

SNMPによるTMN装置の監視／制御のための SNMP/OSI管理ゲートウェイの実装

堀内浩規 黒木哲也 杉山敬三 小花貞夫

国際電信電話株式会社 研究所

〒356 埼玉県上福岡市大原 2-1-15

本稿では、既存のTMN装置(OSI管理エージェント)を、なんら変更を加えず、SNMPのマネージャから監視／制御を可能とするSNMP/OSI管理ゲートウェイの実装について述べる。本ゲートウェイの実現では、管理操作と管理情報の効率的な変換手法が重要な課題となる。この課題に対して、NMF(Network Management Forum)のIIMC(ISO/CCITT and Internet Management Coexistence)による管理情報定義の対応付けをベースに、管理操作／管理情報変換の基本変換規則を新たに規定するとともに、キャッシュ、レプリカ、リンクオブジェクト等を用いた拡張変換規則を規定した。また、本ゲートウェイでは、GDMO定義に依存しないプログラム構造とすることと、GDMO定義から管理操作／管理情報変換に必要なテーブル等を自動生成することにより、種々のTMNエージェントに対応可能とした。性能等の評価の結果、本ゲートウェイの実用性を確認した。

Implementation of SNMP/OSI Management Gateway for Management of TMN-based Equipments by SNMP Manager

Hiroki Horiuchi, Tetsuya Kuroki, Keizo Sugiyama and Sadao Obana

KDD R&D Laboratories

2-1-15 Ohara Kamifukuoka-shi, Saitama 356, Japan

As the standardization of Telecommunication Management Network (TMN) progresses, TMN-based management for network elements such as transmission systems are being widely accepted. On the other hand, in management of Internet resources such as routers, SNMP has been widely used. We have developed SNMP/OSI Management Gateway which convert management operation and information between SNMP and OSI Management, technology of TMN, in order for SNMP managers to control and monitor TMN-based equipments (OSI Management Agents). For the gateway, efficient mapping mechanism on management operation and information have been developed, based on IIMC (ISO/CCITT and Internet Management Coexistence) by NMF(Network Management Forum). This paper describes implementation and evaluation of the gateway.

1. はじめに

TMN(電気通信管理網)¹⁾の標準化の進捗に伴い、これに基づいた通信機器等の管理が普及しはじめている。また、LAN機器等の管理は、SNMP(Simple Network Management Protocol)²⁾が業界標準として定着しつつある。このような背景のもと、これまでに、統合的な管理を行うために、TMNのベースとなるOSI管理のマネージャにより、SNMPのエージェントを管理可能とする研究が報告されている³⁾。

一方、SNMPのマネージャからOSI管理エージェントの機能を持つTMN装置の監視/制御を行う方法については、NMF(Network Management Forum)のIIMC(ISO/CCITT and Internet Management Coexistence)で、検討が行われている⁴⁾。ここでは、既存のTMN装置にSNMPエージェントのインタフェースも同時に持たせる方式をとっており、OSI管理のMIB(管理情報ベース)からSNMPのMIBへの管理情報定義の対応付け規則を規定している。しかしながら、この方式ではTMN装置にSNMPエージェントのインタフェースの機能追加が困難な場合には適用できないことや、適用できる場合であっても、そのインタフェースを提供するための負荷が増大し、本来提供すべきサービスに悪影響を与える可能性があるという問題点がある。そこで、筆者等は、エージェントとは別のワークステーション上で、SNMPとOSI管理との間で、管理操作/管理情報変換を行うSNMP/OSI管理ゲートウェイ⁵⁾を実装した⁶⁾。本ゲートウェイの実現では、管理操作と管理情報の効率的な変換手法が重要な課題となる。本稿では、効率的な管理操作/管理情報変換方式の提案と、それに基づいて変換を行なうゲートウェイの実装と評価について述べる。

2. IIMCにおける管理情報定義の対応付け規則

IIMCでは、図1に示すように、OSI管理インタフェースを持つ既存のTMN装置に、SNMPエージェントのインタフェースを同時に持たせるために必要となる、MIB(管理情報ベース)における以下の管理情報定義の対応付け規則を規定している⁴⁾。

- (1) OSI管理の管理情報定義(GDMO)の各管理オブジェクト(MO)クラスは、SNMPのテーブル(クラステーブル)およびエントリ(オブジェクトの集合)に対応付ける。
- (2) 属性型は、基本的に、MOクラスに対応するSNMPのエントリ配下のオブジェクトやポインタ(オブジェクト間の関係を示す)に対応付ける。この際、属性型のシンタックスが、単純形とSEQUENCE OF型等の構造形の場合で、以下のように対応付けが異なる。

●単純形の場合：単一のSNMPのオブジェクトに対応付ける。

●構造形の場合：SNMPのオブジェクトでは構造形を使用できないため、ポインタ、ポインタの値が指し示すサイドテーブル(構造形のメンバ集合)およびエントリに対応づける。また、各エントリを識別するためのインデックス(識別番号)を割り当てる。単純形に到達するまで、以上の対応付けを繰り返す。

- (3) MOクラス内にM-ACTIONが定義される場合は、SNMPのオブジェクトとして、新たにActionInfo/ActionTrigger等に対応付け、MOインスタンスの生成/削除が定義される場合は、RowStatusに対応付ける。

上記の対応付け規則の適用例として、ITU-T勧告X.721の属性型CorrelatedNotificationsのシンタックスを図2(a)に、SNMPへの対応付けを図2(b)に示す。ここでは、2段のサイドテーブル、INTEGER型等を示すオブジェクト、これらの関連付けを行うポインタ等から構成される。

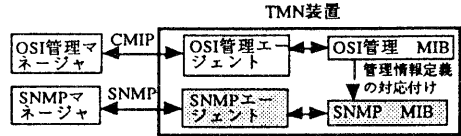


図1 IIMC方式の概念図

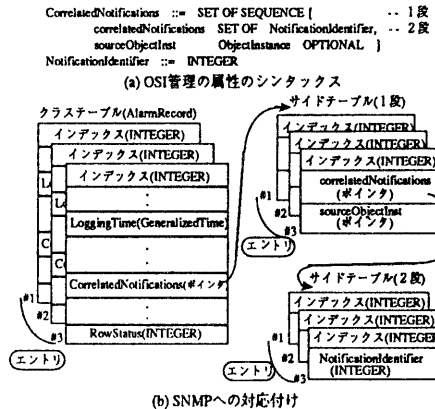


図2 OSI管理からSNMPへの管理情報定義の対応付け例

3. SNMP/OSI管理ゲートウェイの位置付け

SNMP/OSI管理ゲートウェイは、上記2章のIIMCの方式と異なり、TMN装置(OSI管理エージェント)とは別のワークステーション上で、SNMPからOSI管理への管理操作/管理情報変換を行う(図3)。言い替えば、SNMPマネージャに対しては、SNMPエージェントとして、また、OSI管理エージェントに対しては、OSI管理マネージャとして振る舞う。

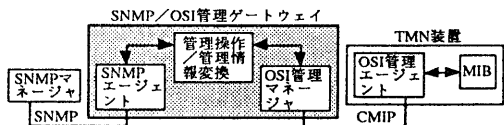


図3 SNMP/OSI管理ゲートウェイの概念図

4. SNMP/OSI管理ゲートウェイのための効率的な管理操作/管理情報変換方式の提案

以下、ここでは、特に、ことわりのない限り、マネージャはSNMPマネージャ、エージェントはOSI管理エージェント、ゲートウェイはSNMP/OSI管理ゲートウェイを、それぞれ示す。

4.1 基本方針

本ゲートウェイにおいて、管理操作/管理情報変換を開発する上で、以下の基本方針をたてた。

(1) 管理操作/管理情報変換

管理情報定義の対応付け規則は、IIMCをベースとすることにより、SNMPのマネージャにとっては、IIMCと本ゲートウェイでは、同様のSNMPオブジェクトとして扱えることとする。具体的な管理操作や管理情報を変換する方法は、IIMCでは規定しておらず、新たに定義する。

(2) 変換の効率化

機能の高いOSI管理エージェントを、機能の低いSNMPで対応付けるため、マネージャ、ゲートウェイ、エージェント間の管理操作回数が増えるので、これらの回数を削減し、管理操作/管理情報変換の効率化を図る。

以下では、上記の(1)に対応した基本変換規則を4.2節で、(2)に対応した拡張変換規則を4.3節で述べる。

4.2 基本変換規則

[前提条件] ゲートウェイは、管理操作/管理情報変換の前提として、①エージェントのMOクラス等のGDMO定義、②GDMO定義に対応付けられたSNMPオブジェクト定義、③エージェントが実装しているMOインスタンスの包含木と識別名からなるインスタンス情報を知識として持つこととする。また、ゲートウェイは、④インスタンス情報を利用して、各MOインスタンスに対応するSNMPのエントリ中のインデックスの値を割り当てることとする。

これらの前提条件のもとで、以下に基本変換規則を定義する。SNMPとOSI管理(CMIP)の管理操作の対応付けを表1に示す。

表1 管理操作の対応付け

操作種別	SNMP	OSI管理 (CMIP)
属性値の取得	GetRequest (GetNextRequest) / GetResponse	M-GETreq / cnf
属性値の設定	SetRequest / GetResponse	(M-GETreq / cnf) / M-SETreq / cnf
オブジェクト生成	SetRequest / GetResponse	M-CREATEreq / cnf
オブジェクト削除	SetRequest / GetResponse	M-DELETEreq / cnf
アクション実行	SetRequest / GetResponse, (GetRequest / GetResponse)	M-ACTIONreq / cnf
通知受信	Trap	M-EVENT-REPORTInd (M-EVENT-REPORTrsp)

(1) OSI管理の属性値の取得

SNMPのGetRequest操作のVarBindlistパラメータに含まれるオブジェクトが属性に対応付けられる場合、属性型とインデックスを示すオブジェクト識別子から、取得対象となるエージェントのMOクラス、属性ID、MOインスタンス名等の管理情報を抽出し、MOインスタンス毎に仕分けし、それぞれのMOインスタンスに対してM-GETreq操作を発行する。エージェントからのM-GETcnf受信後、属性型が単純形であれば、取得した属性値をVarBindlistパラメータに入れてGetResponse操作をマネージャに発行する。属性型が構造型である場合には、属性値をSNMPのオブジェクトやポインタに分解して、必要なオブジェクトを抽出する処理が追加される。また、取得対象の属性値がMOインスタンスの識別名に対応する場合には、ゲートウェイ内で値を保持しているので、エージェントへのアクセスは不要である。

(2) 属性値の設定

SetRequest操作のVarBindlistパラメータに含まれるオブジェクトがMOインスタンスの属性に対応付けられるオブジェクトの場合、設定対象となるMOインスタンスのMOクラス、MOインスタンス名、属性IDの管理情報を抽出する。この際、属性型が構造型で一部のメンバの値のみが変更される場合には、OSI管理ではメンバのみの変更を実行できないので、最初、①M-GETreq/cnf操作によりその属性値を取得する。その後、②設定するメンバの値を属性値に埋め込んだ後、その結果を持つM-SETreqをインスタンス毎に発行する。最後に、③エージェントからのM-SETcnf操作における属性値変更の成否をGetResponseのVarBindlistパラメータに設定してマネージャに発行する。

(3) オブジェクトの生成および削除

SetRequest操作に含まれるオブジェクトがRowStatusで、設定する値がcreateAndGoであれば、エージェントの対応するMOインスタンスに対してM-CREATEreqを発行し、値がdestroyであればM-DELETEreqを発行する。M-CREATEcnfおよびM-DELETEcnf受信後、生成/削除の成否をGetResponseのVarBindlistパラメータに設定してマネージャに発行する。

(4) アクションの実行

SetRequest操作に含まれるオブジェクトが同時にActionInfoとActionTriggerを指定した場合、M-ACTIONreqを発行する。M-ACTIONcnf受信後、GetResponse操作をマネージャに返す。この際、M-ACTIONcnf操作のActionResultや正常に実行されたか否かの情報(ActionStatus)は、SNMPのSetRequest操作の結果となるGetResponse操作には含めることができないため、SNMPのオブジェクトとしてゲートウェイに保持させ、マネージャが必要に応じてGetRequest操作で検索できるようにした。

(5) 通知

エージェントからのOSI管理のM-EVENT-REPORTIndの受信に対し、マネージャへTrapを発行する。この際、OSI管理のEventTypeとEventInfoパラメータを、SNMPのTrapのgenericTrapとVarBindListパラメータへ、それぞれ、変換する。この際、通知がオブジェクト生成/削除の場合には、エージェントのインスタンス情報を更新も併せて行なう。

4.3 変換の効率化

管理操作と管理情報の変換は、機能的には、3.2節で示した基本変換規則で十分であるが、以下に示すように、アクセス回数の増加や無効なアクセスが発生する場合があるため、効率化を図る必要がある。

[問題1] 構造型属性値を取得する際のアクセス回数増加

GDMO定義の構造型の属性値は、複数のオブジェクト、ポインタやサイドテーブル等に対応づけるため、属性値の取得の際には、図2に示すように、先頭から順にポインタをトレースする等が必要となり、マネージャからゲートウェイ、および、ゲートウェイからエージェントへの多数のアクセスが発生する。

[問題2] 構造型属性値を設定する際のアクセス回数増加

構造型の属性値の一部メンバに対応するSNMP側のオブジェクトの値を変更する場合には、エージェントにM-SETreqを発行するために、予め、他のメンバに対応するオブジェクトの値をマネージャまたはゲートウェイが取得する必要がある。このため、マネージャからのGetRequest操作や、ゲートウェイからエージェントへのM-GETreq操作の発行回数が増加する。

[問題3] エージェントにより内容の変更が発生しない属性値取得時のアクセス回数の増加

内容の変更が発生しない、または、値の変更が通知される属性値の場合には、マネージャからの取得要求毎にゲートウェイがエージェントへアクセスするのは冗長である。

[問題4] オptional指定等に対する無効アクセス

SNMPでは、属性型のシンタックスにOptional指定やCHOICE型を許していない。しかしながら、IIMCでは、これらの要素の有無を示すオブジェクトが定義されていないために、マネージャは実際にアクセスするまで存在の有無がわからず、無効なアクセスが発生する。また、SEQUENCE OF、SET OF型のメンバの繰り返し数やMOインスタンスの個数も定義されていないため、マネージャからの無効な操作が発生する。

以下にこれらの課題を解決する拡張変換規則を述べる。

4.3.1 拡張変換規則

以下のキャッシュ、レプリカ(複製)、リンクオブジェクトを新たに導入するとともに、管理情報定義の拡張を行なう。

(1) キャッシュ (問題1への対処)

GetRequest/Get-NextRequestに対応づけられたM-GETreqによって、得られた属性値が構造型の場合、値をキャッシュとしてゲートウェイに保持する。次からの検

素要求が、キャッシュした属性に対するものであれば、キャッシュの情報を返送し、エージェントへの冗長なアクセスを削減する(図4)。キャッシュは、マネージャ毎に一つの構造形の属性値を保持し、キャッシュ以外の属性にアクセスが発生した場合や、SetRequestが発生した場合には消去する。

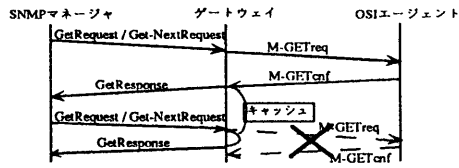


図4 キャッシュによる効率化

(2) レプリカ (問題1, 2, 3への対処)

ゲートウェイ上に、内容の変更が発生しない、または、変更の結果がゲートウェイに通知される属性値の複製(レプリカ)を持たせて、構造形の属性値の一部メンバの値変更時に、このメンバ以外の値をレプリカから補うことにより、GetRequestとM-GETreqの発行回数を削減する(図5)。レプリカは、キャッシュと異なり、エージェント毎に、複数の単純形と構造形の属性値を長期間保持する特徴を持つ。また、マネージャが属性値の検索時に、該当する属性のキャッシュが存在しない場合でも、レプリカを利用することで、エージェントへの冗長なM-GETreqを削減することができる。レプリカは、マネージャがGetRequest操作により属性値を取得した際に生成する。また、レプリカは、一定の個数を越えた場合等は、古いものから削除することとした。

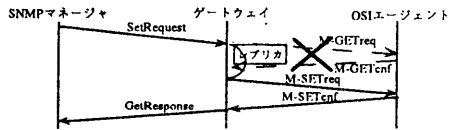


図5 レプリカによる効率化

(3) リンクオブジェクト (問題1への対処)

サイドテーブルで使用するインデックス値を、その階層構造に従って保持するリンクオブジェクトとして、ゲートウェイに新たに定義する。マネージャがリンクオブジェクトの情報を取得することで、サイドテーブルの接続関係がわかり、ポインタをトレースせずに、目的とするSNMPのオブジェクトに直接アクセス可能となる。リンクオブジェクトのASN.1定義を図6に示す。これは、マネージャからの1回のアクセスで取得可能なように、Opaque型で符号化してゲートウェイに格納され、マネージャが復号して利用する。

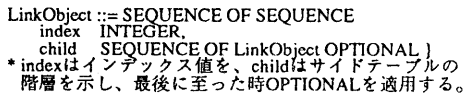


図6 リンクオブジェクトのASN.1定義

リンクオブジェクトは、GDMO定義のMOクラス毎に定義し、マネージャがリンクオブジェクトの値を取得する場合には、属性に対応するオブジェクトのオブジェクト識別子(OID)と、その属性を含むMOクラスに対応するリンクオブジェクトのOIDをペアで指定する。

上記(1)、(2)、(3)の手法を導入した場合の属性値の取得と属性値の設定における管理操作の処理の流れを、それぞ

れ図7および図8に示す。

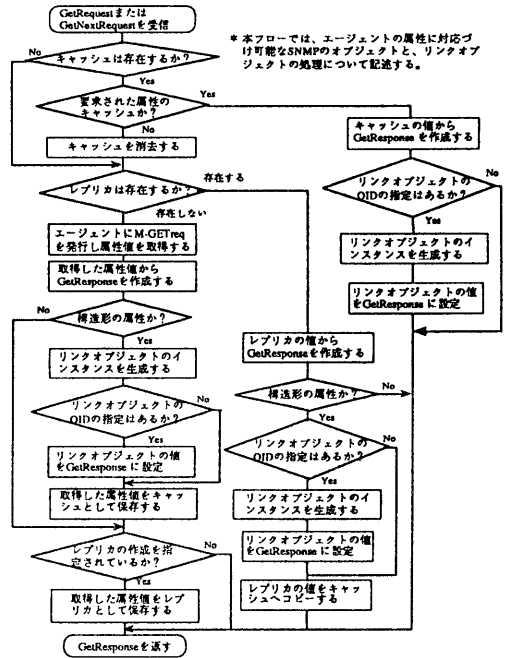


図7 属性値の取得における効率化の処理の流れ

(4) 管理情報定義の対応づけ規則の拡張(問題4への対処)

以下のオブジェクトをゲートウェイ上に新たに定義して、マネージャからの無効なアクセスを抑制する。

- ①属性型がSEQUENCE, SET型である場合、属性中のオプション指定されたメンバの有/無を示すビット列からなるOCTET STRING型のオブジェクト(Flag)。
- ②属性型がCHOICE型である場合、属性中の選択したメンバを示すINTEGER型のオブジェクト(Choice)。
- ③コンディショナルな属性の有無を示すビット列からなるOCTET STRING型のオブジェクト(Flag)。
- ④属性型がSEQUENCE OF, SET OF型の場合、メンバの繰り返し数を示すINTEGER型のオブジェクト(Number)。
- ⑤MOインスタンスの個数を表すINTEGER型のオブジェクト(Number)。

5. 実装

5.1 実装方針

- ゲートウェイ実装にあたり、以下の方針をたてた。
- (1) 4.2節および4.3節で提案した基本変換規則と拡張変換規則により、管理操作/管理情報変換を行う。
- (2) 種々のOSI管理エージェントに対応可能とするため、GDMO定義の変更に対する柔軟性を持たせる
- (3) 複数のマネージャ、および、複数のエージェントに同時に対応可能とする。
- (4) SNMPは現在普及しているバージョン1を対象とし、OSI管理のプロトコル処理には利用可能なCMIPボード[®]を使用する。
- (5) 実装対象の計算機はSUNとする。

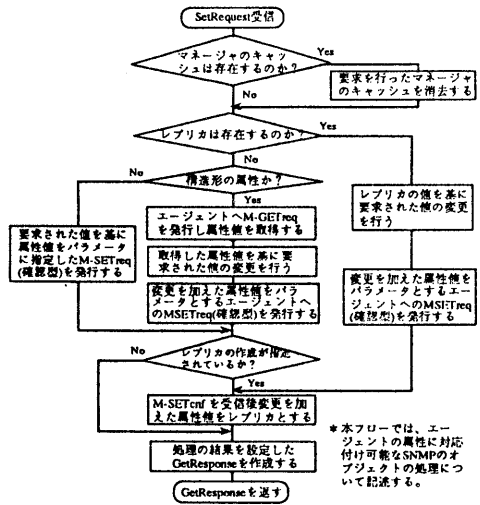


図8 属性値設定における効率化の処理の流れ

5.2 システム構成

図9に示すように、本ゲートウェイは、① SNMPのオブジェクトとOSI管理のMOクラスの対応付けを定義したOSI管理/SNMP管理情報定義対応テーブル(詳細は5.3節(1)参照)、② エージェントの包含木を保持するインスタンス情報、③ SNMPとOSI管理の管理操作と管理情報の変換を行う管理操作/管理情報変換部、④ OSI管理(CMIP)及びSNMPのPDUの作成や解析を行うCMIP及びSNMPプロトコル処理部、⑤ インスタンス情報を維持・管理するインスタンス管理部、⑥ GDMO定義からOSI管理/SNMP管理情報定義対応テーブルを自動生成するGDMO/SNMPオブジェクトトランスレータからなる。

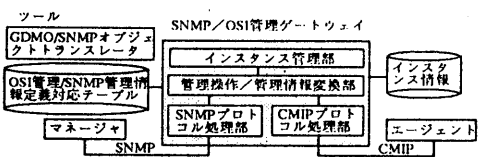


図9 SNMP/OSI管理ゲートウェイのシステム構成

5.3 GDMO定義の変更に対する柔軟性の実現

(1) GDMO定義に依存しないプログラム構成

GDMO定義の変更に対処可能とするため、MOクラス毎に異なる情報は、ゲートウェイのプログラムとして組み込まず、OSI管理/SNMP管理情報定義対応テーブルとして分離して管理することとした。

このテーブルには、① 対応付けられたSNMPオブジェクト定義、② SNMPオブジェクトの種別(属性型、ポインタ、生成/削除、Action等)、③ 関連するOSI管理のMOクラス及びそのオブジェクト識別子、④ 属性型のシンタックス等の項目を格納する。これらの定義項目は、4.2節の前提条件で述べたゲートウェイの知識の一部として提供され、管理操作/管理情報変換において、①はSNMPのGetRequestやSetRequestのVarBindListパラメータから対象のオブジェクト抽出時に、②はエージェントへ発行するOSI管理の操作を決定するため等に、③はOSI管理操作のMOクラスパラメータ設定のために、④はOSI管理の属性とSNMPのオブジェクト間のシンタックス変換時に、それぞれ使用する。さらに、本テーブルには、⑤ MOクラスの属性毎に各マネージャがアクセス可能とするか否かや、⑥ 4.3節で示す

レプリカを作成するか否かの定義項目を設け、ユーザが指定可能とした。

上記④に関連して、属性のシンタックスとして使用されるASN.1定義の符号化/復号関数は、X.721やM.3100等の代表的な動告で使用されるパターンに対応するものを予め登録する方式をとった。登録されているシンタックス以外の場合には、ASN.1の符号化/復号処理のためのライブラリを追加することとした。

(2) GDMO/SNMPオブジェクトトランスレータ

(1)で示したOSI管理/SNMP管理情報定義対応テーブルは、GDMO/SNMPオブジェクトトランスレータによりGDMO定義から自動生成する方式とした。また、マネージャが、ゲートウェイをアクセスするために必要なSNMPのオブジェクト定義も合わせて自動生成する。図10に本トランスレータの構成を示す。

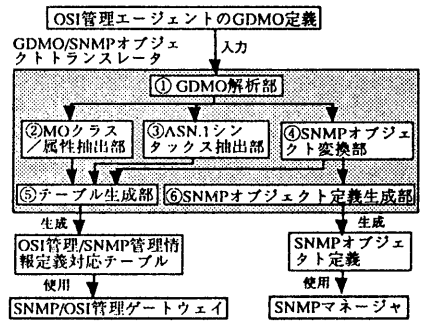


図10 GDMO/SNMPオブジェクトトランスレータの構成

図11に自動生成したOSI管理/SNMP管理情報定義対応テーブルの一部を示す。ここでは、ITU-T動告X.721でのMOクラスalarmRecordを交換しており、SNMPのエントリ(smi2AlarmRecordEntry)、オブジェクト(smi2AlarmRecLoggingTime等)、関連する属性型(logRecordId, loggingTime)、各属性型のオブジェクト識別子とシンタックス、レプリカとするか否か(Y/N)等を示している。

```

+ smi2Translated()
: TRANSLATED(smi2Translated)
: OBJECT
: : CLASS(alarmRecord)
: : : TABLE
: : : : NUMBER
: : : : : ENTRY
: : : : : : INDEX
: : : : : : : FLAG
: : : : : : : : logRecordId
: : : : : : : : loggingTime
: : : : : : : : : SIDE-TABLE(2)
: : : : : : : : : PARENT
: : : : : : : : : ROWSTATUS
: : : : : : : : : : TABLE
: : : : : : : : : : : NUMBER
: : : : : : : : : : : : ENTRY
: : : : : : : : : : : : : INDEX
: : : : : : : : : : : : : : FLAG
: : : : : : : : : : : : : : : SIDE-TABLE(3)
: : : : : : : : : : : : : : : :
logRecordId 2.9.3.2.7.3 SimpleNameType Y 00X00-0000
loggingTime 2.9.3.2.7.3.9 LoggingTime Y 00X00-0000
managementObjectClass 2.9.3.2.7.6.0 ObjectClass N 0000000000
managementObjectInstance 2.9.3.2.7.6.1 ObjectInstance Y 0000000000
eventType 2.9.3.2.7.14 EventType Y 0000000000
eventTime 2.9.3.2.7.13 EventTime Y 0000000000
notificationIdentifier 2.9.3.2.7.16 NotificationIdentifier Y 0000000000

```

図11 OSI管理/SNMP管理情報定義対応テーブル

6. 評価と考察

6.1 性能評価

実装したゲートウェイの性能を評価するため、ゲートウェイ内における処理時間(管理情報/管理操作変換、SNMPおよびOSI管理プロトコル処理等を含む)をプロトコルモニタを用いて測定した(表2)。なお、計算機は

SUN 670MPを使用し、マネージャ、ゲートウェイ、エージェント間はEthernetのLANで接続した。表2の①、②より、単純形の属性値取得ではMOインスタンスが1個増える毎に、M-GETreqへの変換やプロトコル処理のため、約55msの処理時間が増加する。表2の処理時間より、全ての管理操作において、実用的な性能を達成していると言える。

表2 ゲートウェイの処理時間

測定項目	処理時間
① 属性値(loggingTime)取得 (インスタンス1個)	113.9 ms
② 属性値(loggingTime)取得 (インスタンス3個)	224.3 ms
③ 属性値(administrativeState)の設定	109.2 ms
④ MOインスタンス(log)の生成(1個)	97.7 ms
⑤ MOインスタンス(log)の削除	93.0 ms
⑥ アクション(addTpsToGTP)の実行	146.2 ms
⑦ 通知(CommunicationAlarm)の受信	95.6 ms

注) 使用したインスタンスは4個のRDNから構成され、MOクラスは勧告X.721の"AlarmRecord", "Log", "Network", M.3100の"Fabric"を使用。

6.2 拡張変換規則の効果

キャッシュ、レプリカ、リンクオブジェクト、管理情報定義の拡張による管理操作のアクセス回数削減の効果を、ITU-T勧告X.721のMOクラスAlarmRecord含まれる構造形の属性型correlatedNotifications(サイドテーブル2段)、MOクラスLogに含まれる構造形の属性型weekMask(サイドテーブル3段)を使用して(インスタンス数1個)、測定した。

(1) キャッシュ

correlatedNotificationsのメンバNotificationIdentifierの値取得時には、3回のGetRequestに加え、キャッシュ不使用時は3回のM-GETreq/cnfが必要であるが、使用時には1回のM-GETreq/cnfで取得可能となり、ゲートウェイとエージェント間のアクセス回数が削減される。この際のゲートウェイ内の処理時間は、キャッシュ不使用時で342ms、使用時173msとなり、約1/2に削減される。

(2) レプリカ

weekMaskのメンバdaysOfWeekの値設定時には、レプリカ不使用では、マネージャからの4回のGetRequest、1回のSetRequest、ゲートウェイからの4回のM-GETreqと1回のM-SETreqの発行が必要となり、レプリカ使用時には、マネージャからの1回のSetRequestとゲートウェイからの1回のM-SETreq発行のみとなる。この際のゲートウェイ内の処理時間は、レプリカ不使用時で565ms、使用時は120msとなり、約1/5に削減される。また、表2の①に示す単純形の属性値の取得においても、レプリカを使用した場合のゲートウェイ内の処理時間は29.5msなり、約1/4に削減される。

(3) リンクオブジェクト

weekMaskのメンバminutesの値取得時には、リンクオブジェクト不使用では4回のGetRequestとM-GETreq/cnfが必要であるが、使用時には2回のGetRequestとM-GETreq/cnfで取得可能となる。この際のゲートウェイ内の処理時間は、リンクオブジェクト不使用時で455ms、使用時は228msとなり、約1/2に削減される。

(4) 管理情報定義の拡張

correlatedNotificationsのSEQUENCE OFの要素が3個ある場合、オブジェクトNumber不使用では、6回のGetRequestとM-GETreq/cnfが必要であるが、使用時には4回のGetRequestとM-GETreq/cnfで取得可能となる。この際のゲートウェイ内の処理時間は、Numbc不使用時で683ms、使用時は456msとなり、約3/4に削減される。

以上により、提案した拡張変換規則による変換の効率化の効果が大きいことが実証された。これらの効果は、構造形のネストの繰り返し回数やインスタンス数が多くなるに従ってさらに増大する。上記の評価では、ゲートウェイ内の処理時間の短縮しか述べていないが、エージェントの処理時間も短縮されるので、トータルの応答時間のさらなる向上が図れることは自明である。キャッシュ、レプリカ、リンクオブジェクトが有効となる構造形の属性型の定義は、比較的多く(例えば、勧告X.721で定義される属性型71個のうち、46個が構造形)、その効果が期待できる。

6.3 GDMO定義の変更に対する柔軟性

本ゲートウェイでは、MOクラスの変更や追加が行われた場合には、変更されたGDMO定義からトランスレータが生成するOSI管理/SNMP管理情報定義対応テーブルを入れ替えることにより、容易に対処できる。属性のシンタクスのASN.1が登録されているものと一致しない場合でも、その型が単純形であったり、また、構造形であっても、登録されているシンタクスとメンバの構造が変わらない場合には変更の必要はない。変更が必要な場合でも、ASN.1コンパイラの利用と、構造形のメンバ毎に10行程度のプログラムの追加することにより容易に対応可能となる。以上のことからGDMO定義の柔軟性は高いと言える。

7. おわりに

本稿では、SNMPマネージャからTMN装置の監視/制御を可能とするSNMP/OSI管理ゲートウェイの実装について報告した。本ゲートウェイの実現には、管理操作や管理情報の効率的な変換が重要となる。これに対し、NMFのIIMCの管理情報の対応付けをベースに、管理操作/管理情報変換の基本変換規則を新たに規定するとともに、効率化を図るため、キャッシュ、レプリカ、リンクオブジェクトを導入するとともに、管理情報定義の拡張を行ない、これらによる拡張変換規則を規定した。また、種々のエージェントへの容易な対応を可能とするため、GDMO定義に依存しないプログラム構造とすることとGDMO定義から管理操作/管理情報変換に必要なテーブル等を自動生成することとした。さらに、性能等の評価の結果、本ゲートウェイが実用できることを示した。

本ゲートウェイにより、既存のTMN装置を、なんら変更を加えず、SNMPを具備するパソコンやワークステーション等からSNMPの体系で、統一的に監視/制御可能となった。

最後に日頃ご指導頂くKDD研究所 浦野義頼 所長、鈴木健二 次長に感謝します。

参考文献

- [1]: ITU-T Rec. M.3010, "Principles for Telecommunications Management Network", 1992.
- [2]: J. Case, M. Fedor, M. Schoffstall, J. Davin, "A Simple Network Management Protocol (SNMP)", IETF, RFC 1157, May 1990.
- [3]: 堀内, 杉山, 小花, 鈴木 "TMNに基づく統合管理のためのOSI管理/SNMPゲートウェイの設計と実装", 信学技報 IN94-85, Sep. 1994.
- [4]: 宮内他, "分散LANドメインのOSIによる統合管理", 情報処理学会論文誌, vol.34 No.6, 1993.
- [5]: NM Forum "Forum 030: Translation of ISO/CCITT GDMO MIBs to Internet MIBs", Oct. 1993.
- [6]: 堀内, 黒木, 杉山, 小花, 鈴木 "SNMP/OSI管理ゲートウェイの設計", 情処第51回全大, 1F-03, Sept. 1994.
- [7]: 黒木, 堀内, 杉山, 小花, 鈴木 "SNMP/OSI管理ゲートウェイにおける効率的な管理操作変換の実現", 情処第51回全大, F-04, Sept. 1994.
- [8]: 加藤, 堀内, 井戸上, 鈴木 "パソコン用CMIPボードの開