



基于 IEC61850 的 ASN.1 编码解码算法研究

Research the Algorithm of ASN.1 Encoding/Decoding Based on IEC61850

第五柯鹏

Diwu Kepeng

(广东公诚通信建设监理有限公司, 广东 广州 510610)

(Guangdong Gongcheng Telecommunication Project Management Co.Ltd, Guangdong Guangzhou 510610)

摘 要: 本文首先介绍了 ASN.1 以及其几种编码规则。按照变电站通信网络和系统系列标准 IEC 61850 应用层消息结构与 ASN.1 之间的对应关系,介绍了应用于变电站自动化通信系统中的 ASN.1 编解码模块的设计与实现。

关键词: 抽样描述文法(ASN.1);IEC61850;编解码器

中图分类号: TN914

文献标识码: A

文章编号: 1671-4792-(2010)7-0020-03

Abstract: The paper first briefly introduces ASN.1 and its encoding/decoding rules. In term of the corresponding relation of ASN.1 and IEC61850 (Communication networks and system standards in substations) application layer's message structure, it explains the design and realization of ASN.1 encoder/decoder module which is applied in communication system in substations.

Keywords: ASN.1 (Abstract Syntax Notation 1); IEC61850; Encoder/Decoder

0 引言

IEC 61850^[1]作为未来电力系统无缝通信体系的基础,其底层映射到 MMS,可见 MMS 是实现 IEC 61850 通信协议的一个基础。MMS 标准采用 ASN.1 作为其描述工具,ASN.1 的作用就是具体描述各种协议标准,使人们既可以清楚简洁地表示协议中传递的各种信息,又能容易实现相应的协议标准。而编解码主要是完成 MMS PDU 的编码、解码过程,以消除异种机之间数据表示的二义性。

1 抽象描述文法 ASN.1

OSI 定义抽象对象的方法为 ASN.1^[2,3],把这些对象转换成“0”和“1”的比特流的一套规则成为 BER (Basic Encoding Rules)。ASN.1 是一套灵活的记号,它允许定义多种数据类型,从 Integer、bit string 一类的简单类型到结构类型,如 set 和 sequence,还可以使用这些类型构建复杂类型。BER 说明了如何把每种 ASN.1 类型的值编码为 8bit 的 octet 流。通常每一个值有不止一种的 BER 编码方法。一般使用另外一套编码规则 DER,它是 BER 的一个子集,对每个 ASN.1 值只有唯一一种编码方法。

1.1 ASN.1 数据类型

ASN.1 中定义的简单类型有 BOOLEAN、INTEGER、REAL、BIT STRING、OCTET STRING、ENUMERATED、NULL、

OBJECT IDENTIFIER; 结构类型包括 SEQUENCE、SEQUENCE OF、SET、SET OF、CHOICE。

1.2 编码规则

ASN.1 的基本编码规则 BER 采用的编码结构由 Identifier (标识符)、Length (长度)以及 Content (内容)三部分构成,如表一所示。

表一 基本编码格式

标识符字节 (Identifier)	数据长度字节 (LENGTH)	内容字节 (VALUE)
--------------------	-----------------	--------------

(1) Identifier 标识符字节

bit8~bit7 表示 Tag 的类型,ASN.1 中定义了 4 种 Tag 类型,即 Universal (通用类)、Application (应用类)、Context-Specific (上下文确定类)和 Private (专用类)。其分配格式如表二所示:

表二 标识符类型编码

Bit8	Bit7	类型
0	0	Universal
0	1	Application
1	0	Context-Specific
1	1	Private

bit6 表示 Tag 内容字节的结构。如果为 0,表示内容字节为简单结构,即只包含数值内容;如果为 1,则表示内容字节为复合结构,即内容字节还包含 T-L-C 结构,直至结构中标

标识的 bit6 为 0。

bit5~bit1 表示 Tag 的值,也就是说 Tag 的值只能是从 0 到 31。所以标识符字节一般由单个字节组成,当 PDU 的序号超过 31 时,标识符可以进行扩展为两个字节。其中第一个八位位组的 bit8~bit6 仍然是 Tag 类型的标记,而 bit5~bit1 用全 1 表示,后面一个字节表示 Tag 的值,这样 Tag 的值就可以表示 0~255。

(2) Length 长度字节。

长度字节分为定长和非定长两种格式,其中定长格式又分为短型和长型。如果标识符的 bit6 为简单结构,采用定长格式;如果标识符的 bit6 为复合结构并且全部立即可用,则采用定长格式;如果标识符的 bit6 为复合结构而非全部立即可用,则采用非定长形式。

(3) Value 字节

对值本身进行编码。在简单类型的编码中,该字节表示实际的内容;对于结构类型来说,该字节是一个或多个 TLV (Tag/Length/Value)形式。

2 ASN.1 编/解码的设计与实现

协议数据单元(PDU)的 ASN.1 描述是独立于任何环境的一种抽象描述,在具体实现时用和编程语言有关的数据结构来表示的,本文采用的是 C++ 语言来实现的。因此每一个 PDU 都存在一个 ASN.1 抽象定义和一个相应的数据结构。每一个 PDU 都被分配一个 Tag 编解码器,要能对一个数据编码,必须知道这个数据的类型和值。对于 PDU 来说,它自身的类型信息由它的 Tag 提供,而 PDU 中所包含的各个数据类型信息在该 PDU 的 ASN.1 定义中,各个数据的值包含在其相应的数据结构中。

2.1 编解码模块实现

目前实现 ASN.1 编解码的工作有软件和硬件两种实现方案。硬件编解码一般处理速度快,但是维护、升级等都比较困难。软件编解码可以采用第三方的平台或者编译器来做,如 ESNACC、ASN1C 等,但是由于其通用性而导致占用资源增多,存在着许多冗余^[9]。本系统采用的是软件实现,借鉴通用工具的同时去除其中的冗余,达到精简的目的。

经 ASN.1 编译后,ASN.1 消息转换成 C/C++ 的消息。与 ASN.1 标记相对应,翻译为树型嵌套结构,层层嵌套,一直到 ASN.1 的基本数据类型对应的 C/C++ 结构^[9]。此外还有它所包含的各参数的 ASN.1 标记特征值的配置,这些配置标识了各参数属于何种 ASN.1 基本类型。完成翻译的结果保持在.h 和.c 文件中。

下面是一个具体的应用:

```
BCAS DEFINITIONS ::= BEGIN
BBCard ::= SEQUENCE {
    name IA5String (SIZE (1..60)),
    address IA5String (SIZE (1..60)),
    index INTEGER (1..100),
    position IA5String (SIZE (1..60)),
    Domainstate ENUMERATED
    {good(0), bad(1), well_balanced(2)},
}
myCard BBCard ::= {
    name "Casey",
    address "Jinjiu Line",
    index 32,
    position "nanchang station",
    Domainstate good,
}
END
```

生成所对应的 C++ 结构如下:

```
struct EXTERN BBCard_Domainstate {
enum Root {
    good= 0,
    bad = 1,
    well_balancde = 2
};
};
struct EXTERN ASN1T_BBCard : public ASN1TPDU {
    ASN1IA5String name;
    ASN1IA5String address;
    OSUINT8 index;
    ASN1IA5String position;
    ASN1T_BBCard_Domainstate Domainstate;
    ASN1T_BBCard ();
    ~ASN1T_BBCard ();
};
```

编码函数有两个,分别是 asn1E_BBCard_Domainstate() 和 asn1E_BBCard():

```
EXTERN int asn1E_BBCard_Domainstate (ASN1CTXT*
pctxt,ASN1T_BBCard_Domainstate *pvalue, ASN1TagType
tagging);
```

```
EXTERN int asn1E_BBCard (ASN1CTXT* pctxt,
ASN1T_BBCard *pvalue, ASN1TagType tagging);
```

解码函数也对应有两个, 分别是 `asn1D_BBCard_Domainstate()` 和 `asn1D_BBCard()`:

```
EXTERN int asn1D_BBCard_Domainstate (ASN1CTXT*
pctxt,ASN1T_BBCard_Domainstate *pvalue, ASN1TagType
tagging, int length);
```

```
EXTERN int asn1D_BBCard (ASN1CTXT* pctxt,
ASN1T_BBCard *pvalue, ASN1TagType tagging,int length);
```

其中第一个参数是上下文结构变量, 第二个参数传递实际变量用于编码解码, 第三个参数指定使用隐式还是显式的标签, 解码函数的第四个参数表示长度。

2.2 内存分配与管理模块

内存函数主要负责内存的分配与管理。

本系统采用的内存管理方法是使用库内存管理函数进行分配和释放内存缓冲区。在执行中只需调用一个简单的 `asn1Free_BBCard ()` 函数就可以释放所有分配的内存, 当 `asn1Free_BBCard()` 被调用时, 所有内存立刻释放。具体的函数定义如下:

```
EXTERN void asn1Free_BBCard (ASN1CTXT *pctxt,
ASN1T_BBCard* pvalue) {
    if (pvalue==0) return;
    rtMemFreePtr (pctxt, pvalue->name);
    rtMemFreePtr (pctxt, pvalue->address);
    rtMemFreePtr (pctxt, pvalue->position);
}
```

3 结束语

ASN.1 是一种重要的形式化描述工具, 是描述应用层协议的主要工具。由于 ASN.1 描述的中立性和清晰性, A-Profile 中的表示层、应用层协议规范都采用了 ASN.1 作为其编/解码的规范。ASN.1 可以有多种传输语法的支持, 如 BER (基本编码规则)、CER (正规编码规则)、DER (特异编码规则)、PER (紧缩编码规则), 但是目前使用较多的还是 ASN.1 基本编码规则。基于 ASN.1 所建立的开发环境有效解决了应用层协议中数据单元的描述与编码问题。

参考文献

- [1] IEC61850:Communication networks and systems in substations-Part1-5(First edition)[S] .
- [2]ITU-T Recommendation X.680, Information Technology Abstract Systax Notation One(ASN.1):Specification of Basic Notation[R].2001.
- [3]ITU-T Recommendation X.690, Information Technology ASN.1 Encoding Rules[R].2001.
- [4]余鹰, 范辉, 见春蕾.基于 BER 的 ASN.1 编解码原理与设计实现[J].计算机工程与科学,2005,27(1).
- [5] 赵新宁, 王汝传.ASN.1 到 C/C++ 编译器实现的研究[J].南京邮电学院学报,2003,(6).

作者简介

第五柯鹏 (1980—), 男, 工程师, 主要研究方向: 计算机通信与网络控制。