MMS 协议

MMS 基础知识

MMS(Manufacture Message Specification)通过对实际设备进行面向对象建模的方法，实现了网络环境下不同制造设备之间的互操作。

IEC61850 标准把 MMS 引入电力自动化领域，将其核心 ACSI 服务直接映射到 MMS 标准。

MMS 规范位于 OSI 七层参考模型的第七层——应用层。

MMS 通信流程

MMS 通信采用客户端/服务器模式(C/S 架构);

MMS 服务可以分为带确认（Confirmed）和不带确认（UnConfirmed）两类。

MMS 的基本思想

MMS 的基本思想就是定义一种大家都能理解的公共语言。一些来自不同生产厂家的智能装置，这些装置可能运行着不同的操作系统和程序，硬件结构也可能存在千差万别，设置同一厂家不同时期生产的不同型号的装置在硬件和软件上也可能存在差异。

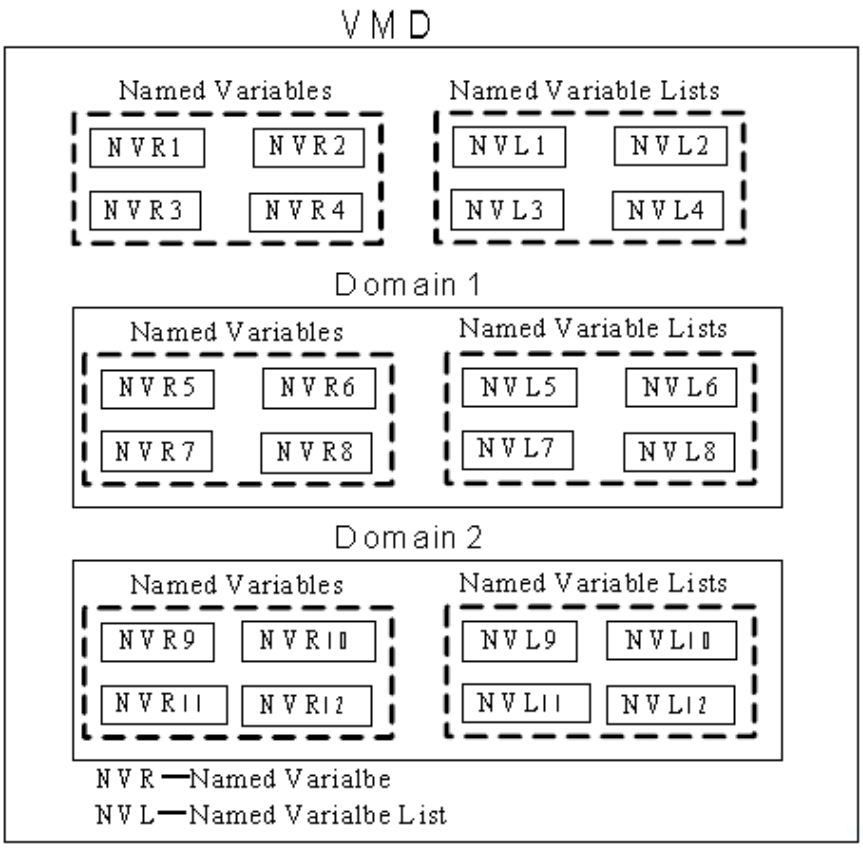
要达到互操作的目的，就必须先考虑“屏蔽”叼各装置的具体技

术细节（如 CPU 型号、操作系统、程序编程语言、开关量子系统），使各装置在网络通信方式上遵守同一的规定，具有相同或相近的外部接口。

MMS 采用构建虚拟设备（即 VMD）的方法来达到隐藏各类装置技术细节的目的。虚拟设备由实际设备映射得到，它和装置的具体细节无关，具体通信时信息交互在客户端和虚拟设备之间进行。这样就“屏蔽”掉了装置的技术细节，客户端就可以和不同厂家、不同型号的多种装置进行通信。

MMS 对象和服务

MMS 定义了 VMD（虚拟制造设备）、Domain（域）、NamedVariable（有名变量）、NamedVariableLists（有名变量列表）等对象模型，另外还定义了大概 80 多种服务



IEC61850 与 MMS 的对象映射

编号	IEC61850 对象	MMS 对象
1	服务器（Server）	虚拟制造设备（VMD）
2	逻辑设备（Logical Devices）	域（Domain Objects）
3	逻辑节点（Logical Nodes）	有名变量 （Named Variable Objects）
4	数据（Data）	
5	报告控制块（RCB）	
6	定值组控制块（SGCB）	
7	日志控制块（LCB）	

8	控制（Control）	
9	数据集（Data Set）	有名变量列表 （Named Variable List Objects）
10	日志（Logs）	日志（journal Objects）
11	文件（Files）	文件（Files）

IEC 61850 与 MMS 的服务映射

编号	IEC61850 对象	ACSI 服务	MMS 服务
1	服务器（Server）	GetServerDirectory	GetNamedList
2	逻辑设备（LDevice）	GetLogicalDeviceDirectory	GetNamedList
3	逻辑节点（LNode）	GetLogicalNodeDirectory	GetNamedList
4		GetAllDataValues	Read
5	数据（Data）	GetDataValues	Read
6		SetDataValues	Write
7		GetDataDirectory	GetVariableAccessAttribute
8		GetDataDefinition	GetVariableAccessAttribute

9	数据集（DataSet）	GetDataSetValues	Read
10		SetDataValues	Write
11		CreateDataSet	DefineNamedVariableList
12		DeleteDataSet	DeleteNamedVariableList
13		GetDataSetDirectory	GetNamedVariableListAttribute
14	报告控制块（RCB）	Report	InformationReport
15		GetBRCBValues	Read
16		SetBRCBValues	Write
17		GetURCBValues	Read
18		SetURCBValues	Write
19	控制（Control）	Select	Read
20		Select With Value	Write
21		Cancel	Write
22		Operate	Write
23		TimeActivatedOperate	Write
24		CommandTermination	InformationReport

25	文件（File）	GetFile	FileOpen，FileRead，FileClose
26		SetFile	ObtainFile
27		DeleteFile	FileDelete
28		GetFileAttributeValues	Sequence of FileDirectory
29	定值组控制块（SGCB）	SelectActiveSG	Write
30		SelectEditSG	Write
31		SetSGValues	Write
32		ConfirmEditSGValues	Write
33		GetSGValues	Read
34		GetSGCBValues	Read
35	日志控制块（LCB）	GetLCBValues	Read
36		SetLCBValues	Write
37		GetLogStatusValues	Read
38		QueryLogByTime	ReadJournal
39		QueryLogAfter	ReadJournal
40	取代（Substitution）	GetDataValues	Read
41		SetDataValues	Write

42	GSE / GOOSE	GetGoCBValues	Read
43		SetGoCBValues	Write
44	GSE / GSSE	GetGsCBValues	Read
45		SetGsCBValues	Write
46	关联（Associate）	Associate	Initiate
47		Abort	Abort
48		Release	Conclude

MMS 与 ASN.1 编解码

MMS 的编解码采用 ASN.1 规范。ASN.1 是抽象语法标记（abstract syntax notation one）的英文缩写，它位于 ISO/OSI 七层开放互联模型的第六层——表示层。它分为语法规则和编码规则两部分：语法规则用于描述信息对象的具体构成（格式），如数据类型、内容顺序或结构。编码规则定义了信息的具体编/解码语法。

ASN.1 基础

ASN.1 数据类型

ASN.1 中定义的数据类型既有简单的基本数据类型，也有复杂的

结构类型。

基本类型是不可再分的，具体包括布尔型（BOOLEAN）、整型（INTEGER）、枚举类型（ENUMERATED）、空类型（NULL），对象标识符（OBJECT IDENTIFIER）等。

除了基本类型，ASN.1 还定义了多种复杂的结构类型，例如：

（1）SEQUENCE：有序的数据结合，由不同的数据类型构成，强调内部成员的排序。

（2）SEQUENCE OF： 有序的数据集合，成员由为同一类型的数据，类似 C 语言的数组。

（3）SET：由不同类型的数据组成的集合，对成员的顺序不做要求。类似于 c 语言中的结构体类型。

（4）CHOICE：选择结构，在列出的内部成员中，只能选择其中之一。类似于 c 语言中的共用体类型。

基本编码规则 BER

ASN.1 提供了多种编码规则，如 BER、DER、CER、PER 等，IEC 61850 在 MMS 编/解码中使用的是 BER 基本编解码规则。

1. BER 编码结构

ASN.1 基本编码规则 BER 采用的编码结构由标记（Tag）、长度（length）以及内容（Value）三个部分构成，一般称为 TLV 结构，如表一所示：

表一：BER 编码结构

标记（Tag）	长度（Length）	内容（Value）

各部分含义如下：

（1）标记：描述数据的类型。

（2）长度：用于说明 Value 部分的长度。

（3）内容：数据的实际值。

基本编码规则采用 8 位位组（即一个字节）作为基本传送单位，因此 TLV 结构的三个部分都由一个或多个 8 位位组组成。

2.标记

标记 Tag 一共有四种类型，即通用类（Universal）、应用类（Application）、上下文相关类（Context）、和专用类（private）(IEC6185 中用到的只有前三个)。

表二是常用的标记分类，标记 Tag 通常由一个或者两个 8 位位组构成，bit7、bit6 用用于说明标记的类型，“00”代表通用类，“01”代表应用类，“10”代表上下文相关类，“11”代表专用类；bit5 说明数据是简单类型还是结构类型；bit4~0 是标记编号（Tag 值），不同的 Tag 值代表不同的数据类型，当标记编号大于 30 时，这 4 个 bit 都为 1，用下一个字节表示标记编号。

表二 常用的标记分类

Bit7~6	Bit5	Bit 0~4	说明	举例
00	0	Tag 值	简单类型，通用类	BOOLEAN、INTEGER
00	1		结构类型，通用类	SEQUENCE SEQUENCE OF
01	1		结构类型，应用类	-----
10	0		简单类型，上下文相关类	IMPLICIT
10	1		结构类型，上下文相关类	IMPLICIT SEQUENCE

表三给出了 ASN.1 规范定义的一批简单通用类数据的 Tag 值，例如

“01h”代表 BOOLEAN 类型，“02h”代表 INTEGER 类型。

表三 ASN.1 中简单通用类数据的 Tag 值

Tag 值(16 进制)	类型	Tag 值(16 进制)	类型
01	BOOLEN	10	SEQUENCE,SEQUENCE OF
02	INTEGER	11	SET,SET OF
03	BIT-STRING	12	NumericString
04	OCTETSTRING(8)	13	PrintableString
05	NULL	14	TeletexString, T61String
06	OBJECT IDENTIFER	15	VideotexString
07	ObjectDescripion	16	IA5STRING
08	EXTERNAL,INSTANCE OF	17	UTCTIME
09	REAL	18	GENERALIZERIME
0A	ENUMERATED	19	GraphicString
0B	EMBEDDED PDV	1A	VISIBLESTRING
0C	UFT8String	1B	GeneralString
0D	RELATIVE-OID	1C	UniversalString
0E	保留	1D	CHARACTER STRING
0F	保留	1E	BMPString

ASN.1 定义的这些 Tag 值无法完全满足 MMS 应用的需要，如无法区分整形（INTEGER）和无符号整型（unsigned）两类数据。因此 MMS 扩展定义了一组专用的 Tag 值（主要针对简单类型和上下文相关类），如表四所示：

表四 MMS 定义的上下文相关类 Tag 值(对简单通用类数据的扩展)

Tag 值(16 进制)	类型	Tag 值(16 进制)	类型
81	数组 array	87	Floating-poing
82	结构体 struct	89	Octet-string
83	boolean	8A	Visible-string
84	Bit-string	8C	timeofday
85	integer	8D	bcd
86	unsigned	8E	boolarray

在 ASN.1 描述文本中，MMS 自定义的上述 Tag 值需要在类型前使

用“[]”来声明，[]中的值是被声明类型的实际 Tag 值。

3.长度

长度字段用于指明内容 Value 部分所含 8 位位组的数目，不包括标记和长度本身的 8 位位组数目。

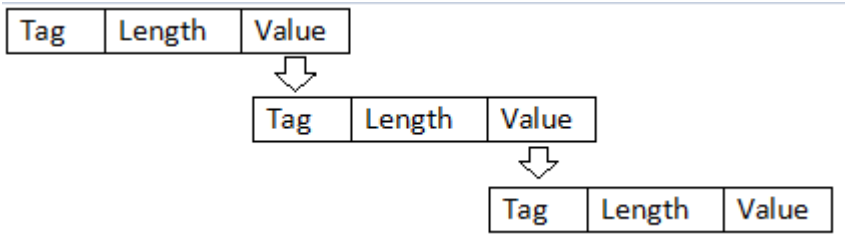
如果 Value 所含的 8 位位组数目（以后用字节数表示）小于等于 127，那么 Lenth 只占一个字节，最高位 bit7 为 0，bit6~0 为长度的二进制编码值。如果 Value 所含字节数大于 127，则 Length 的第一个字节的 bit7 位固定为 1，bit6~bit0 表示后继长度字节的个数，后继的字节二进制编码值表示 Value 部分实际的长度。

4.内容

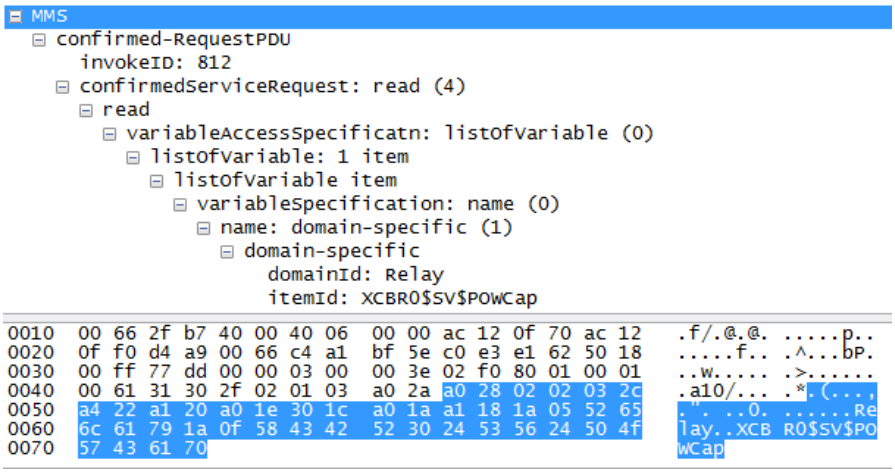
对于简单类型的数据，如位串 Bitstring，内容 Value 字段中是数据的实际值。

对于结构类型的数据，如 SEQUENCE OF，内容 Value 字段中是一个或多个数据的 TLV，形成了分层的结构，从最外层开始，层层嵌套，最后嵌套至最简单的数据类型为止。如下图所示

内容 Value 部分的层次嵌套



MMS 编解码示例



如图所示，MMS PDU 部分报文如下

a0 28 02 02 03 2c a4 22 a1 20 a0 1e 30 1c a0 1a a1 18 1a 05 52 65 6c 61 79 1a 0f 58 43 42 52 60 24 53 56 24 50 4f 57 43 61 70

按照表一所述，PDU **第 1 个字节**为标记 Tag，值为 0xa0，二进制表示方式为 1010 0000 按照表 2 解析如下：

101 代表是上下文相关的结构类型，MMS 报文的第 1 字节，都是 MMSpdu 结构开始，所以第 1 个字节的前 3 位都是 101。

结构 MMSpdu（二：1）是 ISO 9506-2 对 MMS PDU 结构的定义，它

位于分层结构的最外层，可以看做是“根”类型。该结构中共有 14 个成员，代表 14 种服务的 PDU。“CHOICE”关键字表明，编解码时 只能在其内部成员中选择其中之一。

结构中第一部分为成员名，第二部分[]表示该成员的 id，第三部

分为　该成员的类型

0 0000　代表选择 MMSpdu 里”[ ]”内值为 0 的一项，confirmed-RequestPDU（带确认的请求报文）

所以第 1 个字节”a0” 表示该报文是一个带确认的请求报文。

confirmed-RequestPDU 内容部分的定义在第二部分可见，关键字 SEQUENCE 强调内部成员的顺序，因此编码时该 PDU 内部的成员要按顺序排列。

关键字”OPTIONAL”表明，第二个成员和第四个成员是可有可无的。

第 2 个字节值为 0x28，Tag 后面是长度，二进制为 0010 1000，第 1 位是 0，所以 0x28 就是 PDU 剩下的长度，十进制为 40；

第 3 个字节 0x02　二进制 0000 0010

由表三可知，代表选择通用类的简单结构 INTEGER，即整型

第 4 个字节 0x02 代表 INTEGER 的长度占两个字节

第 5、6 个字节 为 INTERGER 的值，0x032c 十进制为 812。

Confirmed-RequestPDU 第二个和第四个成员带有 OPTIONAL 关键字，忽略。

第三个成员 Service 属于通用结构体类型，它的类型为 ConfiredServiceRequest，其内容部分的定义见第二部分，详细定义见保存的网页 MMS Syntax Page。该结构一共包含 86 个成员，代表

86 中带确认的请求报文。关键字 CHOICE 表明它是选择型结构。本报文所用到的结构都在第二章列出，完整的结构在保存的网页 MMS Syntax Page 中定义，不再赘述）

第 7 个字节 0xa4 二进制 1010 0100 101 表示选择的是上下文相关的结构类型，0100 十进制为 4，表示选择 ConfiredServiceRequest 结构中”[ ]”内为 4 的成员。即”read”，所对应的结构是 Read-Request。

第 8 个字节 0x22 表示”Read-Request”结构所占的字节，十进制为 34.

Read-Request 包含两个成员，第一个成员”specificationWithResult”属于布尔类型，上下文相关，由表四可知，它的 Tag 值应为”83”；它在编解码时经常被省略，默认值为 FALSE（由关键字 DEFAULT FALSE 可知）。

第 9 个字节为 0xa1，而非”83”，可知此处成员”specificationWithResult”被省略，采用默认值”FALSE”。0xa1 对应的二进制为 1010 0001 ，101 代表 Read-Request 的第二个成员 variableAccessSpecification 是一个上下文相关的结构类型，00001 代表 variableAccessSpecification 所对应的”[]”内的值为 1。类型 VariableAccessSpecification 声明中的”CHOICE”关键字表明是一个选择结构。

第 10 个字节 0x20 代表 Read-Request 的第二个成员 variableAccessSpecification 的字节数。十进制值为 32;

第 11 个字节 0xa0 二进制 1010 0000， 101 代表选择的是一个上下文相关的结构类型， 00000 说明选择 VariableAccessSpecification 中 ”[]” 内为 1 的成员 listOfVariable。

listOfVariable 的声明中，”SEQUENCE OF SEQUENCE”表明其是嵌套的结构，listOFVariable 本身为”SEQUENCE”类型，其内部成员 variableSpecification 也是”SEQUENCE”类型。第二个成员 alternateAccess 的 OPTIONAL 表明它是可选项。

第 12 个字节 0x1e 表示 listOfVariable 的占用的字节数，十进制为 30。

第 13 个字节 0x30 二进制 0011 0000 ， 001 代表 variableSpecification 是通用的结构类型， 1 0000 表明 variableSpecification 是 SEQUENCE 类型。

第 14 个字节 0x1c 表明 variableSpecification 占用的字节数，十进制为 28。

第 15 个字节 0xa0 二进制 1010 0000 表明选择的是一个上下文相关结构类型，选择”[ ]”里值为 0 的一项，即”name”，其类型为 ObjectName，”CHOICE”表明它是一个选择结构。

第 16 个字节 0x1a 代表”name”所占的字节数，二进制为 26。

第 17 个字节 0xa1 二进制 1010 0001 代表选择的是一个上下文相关结构，选择”[ ]”里值为 1 的一项，即”domain-specific”。声

明”IMPLICIT SEQUENCE”表明它是一个有序的数据集合。

第 18 个字节 0x18 代表”domain-specific”所占的字节数，十进制为 24;

第 19 个字节 0x1a 二进制 0001 1010 000 代表通用简单类型，1 1010 代表 VISIBLESTRING 类型，即”DomainID”的类型。

第 20 个字节 0x05 代表”DomainID”所占的字节数。

第 21~25 个字节 0x52 0x65 0x6c 0x61 0x79，按照 ASCII 解析为 Relay

第 26 个字节 0x1a 与第 19 个字节一样，表示 itemID 是一个通用简单类型的 VISIBLESTRING。

第 27 个字节 0x0f 代表”itemID”所占的字节数。十进制为 15。

第 28~42 个字节 0x58 0x43 0x42 0x52 0x60 0x24 0x53 0x56 0x24 0x50 0x4f 0x57 0x43 0x61 0x70 按照 ASCII 解析为 XCBR0SVPOWCAP。