

MSA(管理システムエージェント)による分散透過な MIB の実現

西山 智 小野 智弘 小花 貞夫

国際電信電話(株) 研究所

OSI 管理のモデルは管理側アプリケーション (AP) と被管理側 AP との 1 対 1 の通信手順のみを規定している。このため、複数の被管理側 AP が MIB を分散格納している場合、管理側 AP には適切な被管理側 AP に対して管理操作を要求するための分散処理機能が必要であった。また、他の管理システムにシステム間接続する場合、これらの管理システム間の MIB のスキーマ定義は通常異なるため、管理 AP には、管理操作を接続相手のスキーマ定義に基づくものに変換するための操作/結果の変換機能が必要であった。これらの問題を解決し、管理 AP の実現を容易とするためには、管理 AP に対して MIB の分散やスキーマの違いを意識する必要のない一元的な MIB(分散透過な MIB) を提供するエージェントが重要である。本稿では、分散透過な MIB を提供するための管理システムエージェント (MSA) と呼ばれるエージェントのモデルを提案する。MSA は名前解析処理とイベント転送処理により管理 AP に MIB の分散を見せない。また、異なるスキーマ間の操作/結果の変換処理により管理 AP が一元的なスキーマの元で管理操作を行なうことを可能とする。さらに X.25 網を対象とする CNM を例に、処理方式を説明する。

1. はじめに

近年、OSI(開放型システム間相互接続) 管理^[1]、あるいは OSI 管理をベースとした TMN(電気通信管理網) に基づく網管理システムの構築が多数進められている。

OSI 管理のモデルでは管理オブジェクト (MO) の集合は論理的に管理情報ベース (MIB) と呼ばれる。管理対象のシステム (被管理システム) が複数存在する場合、個々の MO の情報 (属性値) は、管理システム側のアプリケーション (以下、管理 AP と呼ぶ) やそれら複数の被管理システム側のアプリケーション (被管理 AP) に分散格納される。OSI 管理のモデルでは、管理 AP と被管理 AP との 1 対 1 の通信手順のみを規定しており、被管理 AP 間あるいは管理 AP との間に MIB の分散している場合に必要となる分散処理については何も規定していない。このため、これまでは管理 AP 内にそのシステム独自の分散処理の機能を実装する必要があった。

Location Transparent MIB based on MSA (Management System Agent) Model
by Satoshi Nishiyama, Chihiro Ono and Sadao Obana
KDD R&D Laboratories

また、CNM(顧客網管理)^[2]用のゲートウェイや TMN でのサービス OS(オペレーションシステム) 等の管理システムでは、異なる MIB スキーマ定義に基づく既存の管理 AP が MIB の一部を管理している場合がある。このような場合、管理 AP は、その管理のための MIB スキーマ定義に加えて接続する管理システムの MIB スキーマ定義を持ち、異なるスキーマ定義に基づく操作/結果の変換を行なう機能を作り込む必要があった。

これらの分散処理機能やスキーマ間の操作/結果の変換機能は、MIB が分散している場合には常に必要となる汎用的な機能である。従って、この汎用的な機能を持つエージェントを作成し、管理 AP に対して MIB の分散やスキーマの違いを意識する必要のない一元的な MIB(分散透過な MIB) を提供すれば、管理 AP の作成や、システム変更に伴う修正が容易となる。本稿では、分散透過な MIB を提供するためのエージェントである管理システムエージェント (MSA) の概念とその処理方式を提案する。

以下、2.章では OSI 管理の MIB の概要について簡単に述べる。3.章では MIB の分散形態と従来の

MIBの実装方式の問題点について述べる。4.章では、分散透過なMIBを実現するためのMSAと呼ぶ概念を提案する。5.章ではMSAの処理方式を述べる。最後に6.章では、CNMを例に分散透過なMIBの実現例を示す。

2. MIBのモデルと操作

2.1 モデル

OSI管理では全ての管理対象は管理オブジェクト(MO)としてモデル化される。MOは管理対象間の包含関係によって木構造(包含木)に関係付けられ、MOの集合は木構造モデルを持つ仮想的なデータベースとみなすことができる。このデータベースは管理情報ベース(MIB)と呼ばれる。包含木では、MOの全体は仮想的に存在するルートに包含されている。包含木における節はMOを表し、枝は同一のMOに包含されるMOを識別するための相対識別名(RDN)を表す。ルートから包含関係を示す枝をたどることにより個々のMOは識別される。途中の枝に対応するRDNの並びは識別名(DN)と呼ばれる。各MOはそのMOの情報として複数の属性を持つことができる。また、属性には複数の属性値を持つものもある。

実際にはMIBは、管理対象の装置を代行するエージェントと呼ばれる被管理APが管理しており、マネージャと呼ばれる管理APが次節で述べる操作によりMIBを操作することで管理が行なわれる。但し、本稿ではエージェントと言う語を「代理的な動作をするプロセス」という意味で使用するので、OSI管理でのマネージャとエージェントという用語を使用しない。OSI管理でのマネージャとエージェントと言う語はそれぞれ管理AP、被管理APと置き換えて使用する。

なお、MIBには特殊な役割を行なうMOのクラスがいくつか定義されている。例えば、EFD(Event Forwarding Discriminator)は被管理APがMOに発生する各種のイベントを制御し、特定の管理APに通知する際に必要となる、通知先や条件を記述するためのMOクラスである。条件記述の1例を図1に示す。この例ではprotocolEntityクラスのMOでentityIDが"123"で始まり、かつseverityがminorでないあるいはbadPduCountが20以上である場合に、イベントを通知することを示している。

```
( objectClass equal to protocolEntity )
and ( entityID starts with "123" )
and (( severity not equal to minor )
or ( badPduCount greater than 20 ))
```

図 1: EFD でのイベント転送条件の例

表 1: CMIS 操作

M-GET	条件に合致する MO の情報を取得する
M-CANCEL-GET	M-GET 操作を中止する
M-SET	条件に合致する MO の情報を更新する
M-CREATE	条件に合致する MO を作成する
M-DELETE	条件に合致する MO を削除する
M-EVENT-REPORT	特定の事象が発生したことを管理 AP 側に通知する
M-ACTION	条件に合致する MO に何らかの手続きを実行させる

2.2 操作

MIBに対する操作は、OSIの応用層のサービスである共通管理情報サービス(CMIS)上の操作として表1に示す7種類が定義されている。

これらの操作は、M-EVENT-REPORTを除き、管理APが被管理APに要求する。M-EVENT-REPORTは被管理APから自律的に管理APに通知される操作である。

3. MIBの分散形態と従来のOSI管理の実装における問題点

3.1 MIBの分散形態

OSI管理に基づく管理システムでは、以下のようなMIBの分散の例が見られる。

(1) MO位置の分散

MIBが複数の被管理AP間、あるいは管理APとの間に分散している(図2(1a))。例えば、複数の交換機を管理する場合、個々の交換機用の被管理APにMIBが分散管理される。さらに、1つのMOの情報が分散している場合(以下このような場合をMO内の分散と呼ぶ)もある(図2(1b))。例えば、MOの情報には装置名やその機能等のように比較的变化のない情報と、その装置の状態のように時々刻々と変化する情報がある。このため管理システムによっては、検索や格納の効率を考慮して、前者は被管理APではなく、管理APや或は他のAP(例えばX.500^[3]等によるディレクトリサーバ)に格納する場合等が例として挙げられる。

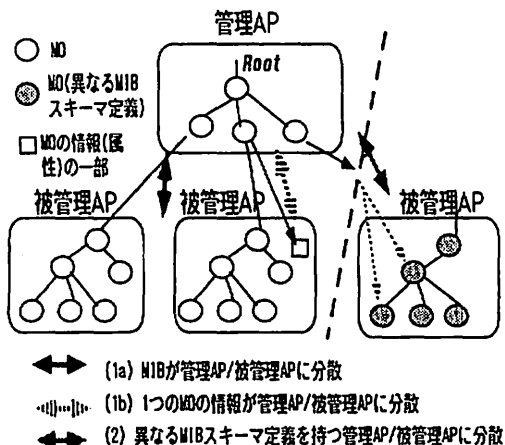


図 2: MIB の分散形態

(2) 異なる MIB スキーマ定義の存在

他の既存管理 AP に対して管理システム間接続を行なう場合等では、MIB が分散している被管理 AP 間あるいは管理 AP との間で MIB のスキーマ定義が異なる場合がある (図 2(2))。具体的には、MIT(管理情報本)の構造が異なる、あるいは含まれる属性やオブジェクトクラス等について MO の定義が異なる、等がある。

3.2 従来の OSI 管理の実装における問題点

従来の OSI 管理の実装では以下のような問題点があった。

● MO の分散に対する分散処理機能が必要

OSI 管理のモデルでは、CMIS 操作による管理 AP と被管理 AP 間の 1 対 1 の通信のみをモデル化しており、分散した被管理 AP に対する分散処理は実装に任されている。従って、これまで管理 AP に分散処理の機能を作り込み、管理 AP が被管理 AP と直接 CMIS 操作の授受を行なう形態 (図 3) による実装が行なわれてきた。この形態では、

- (1) 管理 AP 毎に分散処理機能の作り込みが必要、
- (2) 管理対象の追加等、分散形態の変更により管理 AP の変更が必要

という問題が生じる。

● 管理 AP にスキーマの違いを吸収する操作変換機能が必要

MIB の分散形態 (2) で示したように、管理 AP

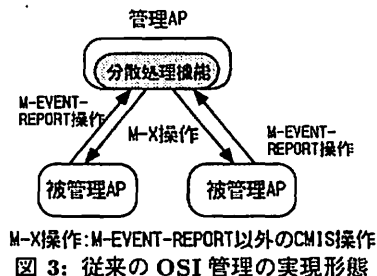


図 3: 従来の OSI 管理の実現形態

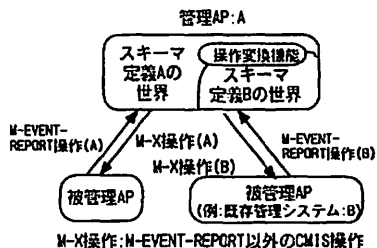


図 4: 異なる MIB スキーマ定義を持つ被管理 AP との接続

と被管理 AP で管理する MIB のスキーマ定義が異なる場合、それらの間で操作/結果の変換を行なう必要がある。一般にはこのような場合、管理 AP には図 4 に示すようにその管理システムの MIB スキーマ定義に加えて接続する被管理 AP の MIB スキーマ定義を持ち、これらのスキーマ定義に基づく操作/結果の変換を行なう機能が必要となる。この機能も管理 AP の本来の管理目的とは異なるため、管理 AP の複雑化の原因となる。

本稿では、上記の問題点を解決するために、MIB の分散を隠蔽すること (位置透過性) と MIB 内のスキーマの違いを隠蔽すること (スキーマ透過性、以下これらを併せて分散透過性と呼ぶ) を実現する機能を管理 AP から分離し、独立したエージェントとして提供することを提案する。このようなエージェントにより、管理 AP は MIB の分散やスキーマの違いを意識する必要がなく、そのエージェントに対して一元的に CMIS 操作を授受することで管理が実現できる。これまで、OSI 管理の標準化作業においても、分散透過な MIB の必要性和そのためのプロキシエージェントと呼ばれるエージェントの概念が議論されてきたが、その実現方式はこれまで明確にされていない。

次章以降では、筆者らが提案する、分散透過な MIB を提供するための管理システムエージェント

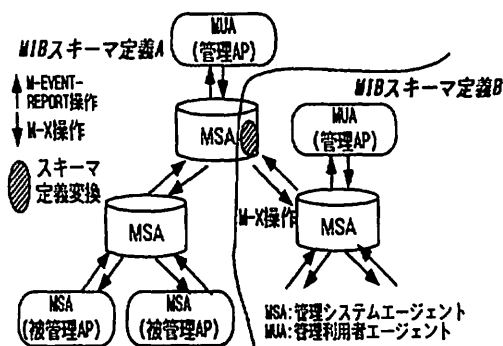


図 5: MSA による分散透過な MIB を用いた OSI 管理の実現

(MSA) の概念と、その実現方式について示す。

4. MSA の提案

ここでは、MIB に OSI ディレクトリ^[3]情報ベース (DIB) での DSA (ディレクトリシステムエージェント) と類似した、分散協調を行なうエージェントである MSA (管理システムエージェント) という概念を導入し、分散透過な MIB を実現する。即ち、MIB は 1 つ或は複数の MSA から構成され、各々の MSA は MIB の一部を格納する。MSA は、DSA と同様に、分散処理やローカルな MO 情報の格納の機能を持つ。MSA は分散処理のための知識 (分散処理知識) を持ち、他の MSA と協調動作することで全体として位置透過な MIB を提供する (図 5)。また、異なる MIB スキーマ定義を持つ MIB を分散収容するために、MSA はスキーマ変換情報を持ち、異なる MIB スキーマ定義に基づく操作間の変換を行なう。管理 AP は CMIS 操作を用いて単一の MSA にアクセスすることで、MIB の分散やスキーマ定義の差異から独立することができる。管理 AP は MIB の利用者であるから、DIB での類推により MUA (管理利用者エージェント) と位置付けられる。従来の個々の被管理 AP は分散やスキーマ定義の差異による操作変換機能を持たない MSA の一種とみなすことができる。

5. MSA の処理方式

本章では、MIB の位置透過性とスキーマ透過性を提供するために MSA が行なう分散処理および操作変換・結果の変換処理の処理方式について示す。

5.1 前提条件

ここで提案する分散透過な MIB を提供するための MSA の分散処理方式は、以下の条件を前提としている。

- 管理 AP 及び被管理 AP との間は CMIS/CMIP で通信することとし、既存の被管理 AP に変更を加えないこと。
- 通信相手によって、使用する MIB スキーマ定義が既知であること。

前者の条件は、標準準拠の被管理 AP を MSA としてそのまま分散透過 MIB に組み込むために必須である。また、CMIS 操作は使用する MIB のスキーマ定義をプロトコル上で交換できないため、後者は必要となる。

5.2 分散処理機能-位置透過性の提供

MIB の分散に対して、位置透過性を提供するために、MSA は分散処理機能を持つ。この機能は上位 MSA (あるいは MUA) から要求される M-EVENT-REPORT を除く操作 (M-X 操作と呼ぶ) を適切な下位 MSA に転送するための名前解析処理と、下位 MSA からの M-EVENT-REPORT 操作を適切な上位 MSA (あるいは MUA) に転送するためのイベント転送処理からなる。また、これらの転送を行なうために MSA は分散処理知識を持つ。以下では、MSA の持つ分散処理知識、名前解析処理およびイベント転送処理について述べる。

5.2.1 分散処理知識

2.章で述べたように、DIB の場合と異なり MIB では管理 AP と被管理 AP 間に操作の方向性がある。そこで、MSA は管理 AP から被管理 AP への操作 (M-X 操作) のための分散処理知識と、被管理 AP から管理 AP への操作 (M-EVENT-REPORT 操作) のための分散処理知識を別に持つ必要がある。

M-X 操作のための分散処理知識

- **内部参照 (INTR):** ローカルに格納する MO 情報へのポインタ。
- **分散参照 (DSTR):** ローカルに格納する MO の一部が MIB スキーマ定義が等しい他の MSA に格納されている場合、その MSA のアクセスポイント情報 (AE 名称と PSAP アドレス)。同じ識別名を持つ INTR に付帯して存在する。1 つの識別名に対して DSTR は複数存在しえる。
- **下位参照 (SUBR), 不特定下位参照 (NSSR):** ローカルに格納する MO の直接下位の MO を

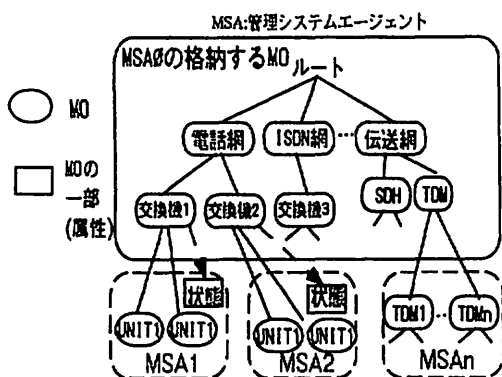


図 6: MIB の例

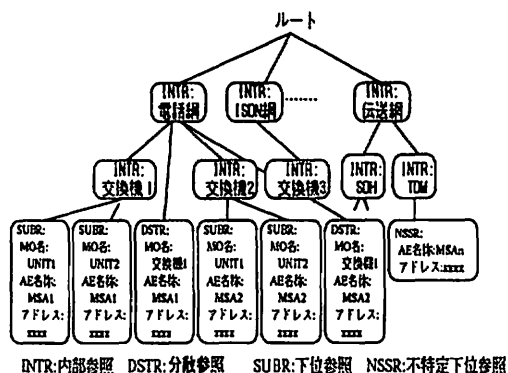


図 7: 知識木の例

格納する MSA のアクセスポイント情報. MO 名が不明な場合は NSSR を用いる.

M-EVENT-REPORT 操作の転送を行なうための分散処理知識

- 上位参照 (SUPR): M-EVENT-REPORT 操作の転送先 MSA のアクセスポイント情報 (AE 名称と PSAP アドレス) と送出条件の並びである. SUPR は DSA の場合と異なり, その MIB 内に存在する EFD が示す内容を知識化したものであるため, 複数の SUPR がシステムに存在しえる.

これらの情報は, その MSA 内で扱う MIB スキーマ定義毎に, 木構造 (KT:知識木) を構成する. 図 6 に示す MO に対して, MSA0 が持つべき知識の例を 図 7 に示す.

5.2.2 名前解析処理手順

名前解析処理では, DSA と類似の分散名前解析処理を実装し, M-X 操作に関する位置透過性を提供する. 異なる MIB スキーマ定義に基づく複数の MIT (管理情報木) が MSA 内に存在するため, MSA の名前解析処理では, 操作で与えられた提示名 (purported name) に加えて, 対象とする MIB のスキーマ定義を識別するための ID (以下スキーマ ID と呼ぶ) を MIT を識別するパラメタとして用いる. M-X 操作に対する名前解析処理手順を以下に示す.

- 名前解析: 提示名を基に, スキーマ ID の KT を参照してローカルに格納する MO (INTR が存在する場合) か, 他の MSA が格納する MO か, あるいは名前誤りかを判定する.
- ローカル MO に対する処理: もし INTR と共に同一識別名を持つ DSTR が存在した場合, DSTR の示す MSA にも同一の操作を転送する. INTR を元に, MO の内容を読み出す. もし, MO の内容にスキーマ変換情報が含まれていた場合, 5.3 節で述べる操作/結果の変換処理を行ない, ローカル MO に対する操作結果, DSTR で転送した MSA からの操作結果と合成し, 結果を要求元 (MSA か MUA) に返送する.
- 他の MSA への操作転送: 他の MSA が管理する MO に対する操作の場合は, 提示名に最大限一致する識別名を持つ SUBR や NSSR を KT から選択し, その知識の示す MSA に操作を転送し, 結果を要求元に返す.
- 操作分割, 結果の変換: 複数の MSA に跨る操作は SUBR や NSSR に従って操作を分割し, 分割後の操作を各々の MSA に送付する. 結果受信時はこれらの結果を合成して要求元に返送する.

DIB と異なり MIB では MSA 間の CMIS 操作の方向が固定的であり, また CMIS 操作には MSA 間の分散処理のためのパラメタがない. このため以下の点で MSA での名前解析処理は DSA でのそれとは異なる.

- MSA での名前解析は常に提示名の全体に対して行なわれる. DSA 間では転送元の DSA が提示名中の既に解説した部分を通知できたが, MSA 間ではこの情報を交換できないためである.

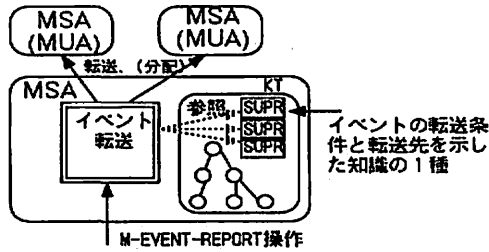


図 8: M-EVENT-REPORT 操作の転送

- 提示名に対して名前解析が全く行なえない場合は、操作誤りとする。

5.2.3 イベント転送処理

M-EVENT-REPORT 操作については、図 8 に示すように、名前解析処理ではなくイベント転送処理により転送制御を行なう。イベント転送処理は、他の MSA からの M-EVENT-REPORT 操作受信時、および自 MSA 内でイベントが発生した場合に以下の処理を行なう。

- **M-EVENT-REPORT 操作の受信:**MSA が M-EVENT-REPORT 操作を受信した場合、そのスキーマの KT に含まれる SUPR に示された、イベント転送条件に合致するか否かを順次判定し、条件に合致した SUPR の示す MSA(或は MUA) に通知する。
- **M-ENEVT-REPORT 操作の発生時:**MSA がローカルに格納する MO に、(ローカルに管理する)EFD で指定された MO の変化が発生した場合、EFD と SUPR の両方の条件を満たす MSA(或は MUA) に M-EVENT-REPORT 操作により通知する。

5.3 操作/結果の変換処理 - スキーマ透過性の提供

操作/結果の変換処理は、ある MIB スキーマ定義に基づく CMIS 操作を、他のスキーマ定義に基づく操作に変換し、他の MSA に転送すると共に、その操作結果から元の CMIS 操作への結果に変換する処理である。

5.3.1 操作変換の前提

- 操作変換のためのスキーマ変換情報は、基本的には個々の MO に持つ。このため、操作変換を行なう MSA は変換元となる全ての MO をローカルに管理する必要がある。

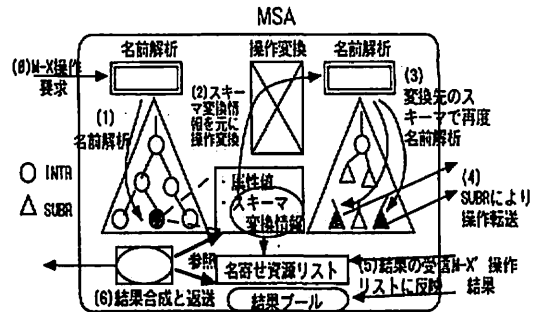


図 9: M-X 側からの操作の変換

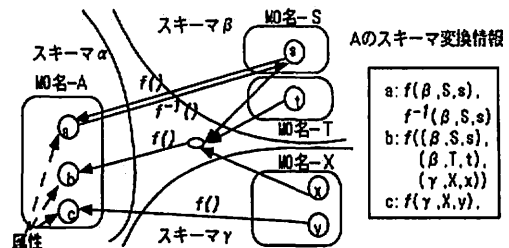


図 10: スキーマ変換情報

- MOクラス毎に,MOの生成およびEFD内容の変更時に必要となるスキーマ変換規則を持つ.

5.3.2 M-X 操作の変換

M-X 操作の変換を、図 9 に示す。

(a) スキーマ変換情報の読みだし スキーマ変換が必要な場合、名前解析処理の結果見つかった内部参照の示す MO 情報には、ローカルな MO 情報 (MO のクラスやローカルに格納する属性) に加えて、図 10 に示すスキーマ変換情報が含まれている。スキーマ変換情報は以下のフィールドからなる。

- 変換元の属性 OID(その属性に付与された ASN.1 のオブジェクト識別子)
- f (資源識別子の並び)
- f^{-1} (属性値, 資源識別子) の並び

ここで、資源識別子は、スキーマ ID、識別名、属性 OID からなる。ここでも操作/結果の変換処理に MIB の分散を意識させないために、その資源を持つ MSA のアクセスポイントを直接指定する代わりに、識別名を指定し、名前解析処理を再度行なうことで、アクセス先 MSA を決定する。また、 $f()$ は指定した属性の値を得るための関数で、その属性 OID が示す ASN.1 型の値を返す。 $f^{-1}()$ は指定した属性の値を、それが構成する資源 (MO の属性) の値

に変換する関数で、更新が変換先に反映可能である場合に定義される。

(b) 名前解析処理および他 MSA への転送 変換後のスキーマIDの元で、合成後の操作の名前解析を再度実施し、分散処理知識に従って指定されたMSAに操作を依頼する。

(c) 結果の変換 操作を依頼したMSAから結果を受信すると、名寄せ資源リストの該当資源を受信済とする。名寄せ資源リスト中の全ての資源が受信済になった場合、スキーマ変換情報を用いて、要求されたMOの情報に変換する。

5.3.3 M-EVENT-REPORT の変換

M-EVENT-REPORT 操作の変換を図11に示す。

- M-EVENT-REPORT 操作を受信すると、イベント発生元のKTでのSUPRを用いてイベント変換部にイベントが転送される。イベント変換部では、転送先のMIBでのEFDの記述内容を発生元のMIBのクラス名や属性名の条件に変換し、それらをキーとして編集したイベント逆引き表を維持しており、その情報を元に転送先のM-EVENT-REPORT操作を作成する。この操作は通常のイベント転送処理により分配、転送される。
- スキーマ変換情報をMO毎に持つため、転送先のスキーマ定義でのEFDは変換を行なうMSAにローカルに格納されている。従って、操作変換を行なうMSAはこのEFDと対応する発生元でのEFDの内容を一致させる必要がある。転送先のEFDの追加、更新、削除操作を受けると、クラス毎のスキーマ変換規則に基づき、発生元のMIBでのEFDを作成、更新、削除する操作を作成する。併せて、イベント逆引き表も更新する。

6. CNM への適用例

本章では、MSAの適用例として、図12に示すようなX.25網を対象とするCNM^[2]に適用した場合を示す。また、アラーム通知サービスを例に、このCNMシステムおよび通信提供者側の網管理システムで使用するMO定義を図13に示す。

図14にCNM用MSAでの分散処理知識およびイベントの変換例を示す。例えば顧客1のcnmX25Entity(名前:44081xxx)はローカル属性cnmX25EntityIDを持ち、また属性

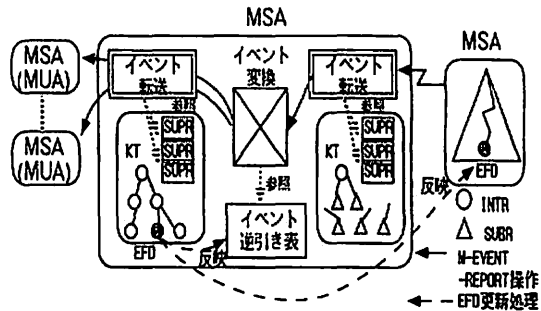


図 11: M-EVENT-REPORT 操作の変換

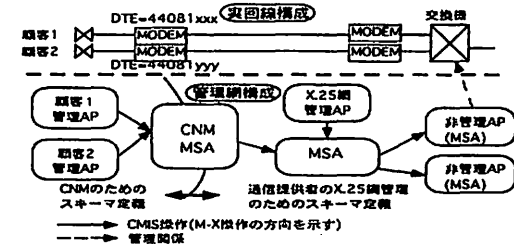


図 12: X.25 網および網管理用網の構成例

“DMI”:administrativeStateは通信提供者のMIBでの(X.25網、交換機1,81xxx)の同名の属性にマッピングされる。また、イベントについては、CNMでのcnmX.25端点のイベントは概ね通信提供者側のX.25端点のそれらに対応するので、X.25端点に関するイベントは全てイベント変換処理に転送することが提供者側のKTのSUPRに記述される。ここで、CNM側でcnmX.25端点のオブジェクト生成通知をイベントとして通知するというEFDが作成されると、通信提供者側でX.25端点に関する同

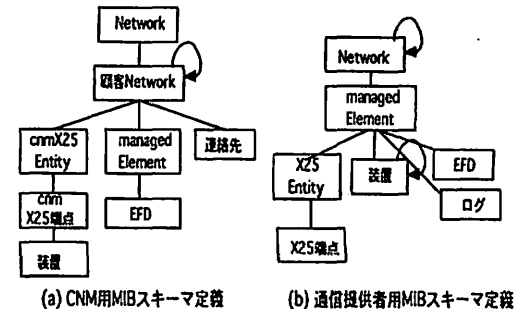


図 13: X.25 網での CNM 用および通信提供者用 MIB スキーマ定義

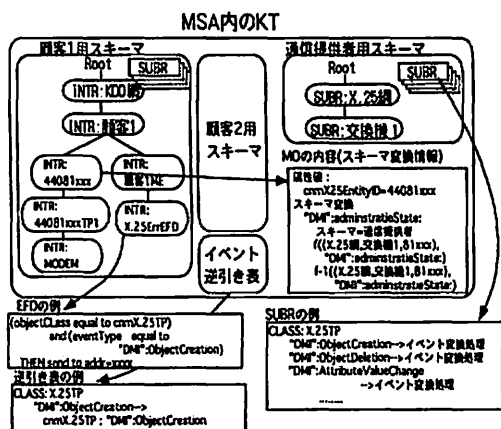


図 14: CNM 用 MSA の分散処理知識および MO 情報の例

様な EFD が生成されるとともに、逆引き表に X.25 端点のオブジェクト生成イベントを cmnx.25 端点のオブジェクト生成イベントに変換する情報が加えられる。

なお、顧客が通信提供者の情報にアクセスする CNM では、顧客に対して MIB の他の部分がアクセスできないようにするアクセス制御が重要である。本稿で提案する MSA では複数のスキーマ定義に基づく MIT が MSA 内に存在できる。そこで、これを利用して顧客毎に顧客専用の MIB を提供することで、アクセス制御の問題を解消することができる。

7. 考察

- MSA の概念に基づいて網管理システム構築を行なっていれば、例えば CNM のように、新規管理システムを既存管理システムに接続するような場合にも、ゲートウェイとなる MSA を新規に追加することで、容易に対応できることが分かった。なお、本提案方式では名前解析やイベント転送のための分散処理知識の更新についてはローカル事項と仮定したが、その更新メカニズムを今後検討する必要がある。
- CMIS の更新操作では操作の原子性が指定できるため、管理 AP の発行した更新操作が複数の MSA 間に跨った場合にも原子性を保証する必要がある。また、MSA でスキーマ変換を行なうと、単一の操作が複数の操作に変換される場合があり、この場合も操作間で更新の原子性

を保証する必要がある。これらの場合、例えばトランザクション処理 (TP) プロトコルとの併用^[4]等により、トランザクション機能を実現する必要がある。

- 本稿では、異なるスキーマ間での属性値の変換を関数を提供することにより行なう方法を提案した。現在、OSI 管理では形式記述言語により MO 間のデータの依存関係を記述する検討がなされている。スキーマ変換情報は、異なる MIB スキーマ定義間の元で、MO 間のデータの依存関係を記述したものであるから、形式記述言語によりスキーマ変換情報を記述する方法についても今後検討する必要がある。
- 従来、管理 AP の高速化手法として、MO 情報を管理 AP 内にキャッシュすることが行なわれてきた。MSA がこのような機能を基本機能として具備することで、管理 AP の実現が容易となる。

8. まとめ

本稿では、管理 AP に分散透過な MIB を提供するために、管理システムエージェント (MSA) を導入する提案とその実現方法を示した。MSA は、ローカルな MO の情報格納機能に加えて、名前解析処理機能やイベント転送機能により MIB の分散に対する位置透過性をその利用者 (MSA あるいは MUA) に提供する。また、操作/結果の変換機能により MIB が異なる MIB スキーマ定義を持つ被管理 AP 間に分散している場合に、MIB のスキーマ透過性を利用者に提供する。さらに本稿では X.25 網に対する CNM システムを具体例としてスキーマ変換の例を示した。今後、筆者らが既に開発した MIB 用データベース ASSIST/M^[5]をベースに MSA の実装を行なっていく予定である。最後に日頃御指導頂く KDD 研究所浦野所長、鈴木次長に感謝します。

参考文献

- [1] ITU-T 勧告 X.700 シリーズ “OSI 管理,” 1991.
- [2] ITU-T 勧告 X.160 シリーズ “公衆データ網のための顧客網管理,” 1995.
- [3] ITU-T 勧告 X.500 シリーズ “OSI ディレクトリ,” 1988.
- [4] ITU-T 勧告 X.702, “OSI: Application Context for Systems Management with Transaction Processing,” 1995.
- [5] 西山他, “OSI 管理情報ベース (MIB) 用データベースの設計と実装,” 情報処理報告 DPS 95-7, 1993.