

CMIP(共通管理情報プロトコル)ボード用

XMP/XOMインタフェースの設計

1 F-5

杉山敬三

黒木哲也

堀内浩規

小花貞夫

鈴木健二

KDD研究所

1. はじめに

OSI 管理に準拠した網管理システムでは、通信プロトコルとして CMIP(共通管理情報プロトコル)^[1]を用い、マネージャとエージェント間で管理情報を交換する。筆者等は、これまでに CMIP 以下の OSI 7 層のプロトコル処理をサポートする既存の CMIP ボード^{[2][3]}を利用し、そのアプリケーションとして GDMO(Guidelines for the Definition of Managed Objects)コンパイラ^[4]等の開発を行ってきた。

この CMIP ボードは、インタフェースとして独自のプリミティブを用いている。一方 X/Open では、CMIP 及びインターネット管理用プロトコルである SNMP(Simple Network Management Protocol)の両方に適用可能である標準的な API (応用プログラムインタフェース)として XMP (X/Open Management Protocols)^[5]/XOM(X/Open OSI-Abstract-Data Manipulation)^[6]を規定しており、それを使用したアプリケーションが普及しつつある。そこで CMIP ボードに XMP/XOM インタフェースを持たせるための設計を行ったので、その概要について報告する。

2. XMP/XOM及びCMIPボードのAPIの概要

2.1 XMP/XOM の概要

X/Open では、m-GETreq 等の各 CMIS(共通管理情報サービス)や GetNextReq 等の SNMP のサービスの両方に対応可能な API として、C 言語の関数に基づいた XMP^[5]を規定している。表 1 に、XMP API の関数一覧を示す。XMP の関数には、Action-req のように、応答を受信するまで処理をブロックする同期型及びブロックしない非同期型の両方のモードを持つものが存在する。

表 1 XMP API の関数

対応するプロトコルが存在する関数	Abort-req, Action-req*/rsp, Assoc-req*/rsp, Cancel-Get-req*/rsp, Create-req*/rsp, Delete-req*/rsp, Event-Report-req*/rsp, (Get-Next-req*) Get-req*/rsp, Release-req*/rsp, Set-req*/rsp,
インタフェースのみに関連する関数	Abandon, Bind, Error-Message, Get-Assoc-Info, Get-Last-Error, Initialize, Negotiate, Receive, Shutdown, Unbind, Validate-Object, Wait

注：*は非同期型でも呼び出し可能な関数であり、それ以外は同期型のみで呼び出し可能である。()は SNMP のみに適用される。

XOM^[6]は、種々の応用固有な API と組み合わせて使用され、それらのパラメータを操作する汎用的な API である。そのデータ構造は OM オブジェクトと呼ばれ、表 2 に示すように OM アトリビュートのリストからなり、個々の OM アトリビュートはさらにタイプ・シンタックス・値から構成される。

OM オブジェクトには、ワークスペースと呼ばれる領域に保持され、アプリケーションから直接操作することができないプライベート OM オブジェクト (PRI)と、アプリケーションから直接アクセス可能なデータ構造から構成されるパブリック OM オブジェクトが存在する。パブリック OM オブジェクトは、その記憶領域をアプリケーションとインタフェースのどちらが割り当てるかで各々 SPUB と CPUB と呼ばれる。XOM では、OM オブジェクトを操作するために om_create や om_get 等の関数を規定している。

表 2 CMIS-Event-Report-Argument OM オブジェクトの構造

タイプ	シンタックス	値の個数
managed-Object-Class	Object(Object-Class)	1
managed-Object-Instance	Object(Object-Instance)	1
event-Time	String(Generalized-Time)	0 or 1
event-Type	Object(Event-type-Id)	1
event-Info	any	0 or 1

注：Object(x)はさらに別の OM オブジェクト x を指すことを示す

2.2 CMIP ボードの API の概要

CMIP ボード^[2]では、C 言語の構造体をネストさせた構造化されたプリミティブを用いており、これがプログラムインタフェースとして提供されている。M-EVENT-REPORTreq プリミティブのデータ型の一部を図 1 に示す。ここでは、種々のユーザプログラムに対応するため、属性値や m-EVENT-REPORTreq の EventInfo 等 CMIP の ASN.1 定義で ANY と規定される部分は、ユーザに符号化・復号を行わせている。

```

typedef struct { prim_header_t      PrimHead;
                 u_long Invokeld;
                 u_long Mode; /* Confirmed or not*/
                 obj_class_t      MOClass;
                 obj_instance_t    MOInstance;
                 cm_time_t         EventTime;
                 cmip_ident_t      EventType;
                 u_long             EventInfoLen;
                 pointer_t(PDU_t)  EventInfo; } m_event_report_req_t;
typedef struct { u_long             FormFlag;
                 union { object_id_t GlobalForm;
                          u_long LocalForm; } Id; cmip_ident_t;
typedef struct { u_long             FormFlag;
                 union { dist_name_t DN;
                          char_str_t NonSpecific;
                          dist_name_t LocalDN; } Id; obj_instance_t;
typedef struct { u_long Num; pointer_t(rdn_t) Rdn; dist_name_t;
typedef struct { u_long Num; pointer_t(ava_t) Ava; rdn_t;
typedef struct { object_id_t AttrbType; any_t AttrbVal; } ava_t;
typedef struct { u_short           ValFlag;
                 union { char_32_t Comp;
                          cm_char_str_t Buff;
                          } Val; } any_t;
    
```

図 1 M-EVENT-REPORT 要求プリミティブのデータ型

3. CMIPボード用XMP/XOMインタフェースの設計

3.1 基本方針

(1)CMIP ボードのプリミティブ自身を変更するのではなく、XMP/XOM と独自プリミティブとの対応付けを行うことで実現する。

Design of XMP/XOM Interface for CMIP Board

Keizo SUGIYAMA, Tetsuya KUROKI, Hiroki HORIUCHI, Sadao OBANA and Kenji SUZUKI

KDD R&D Laboratories

(2)表1に示した XMP の全関数を同期/非同期モードとも実現する。また、マネージャ/エージェントとも動作可能とし、SNMP もサポートする。

(3)CMIP の ASN.1 定義で ANY と規定される部分の符号化/復号を XMP 内部で行う Automatic ASN.1 BER Encoding/Decoding 機能はサポートしない。

(4)実装環境は UNIX(SunOS 4.1.3) とし、CMIP ボードには VME バス対応 OSI 7 層ボード^[3]を使用する。SNMP モジュールは独自に開発したものを使用する。

3.2 ソフトウェア構成

図2にソフトウェア構成を示す。網掛けの部分がある今回の実装対象を示している。

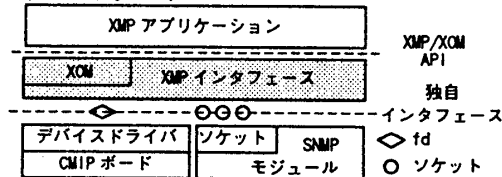


図2 ソフトウェア構成

XMP/XOM インタフェースが生成するパブリック OM オブジェクト (SPUB) の領域は本モジュール内に確保されるが、XMP アプリケーションから直接アクセス可能とする必要があるため、本モジュールは XMP アプリケーションのライブラリとし、同一プロセスとして実現することとした。また、CMIP ボードには UNIX の file descriptor (fd) によりアクセスし、SNMP モジュールは UNIX のソケットにより通信を行うため、本モジュールの起動時に fd およびソケットを取得するようにした。

3.3 管理操作及びパラメタの対応付け

(1)管理操作の対応付け

m-GETreq 等の CMIP や SNMP の管理操作については、表1の対応するプロトコルが存在する関数に示すように、XMP の関数と一対一に対応している。

(2)パラメタの対応付け

CMIP ボードのプリミティブ及び XMP は基本的に、ASN.1 の構造型毎に各々 C 言語の構造体と OM オブジェクトを、ASN.1 の単純型毎に各々構造体のメンバと OM アトリビュートを割り当てており、容易に対応付け可能である。

例えば、表2と図1に示すように、ASN.1 定義が ObjectClass である managed-Object-Class は、プリミティブと XMP で各々構造体 cmip_ident_t と OM オブジェクト Object-Class に対応しており、共に globalForm と localForm の要素を持つ。また、Automatic ASN.1 BER Encoding/Decoding を行わない場合、ASN.1 定義が ANY の要素は、XMP アプリケーションで符号化した値が XOM の EncodingString 型に入る。したがって、属性値の場合は、any_t 構造体のメンバ (Comp/Buf) と EncodingString を対応付ける。

3.4 同期・非同期モードの実現方法

(1)同期モードの実現

XMP の管理操作の関数が呼ばれたら、本モジュール内でインボーク ID を生成して CMIP ボードのプリミティブを発行し、操作応答プリミティブが到着す

るまで fd を監視し続ける。ここで、受信するプリミティブは連結応答であったり他のプリミティブの可能性があるので、プリミティブを一旦本モジュール内のキューに格納することとした。同一のインボーク ID やリンク ID を持つプリミティブを全て受信したら、応答に対応する OM オブジェクトを生成し、関数からリターンする。他のプリミティブは、後に非同期操作の結果を取得する関数である Receive が発行された時点でキューから取り除くようにした。

(2)非同期モードの実現

XMP の管理操作の関数が呼ばれたら、プリミティブを発行し、即座にリターンする。この時、本モジュールで生成したインボーク ID をユーザに通知する。

その後 Receive が呼ばれた時点で、キューまたは fd にプリミティブが到着しているかを調べる。プリミティブが到着していたら、その種類にかかわらず対応する OM オブジェクトを生成し、受信したインボーク ID とともにリターンする。データが到着していない場合、その旨をフラグ (Completion-Flag) にセットしてリターンする。

3.5 OM オブジェクトの管理

OM オブジェクトには、個々のオブジェクト毎にどのワークスペースに属するかを示すポインタや、PRI や SPUB といった OM オブジェクトの種類を示すフィールドが存在するため、ワークスペースや OM オブジェクトの種類毎に管理する必要はない。しかしながら、OM オブジェクトの検索の高速化の観点から、各ワークスペース毎に PRI と SPUB を別に管理するようにし、個々の OM オブジェクトはポインタでリンクするようにした。

4. おわりに

本稿では、CMIP ボードに XMP/XOM インタフェースを持たせるためのソフトウェア構成、管理操作及びパラメタの対応付け、同期・非同期モードの実現方法、OM オブジェクトの管理について述べた。CMIP ボードのプリミティブと XMP のパラメタで使用する OM オブジェクトは共に ASN.1 定義に基づいたデータ構造であるため、容易に対応付けできることがわかった。今後は Automatic ASN.1 BER Encoding/Decoding に対応するため、GDMO 定義から XMP/XOM の関数やデータ構造を生成させる拡張を行う予定である。最後に日頃ご指導頂く KDD 研究所浦野所長に感謝します。

参考文献

- [1]:ITU-T Rec. X.711, "Common Management Information Protocol Specification for CCITT Applications", 1991
- [2]:加藤他「パソコン用CMIPボードの開発」, 1992信学秋大
- [3]:井戸上他「パーソナルコンピュータおよびワークステーションのためのOSI 7層ボードの実装と評価」情処論文誌 Vol. 36 No. 3
- [4]:堀内他「OSI管理のシステム管理機能(SMF)のためのプログラム開発支援ツールの実装」, 1993信学春大
- [5]:X/Open CAE Spec. C306, "Systems Management:Management Protocols API(XMP)", Mar. 1994
- [6]:X/Open CAE Spec. C315, "OSI-Abstract-Data Manipulation API(XOM), Issue 2", Feb. 1994