マルチメディア通信と分散処理　85－40
（１９９７．１１．７）

分散オブジェクトを用いたネットワーク管理における TMN/SNMP収容方式の提案

堀内浩規　　吉原貴仁　　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

ネットワーク管理システムの構築への分散オブジェクト技術の適用が注目されており、今後の普及が予想される。ここでは、分散オブジェクト環境へTMN(電気通信管理網)やSNMP(Simple Network Management Protocol)に準拠した既存装置の収容が必須となる。本稿では、分散オブジェクトの業界標準であるCORBA(Common Object Request Broker Architecture)環境にTMN/SNMP装置を収容する方式を提案する。ここでは、X/OpenとNMF(Network Management Forum)によるJIDM(Joint Inter Domain Management)が規定した、TMNやSNMPの管理情報定義とCORBAのIDL(インタフェース定義言語)定義との対応付けの規定をベースとして、そのIDL定義を拡張し、また、新たにIDL操作と管理操作の対応付け等を規定した。提案方式に基づいてCORBA/SNMPゲートウェイを実装し、その評価結果を通じて提案方式の有効性を示した。

Accomodation Method of TMN/SNMP in Network Management System based on Distributed Object Technology

Hiroki Horiuchi, Kiyohito Yoshihara and Sadao Obana

KDD R&D Laboratories

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

This paper proposes accomodation method of TMN (Telecommunications Management Network)-based and SNMP (Simple Network Management Protocol)-based equipments in network management systems based on distributed object technology. Although JIDM (Joint Inter-Domain Management) group by X/Open and Network Management Forum specifies translation algorithms from TMN and SNMP MIB (Management Information Base) definition to CORBA (Common Object Request Broker Architecture) IDL (Interface Definition Language) definition, the IDL definition is not sufficient for the accomodation.

Therefore, the proposed method extends the IDL definition for efficient operations and newly defines the mapping rules among management operations and IDL operations, based on JIDM. Furthermore, we implement CORBA/SNMP gateway based on the proposed method and we show the effectiveness of the method through the implementaion and evaluation.

1．はじめに

ネットワーク管理システムの構築への分散オブジェクト技術の適用が注目されており、今後の普及が予想される。ここでは、分散オブジェクト環境へTMN(電気通信管理網)[1]やSNMP(Simple Network Management Protocol)[2]に準拠した既存装置の収容が必須となる。

このため、X/OpenとNMF(Network Management Forum)によるJIDM(Joint Inter Domain Management)は、TMNとSNMPの管理情報定義を分散オブジェクト環境の業界標準であるCORBA(Common Object Request Broker Architecture)[3]のIDL(インタフェース定義言語)定義への対応付けを規定している[4]。しかしながら、TMN/SNMP装置のCORBAへの収容には、この規定だけでは十分でない[5,6]。

本稿では、JIDMの規定をベースにそのIDL定義の拡張や新たなIDL操作と管理操作の対応付け等を規定する、CORBAへのTMN/SNMP装置の収容方式を提案する。また、提案方式に基づき、SNMP装置を収容するCORBA/SNMPゲートウェイの実装と評価結果を述べる。

2．CORBAとJIDM仕様の概要

2．1 CORBAの概要

CORBAのアプリケーションはクライアントとサーバから構成され、サーバはオブジェクト(以下、CORBAオブジェクトと呼ぶ)を持ち、クライアントはサーバ上のCORBAオブジェクトが提供するサービスを利用する。CORBAオブジェクトが提供するサービスのインタフェースは、IDLを用いて、CORBAオブジェクトの属性と操作(以下、IDL属性とIDL操作と呼ぶ)、データ型等が定義される。IDL属性は、クライアントからアクセス可能なCORBAオブジェクトのデータであり、また、IDL操作はクライアントがCORBAオブジェクトに依頼する処理である。図1(a)にIDL定義の例、図1(b)にインタフェースとCORBAオブジェクトの関係を示す。

クライアントがIDL属性やIDL操作を呼出す際には、対象となるCORBAオブジェクトをオブジェクトリファレンスにより指定する。オブジェクトリファレンスは、サーバでCORBAオブジェクトが生成される際に付与され、CORBAオブジェクトの名前とオブジェクトリファレンスの組を管理する名前管理サーバに登録する。クライアントは、名前管理サーバを通じて、名前からオブジェクトリファレンスを取得する。CORBAオブジェクトの名前はNaming Graph(NG)と呼ばれる木構造で管理され、節を示すNaming Context(NC)と葉を示すName(NA)からなる。

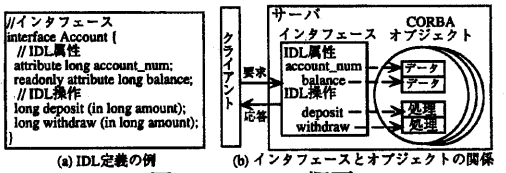


図1 CORBAの概要

2．2 JIDMのIDL定義への対応付けの概要[4]

TMNの管理情報定義(GDMO定義)のIDL定義への対応付けでは、①管理オブジェクト(MO)クラスをインタフェースに、②属性型、アクションおよび通知を、それぞれ、IDL操作に対応付ける。

SNMPの管理情報定義(SNMP-SMI)のIDL定義への対応付けでは、①グループおよびテーブルエントリをインタフェースに、②オブジェクトタイプをIDL属性に対応付ける。

3．CORBAへのTMN/SNMP収容方式の提案

3．1 収容形態

CORBA環境にTMN/SNMP装置を収容するには、TMNのMOクラスやSNMPのグループやテーブルエントリのインスタンスをCORBAオブジェクトに対応付けて、図2に示すように、CORBAのIDL操作とTMN/SNMPの管理操作や通知を相互に変換するゲートウェイが必要となる(図2)。また、ゲートウェイはCORBAオブジェクトの名前とオブジェクトリファレンスの組を名前管理サーバに登録する。

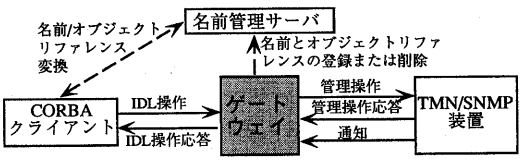


図2 TMN/SNMP装置のCORBA環境への収容形態

3．2 JIDMのIDL定義への対応付けの拡張

IDL属性の取得/設定における管理操作変換の効率化とCORBAオブジェクトの生成/削除を可能とするため、以下のようにJIDMの管理情報とIDL定義との対応付け規定(以下、JIDMの規定と呼ぶ)を拡張する。

3．2．1 IDL属性の取得/設定のための拡張

JIDMの規定では、TMNの一つの属性型を一つのIDL操作に対応付けるため、複数IDL属性の取得/設定時には、属性の個数分だけIDL操作とM-

GET等の管理操作が発行され効率的でない問題点がある。また、SNMPの場合にも、オブジェクトタイプをIDL属性に対応付けるため、複数IDL属性の取得/設定時におけるIDL操作とGetRequest等の管理操作との変換においても同様な問題点がある。これらの問題点を解決するために、取得/設定対象の複数IDL属性が同一CORBAオブジェクトにある場合と、複数CORBAオブジェクトにまたがる場合とに分けて、以下の拡張を行う。

（1）同一CORBAオブジェクトの場合

JIDMに従ったインタフェースに、複数のIDL属性の取得/設定を可能とする新たなIDL操作(get_attribute_list, set_attribute_list)を追加定義する(図3(a))。

（2）複数CORBAオブジェクトの場合

TMNの複数のMOインスタンスにまたがるM-GET等の管理操作に対応付け可能なインタフェース(mo_scope_filter)を新たに定義する(図3(b))。SNMPの場合も、複数のグループにまたがるGetNextRequest管理操作のためのインタフェース(get_next)を新たに定義する(図3(b))。

```
interface System : SNMPMgmt :: SmiEntry {// SNMPの場合
    readonly  attribute      DisplayStringType sysDescr;
    attribute  DisplayStringType sysName;  ..... // 省略
    // 複数IDL属性取得のためのIDL操作
    void get_attribute_list ( in    Sytem_AttributeList attributeList,
                              out   Sytem_ValueList valueList ) ;
    // 複数IDL属性設定のためのIDL操作
     void set_attribute_list ( in   Sytem_AttributeList attributeList,
                               in   Sytem_ValueList valueList ) ;   }
```

(a) 同一CORBAオブジェクトにおけるIDL操作定義の拡張

```
interface mo_scope_filter { // TMNの複数MOインスタンスへの処理
    // 型定義省略
    // 複数IDL属性取得のためのIDL操作
    void get_scope_filter ( in ObjectReference base, in ScopeType scope
          in FilterType filter, out GetResults selectedMOs ) ;
    // 複数IDL属性設定のためのIDL操作
    void set_scope_filter ( in ObjectReference base, in ScopeType scope
          in FilterType filter, in ValueList valueList ) ;    }
interface get_next { // SNMPのGetNextに対応
    // 型定義省略
    void get_next ( in ObjectTypeList typeList, out ValueList valueList ) ;}
```

(b) 複数CORBAオブジェクトにまたがるインタフェース定義の拡張

図3 IDL属性の取得/設定のための拡張

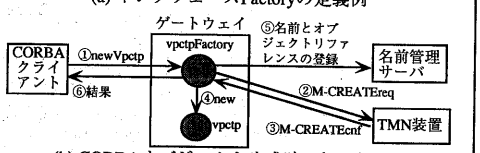
3.2.2 CORBAオブジェクト生成/削除の拡張

CORBAオブジェクトの生成/削除とTMN装置のMOインスタンスの生成/削除の管理操作を対応付けるため、生成/削除可能なMOクラス毎にインタフェースを新たに定義する(図4(a))。このインタフェースは生成と削除のIDL操作を持ち、このインタフェースに従うCORBAオブジェクトをFactoryと呼ぶ。

また、SNMPの場合にも、Factoryを用い、SNMP装置のインスタンスの存在に対応させてCORBAオブジェクトの生成/削除を可能とする。

```
interface vpctpFactory {
  // 型定義省略
  // CORBAオブジェクト生成のためのIDL操作
  void newVpctp ( in ObjectReference superior, // 上位のオブジェクト
                  in Name name,  // 生成するオブジェクトの名前
                  in AttributeList attributeList ) ;     // 属性値
  // CORBAオブジェクト削除のためのIDL操作
  void freeVpctp ( in vpctp singleVpctp ) ; } ;
```

(a) インタフェースFactoryの定義例



(b) CORBAオブジェクト生成時の処理例

図4 CORBAオブジェクトの生成と削除

3.3 IDL操作と管理操作の対応付け

JIDMではIDL操作と管理操作や通知の対応付けは規定していないため、JIDMの規定と3.2節で述べた拡張に関連する対応付けを表1に示す。ここでは、対象となるCORBAオブジェクトのインタフェースとIDL操作に対して、対応付けるTMNまたはSNMPの管理操作や通知を示す。

（1）属性の取得/設定とアクション

3.2.1節で述べたMOクラスのインタフェース毎のIDL操作get_attribute_list/ set_attribute_listは、TMNのattribute_listに複数の属性を指定したM-GET/M-SET管理操作、SNMPのVarBindListに複数のオブジェクトインスタンスを指定したGetRequest/SetRequest管理操作に対応付ける(表1①②⑦⑧)。また、インタフェースmo_scope_filterのIDL操作を、TMNのscopeとfilterを持つM-GET, M-SETおよびM-ACTIONの管理操作に(表1①②③)、インタフェースget_nextのIDL操作はSNMPのGetNextRequestに(表1⑦)対応付ける。

（2）MOインスタンスの生成／削除

3.2.2節で述べた図4(a)のFactoryのインタフェース定義の拡張に従って、TMNにおけるMOインスタンスの生成とCORBAオブジェクト生成時の対応付け例を図4(b)に示す。ここでは、生成のIDL操作に対して、M-CREATE管理操作によりTMN装置のMOインスタンスの生成、および、MOインスタンスに対応するCORBAオブジェクトの生成を行う(表1④)。一方、CORBAオブジェクトを削除するIDL操作の場合には、M-DELETE管理操作によりTMN装置のMOインスタンスの削除、および、対応するCORBAオブジェクトの削除を行う(表1⑤)。また、ゲートウェイは、TMN装置からのMOインスタンス生成/削除の通知受信時にもFactoryにIDL操作を発行する。

表1　IDL操作と管理操作や通知の対応付け

		管理操作と通知	インタフェース名	IDL操作名
TMN	①属性取得	M-GET - 単一属性 - 複数属性 - scope/filter指定	MOCのIF MOCのIF mo_scope_filter() *	<属性ラベル>_get() get_attribute_list() * get_scope_filter() *
	②属性設定	M-SET - 単一属性 - 複数属性 - scope/filter指定	MOCのIF MOCのIF mo_scope_filter() *	<属性ラベル>_set() set_attribute_list() * set_scope_filter() *
	③アクション	M-ACTION - 単一MOI - scope/filter指定	MOCのIF mo_scope_filter() *	<アクションラベル>() action_scope_filter() *
	④MOI生成	M-CREATE	MOCのFactory *	new<MOCラベル> *
	⑤MOI削除	M-DELETE	MOCのFactory *	free<MOCラベル> *
	⑥通知	M-EVENT-REPORT	Notification()	<通知ラベル>()
SNMP	⑦OBJT取得	GetRequest - 単一OBJT - 複数OBJT GetNextRequest	グループ/エントリのIF グループ/エントリのIF get_next() *	<OBJTラベル>() get_attribute_list() * get_next() *
	⑧OBJT設定	SetRequest - 単一OBJT - 複数OBJT	グループ/エントリのIF グループ/エントリのIF	<OBJTラベル>() set_attribute_list() *
	⑨通知	Trap	SnmpNotifications()	snmpTrap()

(注) IF：インタフェース、MOC：MOクラス、MOI：MOインスタンス、OBJT：オブジェクトタイプ、
グループ/エントリ：グループまたはテーブルエントリ、*：JIDMのIDL定義を拡張した定義

SNMPの場合には、ゲートウェイは障害の検出や動的なインスタンスの変化を検出するため、SNMP装置へGetNextRequestを順次発行するポーリングを行う。新たなオブジェクトを検出した場合、オブジェクトを生成するIDL操作をFactoryに発行し、CORBAオブジェクトを生成する。一方、オブジェクトの削除を検出した場合、対応するCORBAオブジェクトを削除するIDL操作をFactoryに発行する。

上記TMNとSNMPのいずれの場合にも、ゲートウェイは、CORBAオブジェクトの生成の場合には名前管理サーバに当該CORBAオブジェクト(名前とオブジェクトリファレンス)の登録を、削除の場合には名前管理サーバに当該CORBAオブジェクトの登録抹消を合わせて行う。

（3）通知

TMNのM-EVENT-REPORT受信時には、パラメータeventTtypeに基づいてIDL操作を決定し、MOインスタンス名からCORBAオブジェクト名への変換等のパラメータの変換後、IDL操作をCORBAクライアントへ発行する(表1⑥)。

また、SNMPのTrap受信時には、パラメータIPアドレスやgeneric-trap番号等の値からIDL操作snmpTrapのパラメータを設定し、CORBAクライアントへ通知する(表1⑨)。

3.4　CORBAオジェクトの名前対応付け

CORBAオブジェクトは2.1節で述べた名前の木構造で管理され、TMNのMOインスタンスとSNMPのオブジェクトの識別名の体系とは異なるため、それぞれ、以下の規則により対応付ける。

(1)TMNの場合

rootの下に①ゲートウェイのNC、②名前木の葉以外のMOインスタンスの相対識別名毎に対応するNaming Context(NC)、③上記②の各NCに対して同じレベルに対応するName(NA)、④名前木の葉のMOインスタンスの相対識別名に対応するNAから構成する。

(2)SNMPの場合

rootの下に①ゲートウェイ、②SNMPエージェント、③グループ名に対応するNC、④上記③の直下でグループに対して一つだけ規定されるオブジェクトタイプを示すNA、⑤上記③の直下にテーブルエントリに対応して複数存在するNAから構成する。

図5にSNMPにおけるMIBII[7]の場合のNaming Graphの構成例を示す。ここでは、interfacesグループのテーブルエントリの一つを/gateway#1/agent#1/interfaces/ifEntry#1/と名付けている。

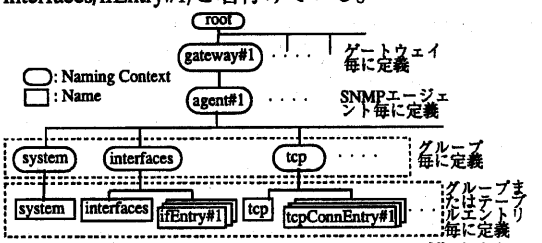


図5 SNMPにおけるNamig Grapghの構成例

4. CORBA/SNMPゲートウェイの実装

3章で提案した収容方式を実証するため、SNMP装置を収容するCORBA/SNMPゲートウェイ(以下、CORBA/SNMP GWと呼ぶ)を実装した。CORBAはIONA社Orbix、計算機はSUN、プログラム開発言語はC++を使用した。また、SNMPの任意の管理情報定義に対応可能とするため、SNMPの管理情報定義からCORBA/SNMP GWプログラムを自動生成させた。

4.1 システム構成

CORBA/SNMP GWは、クライアントからのIDL

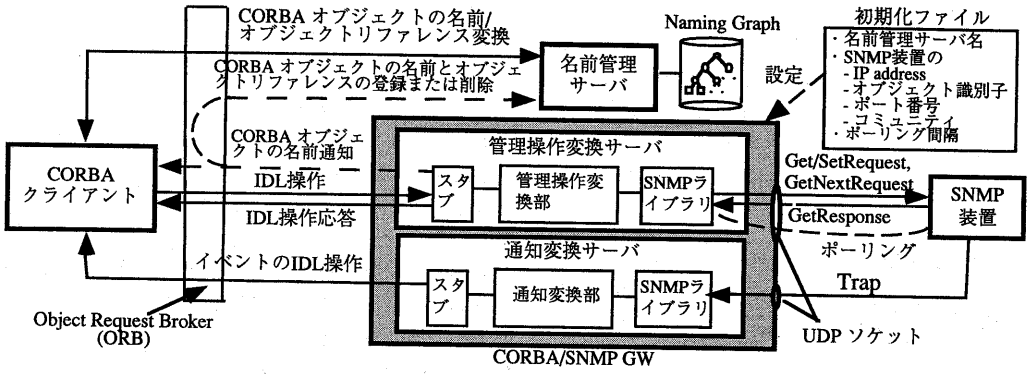


図6 システム構成

操作とSNMP装置からの通知が非同期に発行されるため、1)管理操作変換サーバと2)通知変換サーバの2つから構成し、それぞれ別プロセスとして実現した(図6)。両サーバは、SNMPの管理操作や通知とIDL操作との変換を行なう管理操作変換部あるいは通知変換部に加えて、IDL操作の符号化/復号を行うスタブ、SNMP PDUの符号化/復号を行うSNMPライブラリから構成される。管理操作変換部では、オブジェクトタイプ毎に、SNMPの管理操作とIDL操作との変換を行なうC++の関数(メソッド)を用意する。

管理操作変換サーバはCORBA/SNMP GW起動時およびCORBAオブジェクトが生成される時、CORBAオブジェクトの名前とオブジェクトリファレンスの組みを名前管理サーバに登録する。また、名前管理サーバのあるホスト名や、SNMP装置のIP address、SNMPオブジェクト識別子、コミュニティ等のSNMP装置の情報、ならびにポーリングを行う間隔を初期化ファイルに設定し、起動時に読み込んで変換に使用する。

4.2 CORBA/SNMP GWの処理

CORBA/SNMP GWは、基本的には3.3節のIDL操作と管理操作の対応付けに従った処理を行う。このうち、3.3節(1)の属性値取得の処理の詳細を以下に述べる。

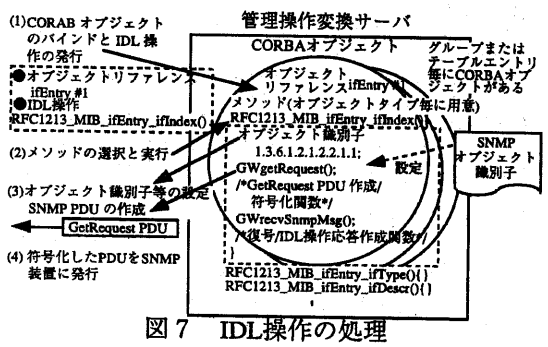


図7 IDL操作の処理

CORBAクライアントはオブジェクトリファレンスにより、SNMPのグループまたはテーブルエントリに対応するCORBAオブジェクトにバインドし、オブジェクトタイプの値取得または設定のためのIDL操作を発行する(図7(1))。CORBAオブジェクトは、IDL操作の名前をキーとして、オブジェクトタイプ毎に用意された関数(メソッド)を選択し、実行する(図7(2))。

このメソッドはすべてのCORBAオブジェクトに共通のSNMPライブラリ(例えば、GWgetRequest())を用いてPDUを作成し、オブジェクト識別子等のパラメータを設定し(図7(3))、次いでPDUを符号化してSNMP装置に発行する(図7(4))。

GetRequestに対するGetResponseを受信した場合、取得値をSNMP PDU作成ライブラリで復号し、IDL操作応答に詰め替えてCORBAクライアントに返す。

GetNextRequestおよび複数のSNMPオブジェクトへの管理操作に対するGetResponseを受信した場合、取得値のSNMPオブジェクトを特定し、対応するCORBAオブジェクトの名前をCORBAクライアントに応答する必要がある。このため、GetResponseのVarBindListに含まれるSNMPオブジェクト識別子から、対応するオブジェクトリファレンスを抽出し、名前管理サーバでCORBAオブジェクトの名前に変換させ、取得値と合わせてCORBAクライアントへ応答する。

4.3 プログラムの自動生成

SNMPの任意の管理情報定義に対応可能とするため、CORBA/SNMP GWプログラムジェネレータを実装した。図8にCORBA/SNMP GWプログラムの自動生成手順を示す。プログラムジェネレータは、SNMPの管理情報定義から対応するIDL定

義と管理操作変換部のプログラムを生成する。また、生成したIDL定義からIDL コンパイラを用いてスタブを生成する。通知変換部とSNMPライブラリは、SNMP の管理情報定義に依存しないため予め用意した。これらをコンパイル・リンクし、CORBA/SNMP GWプログラムを生成する。

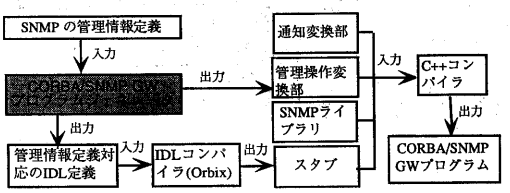


図8　プログラムの自動生成

5．評価と考察

5．1 性能評価

MIB IIの管理情報定義に対して、自動生成したCORBA/SNMP GW プログラムを使用し、応答時間を計測した。図9 に試験構成を、表2 に応答時間を示す。表2の測定項目のうち、JIDM規定を拡張したIDL操作を使用したものは表2(b)(d)(e)である。なお、応答時間は図9(1)～(10)の処理時間の合計とした。

表2の全ての測定項目の応答時間は13～30ms程度であり、このうちゲートウェイの処理時間は6～17ms程度でそれほど大きなオーバヘッドでなく、実用的な性能を達成していると考えられる。

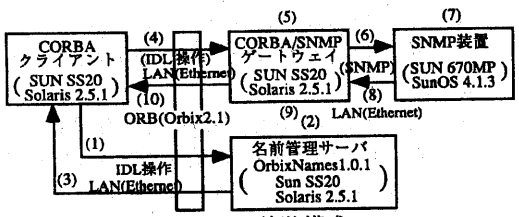


図9　試験構成

表2 プログラムの応答時間

測定項目	応答時間(ms)
(a) RFC1213_MIB_sysUpTime() によるsysUpTime(TimeTicks型) の1個の属性値取得	17.3 (6.2)
(b) get_attribute_list() によるsysUpTIme, sysObjectID, sysDescの3個の属性値取得	29.8 (16.8)
(c)RFC1213_MIB_system_sysLocation() によるsysLocation(DisplayString型) の1個の属性値設定	16.7 (6.2)
(d)set_attribute_list() によるsysLocation(DisplayString型) 等の2個の属性値設定	24.7 (11.4)
(e)get_next()による sysNameの次のifNumberの属性値取得	20.8 (6.2)
(f)snmpTrap()による Trap(coldStart)の受信	13.0 (9.5)

(注1) 応答時間の括弧内の数値はゲートウェイの処理時間(図 9 の(5)と(9)の和)を示す。
(注2) 各数値は100回の試行の平均値。

5．2　IDL定義の拡張の効果

複数のIDL属性の取得(表2(b))および設定(表2 (d))の応答時間は、それぞれ29.8ms、24.7msとなる。これを単一IDL属性の取得および設定を順次行った場合には、それぞれ51.8ms、34.4msとなり、IDL定義の拡張によるIDL操作と管理操作の個数削減の効果が大きいことがわかる。SNMPでは、ATM(非同期転送網)におけるVP(Virtual Path)の設定等のようにGetRequestやSetRequestに複数のオブジェクトを設定する場合は多く、このような観点からも本拡張は有効と考えられる。さらに、表2(e)は複数オブジェクトに渡る属性値の取得となるが、同一オブジェクトの場合の応答時間と同等に実現できた。

5．3　プログラム規模

CORBA/SNMP GWプログラムジェネレータが生成する管理操作変換部およびスタブのプログラム規模は管理情報定義に依存し、システムやTCP/IP等の管理を行うためによく使用されるMIB II (17個のグループまたはテーブルエントリを含む)の場合、それぞれ、16.1Kstep、7.8Kstepであり、コンパクトなプログラムが生成されている。また、管理情報定義に依存しない Trap 変換部とSNMPライブラリは、それぞれ、0.2Kstep、6.7 Kstep であった。

6．おわりに

本稿では、CORBAへのTMN/SNMP装置の収容方式を提案した。ここでは、JIDMが規定するTMNやSNMPの管理情報定義とIDL定義との対応付けの規定をベースとして、そのIDL定義の拡張し、また、新たにIDL操作と管理操作の対応付け、オブジェクトの名前の対応付けを規定した。提案方式を実証するため、SNMP装置を収容するCORBA/SNMP ゲートウェイを実装し、処理時間等の評価を通して提案方式の有効性を示した。

最後に日頃ご指導頂くKDD研究所村上所長、鈴木健二副所長に感謝する．

参考文献

[1]: ITU-T Rec. M.3010, "Principles for Telecomunications Management Network", Oct. 1992.
[2]: J. Case et al., "A Simple Network Management Protocol (SNMP)", IETF, RFC1157, 1990.
[3]: Object Management Group, "The Common Object Request Broker: Architecture and Specification Rev. 2.0", July 1995.
[4]: X/Open Preliminary Specification, "Inter Domain Management: Specification Translation", Nov. 1996.
[5]: 堀内, 吉原, 小花 "分散オブジェクトによるネットワーク管理のための TMN/SNMP 収容方式", 情処全大, 3S-04, Sep. 1997.
[6]: 吉原 , 堀内, 小花 "CORBA/SNMPゲートウェイの実装と評価", 情処全大, 3S-05, Sep. 1997.
[7]: K. McCloghrie et al "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", IETF, RFC 1213, Mar. 1991.