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

3S－4

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

1．はじめに

ネットワーク管理システムの構築では分散配置された複数のシステム間の協調が必要となり、その実現へ分散オブジェクト技術の適用が注目されている。ここでは、TMN(電気通信管理網)[1]やSNMP(Simple Network Management Protocol)[2]のネットワーク管理プロトコルを持つ装置(以下、TMN/SNMP装置と呼ぶ)を分散オブジェクト環境へ収容することが必須となる。

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の業界標準CORBA(Common Object Request Broker Architecture)[3]のIDL(インタフェース定義言語)定義へ対応付ける検討を行っている[4]。

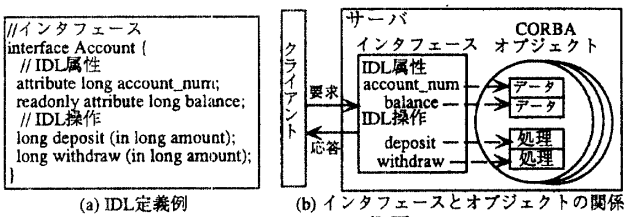
しかしながら、TMN/SNMP装置のCORBAへの収容には、このIDL定義だけでは十分でなく、IDL定義の拡張やIDL操作と管理操作の対応付けの新たな規定等が必要である。本稿ではJIDMのIDL定義をベースとしたCORBAへのTMN/SNMP装置の収容方式を述べる。

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)を実装し、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で定義し、インタフェースの定義には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)、データ型等を定義する。ここで、IDL属性と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、IDL操作とはクライアントがCORBAオブジェクトに依頼する処理の単位である。図1にIDL定義例、インタフェースとCORBAオブジェクトの関係を示す。

また、クライアントがIDL操作を呼出す際には、CORBAオブジェクトを指定するためにオブジェクトリファレンスと呼ばれるデータを与える必要がある。このオブジェクトリファレンスは、サーバ側でオブジェクト作成の際に生成し、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等を通じてクライアントに提供する。



(a) IDL定義例　　(b) インタフェースとオブジェクトの関係

図1 CORBAの概要

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

Hiroki HORIUCHI, Kiyohito YOSHIHARA and Sadao OBANA, KDD R&D Labs.

2．2 JIDMのIDL定義への対応付けの概要

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．TMN/SNMP装置のCORBAへの収容方式の提案

3．1 収容形態

CORBA環境にTMN/SNMP装置を収容するには、TMN装置のMOやSNMP装置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操作をTMN/SNMPの管理操作や通知に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、対応付けたCORBAオブジェクトの名前とオブジェクトリファレンスを名前管理サーバに登録する。

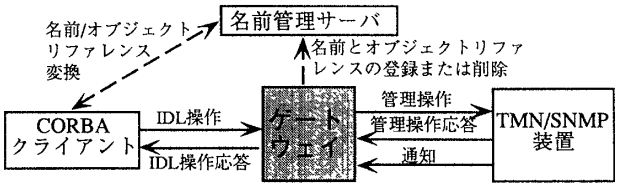


図2 TMN/SNMP装置のCORBAへの収容形態

3．2 IDL定義の拡張

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IDL定義を拡張する。

3．2．1 IDL属性の取得/設定のための拡張

JIDMのIDL定義では、TMNの属性型を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/設定におけるIDL操作とGetRequest等への変換においても同様な問題点がある。こ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操作を追加定義する。これらのIDL操作は、TMNのM-GET/M-SET管理操作(パラメータattribute_listに複数指定有)、SNMPのGetRequest/SetRequest管理操作(パラメータVarBindListに複数指定有)に対応付ける(図3(a))。

(2) 複数CORBAオブジェクトの場合

TMNの複数MOインスタンスに渡るM-GET等の管理操作(パラメータscope, filter有)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))。

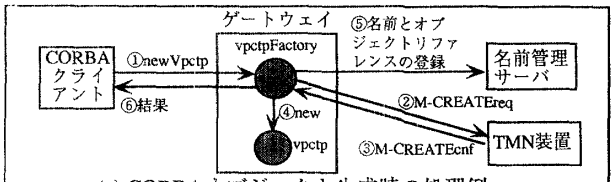
```
interface System : SNMPMgmt :: SmiEntry {// SNMPの場合
   readonly  attribute      DisplayStringType sysDescr;
   attribute  DisplayStringType sysName;  ..... // 省略
   // 複数IDL属性取得のためのIDL操作
   void get_attribute_list ( in    Sytem_AttributeList attributeList,
                             out   Sytem_ValueList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

```
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オブジェクトにおけるインタフェースの例

図3 IDL属性の取得/設定のための拡張



(a) CORBAオブジェクト生成時の処理例

```
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 ) ; } ;
```

(b) Factoryの定義例

図4 CORBAオブジェクトの生成と削除

SNMPの場合も、複数グループに渡るGetNextRequestのためのインタフェース(get_next)を新たに定義する(図3(b))。これらのインタフェースを持つCORBAオブジェクトはゲートウェイ起動時に生成する。

3.2.2 CORBAオブジェクトの生成/削除

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。このインタフェースは生成と削除のIDL操作を持ち、ゲートウェイ起動時にCORBAオブジェクト(Factoryと呼ぶ)として生成する。図4にFactory "vpctpFactory"が生成のためのIDL操作を受信した時の処理例とIDL定義例を示す。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に内部的なIDL操作を発行する。

また、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ポーリングを行い、生成/削除されたグループやテーブルエントリのインスタンスを新たに検出した場合にFactoryに内部的なIDL操作を発行する。

3.3 IDL操作と管理操作の対応付け

JIDMではIDL操作と管理操作の対応付けは規定していないため、3.2節の拡張したIDL定義に関連する対応付けも含め表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名とIDL操作名に対して、対応付けるTMNまたはSNMPの管理操作を示す。

表1 IDL操作と管理操作の対応付け

		管理操作	インタフェース名	IDL操作名
TMN	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() *
	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() *
	MOI生成	M-CREATE	MOCのFactory *	new<MOCラベル> *
	MOI削除	M-DELETE	MOCのFactory *	free<MOCラベル> *
	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() *
	通知	M-EVENT-REPORT	Notification( )	<通知ラベル>()
SNMP	OBJT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() *
	OBJT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() *
	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

3.4 CORBAオブジェクトの名前の対応付け規則

CORBAオブジェクトの名前はNaming Graph(NG)と呼ばれる木構造で表現され、節を示すNaming Context(NC)と葉を示すName(NA)からなる。この名前の体系は、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、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NC、③上記②の各NCに対して同じレベルに対応する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MIB-IIの場合のNG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。

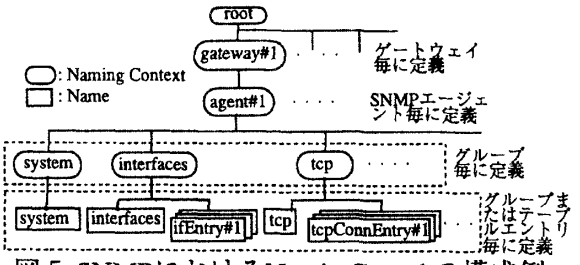


図5 SNMPにおけるNamig Grapghの構成例

4. おわりに

本稿では、分散オブジェクト(CORBA)環境へのTMN/SNMP装置の収容方式を提案した。ここでは、JIDMのIDL定義の拡張、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規則を提案した。最後に日頃ご指導頂くKDD研究所村上所長に感謝する。

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.