

CMIPボード用XMP/XOMインタフェースにおける 符号化/復号機能の拡充

2V-5

杉山敬三

吉原貴仁

堀内浩規

小花貞夫

KDD研究所

1. はじめに

OSI 管理における通信プロトコルである CMIP(共通管理情報プロトコル)^[1]の標準的な API (応用プログラムインタフェース)として, X/Open では XMP (X/Open Management Protocols)^[2]/XOM(X/Open OSI-Abstract-Data Manipulation)^[3]を規定している. 筆者等は先に, CMIP 以下 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^[4]に XMP/XOM インタフェースを持たせるための CMIP ボード用 XMP/XOM インタフェース変換ソフトを開発した^[5].

このインタフェース変換ソフトでは, GDMO (Guidelines for the Definition of Managed Objects)^[6]で定義される管理情報の符号化/復号はアプリケーション側で行うこととしている. 今回, 本ソフトに XMP/XOM の仕様でオプションとして規定される管理情報の符号化/復号機能を拡充したので, その概要について報告する.

2. CMIPボード用XMP/XOMインタフェース変換ソフトとXMP/XOMにおける符号化/復号機能の概要

2.1 CMIP ボード用 XMP/XOM インタフェース変換ソフトの概要

XMP は, OSI 管理の CMIP とインターネットの SNMP (Simple Network Management Protocol)の両方に対応しており, C言語の関数に基づいた API を規定している. また, XOM は XMP 等応用固有な 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. XOM で扱うデータの単位は OM オブジェクトと呼ばれ, タイプ・シンタックス・値からなる OM 属性のリストで構成される. 扱う OM オブジェクトのクラス(以下 OM クラスと呼ぶ)の管理や OM オブジェクトの生成・削除は, OM ワークスペースと呼ばれる領域で行われる.

一方 CMIP ボードでは, CMIS(共通管理情報サービス)に対応して C言語の構造体をネストさせ構造化したプリミティブを API として提供しており, 属性値や通知情報のように GDMO で定義される管理情報はアプリケーションに符号化/復号を行わせている.

CMIP ボード用 XMP/XOM インタフェース変換ソフトの位置付けを図 1 に示す. 本ソフトは, XMP アプリケーションと CMIP ボードのデバイスドライバあるいは SNMP モジュールの間に位置し, 管理操作/通知レベル及びそれらのパラメータレベルでの対応付けを行う. 管理操作/通知レベルでは, CMIP ボードのプリミティブと XMP の関数は一対一に対応付けられる. また, パラメータレベルでは, CMIP の ASN.1 定

義が構造型の場合には CMIP ボードのプリミティブを表わす C言語の構造体と OM オブジェクト間での対応付けを行い, 単純形の場合には構造体のメンバと OM 属性間での対応付けを行う.

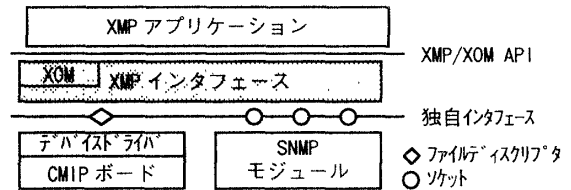


図1 CMIPボード用XMP/XOMインタフェース変換ソフトの位置付け

2.2 XMP/XOMにおける符号化/復号機能の概要

XMP/XOM では, GDMO で定義される管理情報の符号化/復号方法に関して, ①XMP/XOM API のアプリケーションが符号化/復号を行う, ②XMP/XOM API の内部で符号化/復号機能を提供する, の2種類の選択肢が存在する. これは, オプション機能を選択する XMP の negotiate API を用いて指定する.

①の場合, アプリケーションと XMP/XOM API は, ASN.1 基本符号化規則(BER)で符号化された管理情報を String(Encoding)というデータ型に格納して授受する. ②の場合, アプリケーションと XMP/XOM API は管理情報の GDMO 定義に対応する OM オブジェクトを授受し, XMP/XOM の内部において OM オブジェクトと CMIP のパラメータとの間で変換を行う. このため XOM では, OM オブジェクトから ASN.1 BER への符号化及びその逆の復号を行う om_encode と om_decode という関数を定義している.

GDMO 定義から対応する OM オブジェクトへの変換(以下 GDMO/XOM 変換と呼ぶ)規則は, X/Open で規定されている^[7]. この規則では, GDMO 定義を入力として, 表 1 に示す一連の情報(OM パッケージと呼ぶ)を出力する. アプリケーションが使用する OM パッケージは, negotiate API を用いて指定する.

表 1 GDMO/XOM 変換規則の出力情報

種類	詳細
OM パッケージ定義	OM パッケージ名とオブジェクト識別子(OID), GDMO 中の OID, OM クラスのリスト, OM 属性のリスト, OM 属性値のリスト
符号化定義	OM クラス/OM 属性に対応する ASN.1 型及びタグ
C 言語ヘッダ	OM パッケージの OID, GDMO 中の OID, OM クラス定数

3. CMIPボード用XMP/XOMインタフェースにおける符号化/復号機能の実現

以下では, GDMO 定義の変更に柔軟に対応可能とするためのソフトウェア構成と, 符号化/復号の際に必要な OM パッケージの管理機能並びに OM オブジェクトと ASN.1 BER 間での符号化/復号を行う om_encode/decode 関数の処理について述べる.

3.1 ソフトウェア構成

図 2 にソフトウェア構成を示す. 図 2 の OM パッ

Enhancement of Encoding/Decoding Function in XMP/XOM Interface Software for CMIP Board

Keizo SUGIYAMA, Kiyohito YOSHIHARA, Hiroki HORIUCHI and Sadao OBANA

KDD R&D Laboratories

パッケージ管理及び om_encode/decode 関数が新規に開発したモジュールである。OM パッケージ管理では、事前に GDMO 定義ファイルから GDMO/XOM 変換規則に従って変換された OM パッケージファイルを取り込み、3.2 で述べるように使用可能な OM パッケージや含まれる OM オブジェクトの情報を管理する。GDMO/XOM 変換処理には、既存のモジュール^[8]を利用した。また、om_encode/decode 関数は、CMIP ボードのプリミティブの受信時や mp_set_req 等の XMP 関数の起動時に、3.3 で述べる管理情報の符号化/復号を行う。

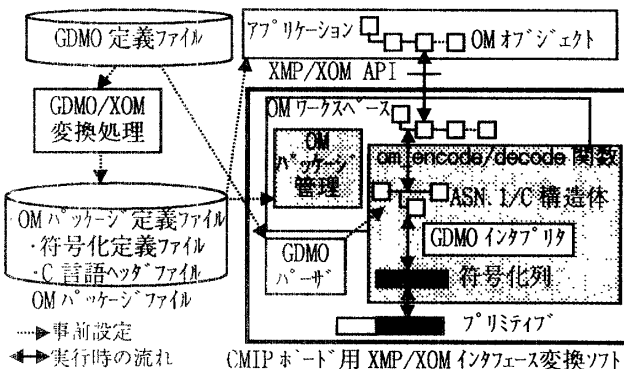


図2 ソフトウェア構成

3.2 OM パッケージ管理機能

符号化/復号時には、OM パッケージに含まれる各 OM クラスと ASN.1 定義とを対応付ける必要がある。そこで、図3に示すように OM パッケージファイルを解析し、対応関係を抽出するようにした。以下では、本解析処理で重要となる図3の OM クラス定義作成処理について示す。括弧内は値の例を示す。

(1) OM クラス情報抽出

C 言語ヘッダファイルから、該当する OM クラス名 (ISO_AE_TITLE_LIST) に対応する ASN.1 型 (AE-titleList) を抽出する。

(2) OM 属性情報抽出

OM パッケージ定義ファイルから、上記(1)の OM クラスにおける OM 属性名 (ISO_AE_TITLE) と OM 属性のシンタックス(Integer) を抽出する。シンタックスが他の OM オブジェクトを含む (Object (ISO_AE_TITLE)), すなわち OM サブオブジェクトの

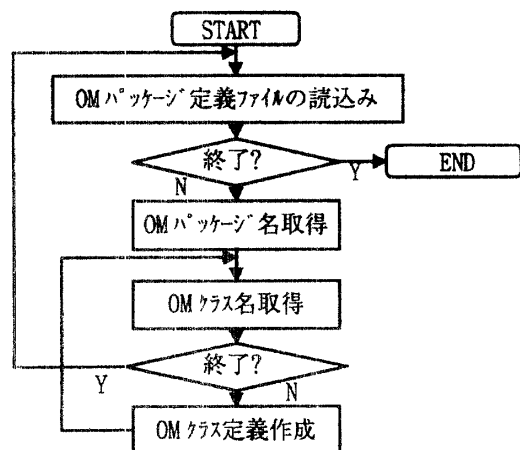


図3 OM パッケージファイル解析処理の流れ

場合、その ASN.1 型名 (AE-title) も抽出する。

(3) OM ワークスペースへの設定

各 OM ワークスペースの情報を管理する OM ワークスペーステーブルに、negotiate API により OM ワークスペースに登録される OM パッケージの情報を追加した。これは OM パッケージテーブルのリストとして実現した。各 OM パッケージテーブルには、OM オブジェクトへのポインタと、上記(1)(2)で抽出した OM オブジェクトの情報、具体的には OM クラス名、OM 属性名、OM クラスに対応する ASN.1 型名、OM サブオブジェクトが存在する場合にはその名前と対応する ASN.1 型名を、OM オブジェクトの個数分保持させた。

3.3 om_encode/decode 関数

om_encode/decode 関数は、図2に示したように既存の GDMO パーサ^[9]により GDMO 定義ファイルを解析して生成された C 言語構造体 (以下 ASN.1/C 構造体と呼ぶ) を保持する。ASN.1/C 構造体では、OM オブジェクトと同様、ASN.1 構造形は C 言語の構造体に、また単純形は構造体のメンバに割り当てる。ASN.1/C 構造体と ASN.1 BER 符号化列の変換には、既存の GDMO インタプリタ^[9]を利用しており、プログラムを変更せずに任意の GDMO 定義に対応可能である。

om_encode 起動時には、OM オブジェクトから ASN.1/C 構造体に変換し、GDMO インタプリタを呼び出すことで ASN.1 BER の符号化列を生成する。一方、om_decode 起動時には、GDMO インタプリタにより生成された ASN.1/C 構造体から OM オブジェクトを生成する必要がある。ここでは、まず復号された ASN.1/C 構造体が保持する ASN.1 型名を抽出して、OM ワークスペーステーブルから対応する OM クラス名と OM 属性名を検索する。次に、OM オブジェクトを生成し、個々の OM 属性に ASN.1/C 構造体中の値を設定する。

4. おわりに

CMIP ボード用 XMP/XOM インタフェース変換ソフトへの符号化/復号機能の拡充では、OM パッケージファイルや既存の GDMO パーサ/インタプリタを利用し、重要な機能である OM パッケージ管理機能と om_encode/decode 関数を効率的に実現した。最後に日頃ご指導頂く KDD 研究所村上所長に感謝します。

参考文献

- [1]:ITU-T Rec. X.711, "Common Management Information Protocol Specification for CCITT Applications", 1991
- [2]:X/Open CAE Spec. C306, "Systems Management: Management Protocols API (XMP)", Mar. 1994
- [3]:X/Open CAE Spec. C315, "OSI-Abstract-Data Manipulation API (XOM), Issue 2", Feb. 1994
- [4]:加藤他「パソコン用CMIPボードの開発」, 1992信学秋大
- [5]:杉山他「CMIPボード用XMP/XOMインタフェースの実装」第51回情報処全大, 1F-5
- [6]:ITU-T Rec. X.722, "Guidelines for the Definition of Managed Objects", 1992
- [7]:X/Open CAE Spec. C502, "Systems Management: GDMO to XOM Translation Algorithm", Oct. 1995
- [8]:HP OpenView Distributed Management Developer's Guide, "ovpackgen", 1994
- [9]:K. Sugiyama et.al. "Implementation and Evaluation of MIB Tester for OSI Management", IM97 1997