部

分模型文件之中，并要求这些包含在部分模型的关联带有基本模型文件中的类的实例的正确的资源标识（Resource ID）。

### 3.3 CIM XML 格式风格导则

RDF 语法一个有用的特征就是它允许电力系统模型的任意子集在文档中序列化。然而，这是一把双刃剑。某方生成的文档如果没有包含另外一方所需的所有属性，则对于另外一方可能是不可用的。而且，一个包含了部分模型的文档如果其资源 URI 与其它文档不一致的话，它也是不可用的。

下面的导则适用于 CIM XML 文档的内容，以促使能够在最大的应用范围中使用该文档。

1. 在每一个资源引入的时候最好是包含主键属性。例如，`cim:Naming.name` 和 `cim:Equipment.MemberOf_EquipmentContainer` 属性即是所需要的属性。

原因：大的应用类想要将模型数据装载入数据库，而许多数据库模式在插入时需要主键值。

2. 包含单值属性而不是它们反向的多值属性。

例如，使用 `cim:Equipment.MemberOf_EquipmentContainer` 而不是 `cim:EquipmentContainer.Contains_Equipments`。

原因：因为这些属性是双向的，一个关于一方的声明暗示了反向的另一方。只包含一方可以减少错误发生，简化文档的编辑和转换。

3. 在遇到多对多的关联的时候，一般有一个引用的主方向。包含主要的引用而不是它们多值的反向引用。例如，使用 `cim:SynchronousMachine.MVArCapabilityCurves`，而不用 `cim:MVArCapabilityCurve.SynchronousMachines`，因为主要的关系是从 `SynchronousMachine` 到 `MVArCapabilityCurve`。

原因：和上面 3 条相同的原因。

4. 当遇到一个单值关联到一个反向单值，可以包含任意一个，但不要同时包含两个。导入工具需要能够处理任意方向的关联，并能补齐反向关联。

原因：因为这些属性是双向的，一个方向上的引用声明暗示了另一个方向上的关联。这将减少出错的可能，并且可证明，将使得编辑和转换文档更加容易。

5.在指定 `rdf:ID` 的时候使用 XML 名字约定规则。也就是说,“一个名字是一个以字母或少许标点符号中的一个符号开头的,紧跟着字符,数字,连字符,下划线,冒号或者句号共同组成的作为名字字符串的标记” [7]

原因:符合 XML 约定规则。尽管这样的标识符可能由一个编号系统命名,但它们应该以一个字母开头。

6.在一个模型分散在多个文档或者使用多个版本的模型的时候对于 `rdf:ID` 应该使用稳定的标识符。一个稳定的标识符是这样的,它不会被重用也从不会改变。从随机数导出标识符是一种减少偶然重用的方法。基于属性值的标识符(例如一个名字)和顺序号获得的标识符容易被改变。

原因:这将会减少文档间引用时的不一致的错误。

### 3.4 从 CIM 视图到 CIM RDF 子集生成

61970 标准系列的第 501 部分讨论了 CIM RDF 模的生成。CIM XML 模型交换文档使用 CIM 的一个子集来满足一个特定用例(参看第 4xx 篇, CIM XML 模型交换规范)的模型交换的需求。一个 CIM 子集定义了 CIM XML 文档导入/导出者应该处理的 CIM 的某一部分。

CIM UML 中的每一个类、属性和关联能被标记上一个相关的视图标识符。例如,在“传输安全协调交换”的用例中,一个“NERC”视图标识指出了 CIM 特定部分是该交换用例子集的一部分。

CIM RDF 模式生成器使用 CIM 其中的一部分将子集信息转换为一个 CIM RDF 模式文档。例如,NERC 子集在每一个相应的类、属性或者关联之后读取。这个信息可以使软件用来只读取它感兴趣的部分;举例来说, CIM XML 验证工具能够就 CIM XML 文档的一部分进行 CIM 检查。

### 3.5 CIM 扩展

XML 中的 X 代表了可扩展(eXtensible)。这就意味着 CIM XML 文档能够为模型提供厂商或者为企业的特别需求进行扩展。CIM RDF 模能够通过提供一个不同的名字空间扩展新类和属性。因为如果使用一个不同的名字空间,用户定制的 CIM XML 文档清晰地区分了哪些是 CIM 标准,哪些是用户定制的。若干种

不同的定制扩展能够在同一个 XML 文档中存在并被清晰地辨识。当这些用户定制的文件导入到并不能识别扩展的信息系统中时，带有不认识的标签的元素将被忽略掉。下面的声明标识了一个扩展的名字空间 **bpa**。

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

例如，我们可以增加一个非 CIM 属性，**OriginalPO**，到 **Breaker** 类中，正如下面显示的。对于对这个扩展不感兴趣的导入程序将会忽略掉这个为 **BPA** 扩展的定制标签。

```
<cim:SynchronousMachine rdf:ID="_31DFC4296BFB4E2EB299642491B3ABC1">
  <cim:Naming.name>IN-2</cim:Naming.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="#_CA32746FA0024C2BBCF47BC97430BF87"/>
  <cim:Equipment.MemberOf_EquipmentContainer
  rdf:resource="#_6CB8701A12F14DE99E68125D95073A75"/>
</cim:SynchronousMachine>
```

相应于这个扩展的 RDF 模式能被加入到一个单独的 RDP 模式文档中，因此保持 CIM RDF 模式被清晰地区别开并允许其中的每一个独立变化。

## 附录 A：RDF 简化语法设计原理

下面几点解释了在本简化语法中所做的选择

1.文字属性可以用特性属性来表达（RDF 语法条款 6.10）。这将使得结构更加紧凑。然而，使用属性元素是因为它们更容易以 XSLT 表达式方式进行处理。（例如，它们可以被分类。）它们也更容易表达为多行文本。

2.语法是扁平的两层的资源 / 属性结构。多层的嵌套结构可以比较简洁。而且，一个精选的嵌套结构可以允许公用的查询更容易地被编码为 XSLT 表达式。另一方面，扁平结构因为是最简单的，并且易于产生和解释，因而被选用。通过避免任何关于嵌入结构细节对应用的依赖，它应该是一个移植性更好的语法。

3.所有的资源都在它们被引入（通过定义元素）的时候给出一个类型。然而，RDF 模型允许一个资源没有类型。在现在的应用中，无类型的资源是不需要的。

## 附录 B： CIM XML 文档示例

### B.1 例子

下面的例子是取自“传输安全协调交换” CIM XML 文档。它以简化 RDF 语法展示了两个资源（Substation 和 EnergyConsumer）

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:cim="http://iec.ch/TC57/2000/CIM-schema-cimu09b#">

  <cim:Substation rdf:ID="_28159176D77E2467">
    <cim:Naming.name>WALDEN</cim:Naming.name>
    <cim:Substation.MemberOf_SubControlArea
      rdf:resource="#_55D3DE366B2AD032" />
  </cim:Substation>
  <cim:EnergyConsumer rdf:ID="_2963867E4A4B1669">
    <cim:EnergyConsumer.pfixed>89.59</cim:EnergyConsumer.pfixed>
    <cim:EnergyConsumer.qfixed>20.42</cim:EnergyConsumer.qfixed>
    <cim:EnergyConsumer.LoadArea rdf:resource="#_9C1602456B178B75" />
    <cim:ConductingEquipment.Terminals
      rdf:resource="#_A920F93C9D23DD45" />
    <cim:Equipment.MemberOf_EquipmentContainer
      rdf:resource="#_A9D1427B3784CD78" />
    <cim:Naming.name>610</cim:Naming.name>
  </cim:EnergyConsumer>

</rdf:RDF>
```

### B.2 CIM XML 文档引用例子

一个“传输安全协调交换”的 CIM XML 示例文档可以从一个开源的 CIM XML Validator 工具中找到。请查看下一节，附录 C。

## 附录 C：CIM XML 验证工具

为了推动互操作试验建立了一个工具，它可以验证 CIM XML 文档是否使用良构 XML，它约束文档使用本标准描述的简化格式，它可以正确地引用 CIM RDF 模式。

这个工具遵循开放源代码协议。也就是说，任何人都可以使用这个工具；然而，任何改进都必须保证是可行的并满足开放源代码协议。

请参看 Source Forge 上的 Xmodel 项目。你可以从那个站点下载 CIMValidate 软件。同时还有“传输安全协调交换”的示例 CIM XML 文档。

<http://xmodel.sourceforge.net>

使用这个工具的方法也可以在这个站点上获取。

## 参考文献

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*, CIM XML 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.rfc2396).
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>
11. Draft IEC 61970: Energy Management System Application Program Interface (EMS-API) –Part 503: CIM XML Model Exchange Format Revision 3b 2004-04-09