

TMN管理情報ベース(MIB)のためのビュー変換の実現方式

堀内浩規 吉原貴仁 杉山敬三 小花貞夫

国際電信電話株式会社 研究所

TMNにおいて管理対象となるNE装置は、管理オブジェクトの集合である管理情報ベース(MIB)として表現される。NE装置にまたがるネットワークの管理や通信サービスに応じた効率的な管理のためには、NE装置のMIBを集約・加工した新たなビューを提供する機能が重要である。本稿では、既存のMIBに対し、様々な異なるビューを容易に提供可能とするビュー変換の実現方式を提案する。本方式では、既存のMIBと新たに提供するビューとの対応関係の記述法を新たに導入し、その記述からビュー変換のプログラムを自動生成する。

1. はじめに

TMN(電気通信管理網)^[1]の標準化が進捗し、これに基づいた通信機器の管理が普及しつつある。ここでは、ネットワーク管理のための5レイヤ(上位から順に、ビジネス管理/サービス管理/ネットワーク(NW)管理/ネットワーク要素(NE)管理/NE装置と呼ぶ)からなる機能モデルを定義している。TMNを実現する課題については、これまで、NE装置やNE管理レイヤを中心に論じられてきたが、次第に、上位のNW管理レイヤやサービス管理レイヤの実装へ移りつつある。また、管理ドメイン間の協調という観点より、顧客へ公衆網の網管理情報のアクセスや制御を可能とするCNM(Customer Network Management)^[2]の実現も課題である。

TMNでは、管理対象であるNE装置は、管理オブジェクト(MO)の集合である管理情報ベース(MIB)として表現される。MIBの定義は、同種のNE装置でもベンダ毎に異なる場合があり、NW管理レイヤでNE装置にまたがる統合的な管理を行うためには、これらのMIBを均質(MIBの定義が同一)に見せるビューが必要となる。また、NW管理レイヤ等の上位のレイヤでは、管理目的

に応じて下位のレイヤのMIBを集約・加工したビューを必要とする。さらに、CNMでも、顧客に提供するビューは、公衆網内のMIB定義とは多くの場合異なる。このため、異なる定義を持った既存のMIBに対して、様々なビューを容易に提供可能なビュー変換の実現方式が重要となる。

ビュー変換の実現方式に関して、従来の研究は、MIB実現における補足的なアーキテクチャを提案するもので、具体的な実現方法に至っていない^[3,4]。SNMP(Simple Network Management Protocol)をTMNの特定のビューに変換するもので、様々なビューに対応できるものではないため^[5,6]、前述のTMNにおける多様なビュー変換の実現には、適用できない。

本稿では、既存のMIBに対し、様々な異なるビューを提供可能とするビュー変換の実現方式を提案する。ここでは、既存のMIBと新たに提供するビューとの対応関係の記述法を新たに導入し、その記述からビュー変換のプログラムを自動生成する^[7,8]。

2. MIBのビュー変換の必要性和利用形態

2.1 ビュー変換の必要性

図1に、ビュー変換前(以下、単に、変換前と呼ぶ)のMIBのMOに対して、新たなビューを提供する場合の概念を示す。例えば、ビューの"MO₁"は、ビュー変換を用いて、変換前の"MO₁"と

Realization Method of View Conversion for Management Information Base (MIB) in TMN
Hiroki HORIUCHI, Kiyohito YOSHIHARA, Keizo SUGIYAMA and Sadao OBANA KDD R&D Labs.

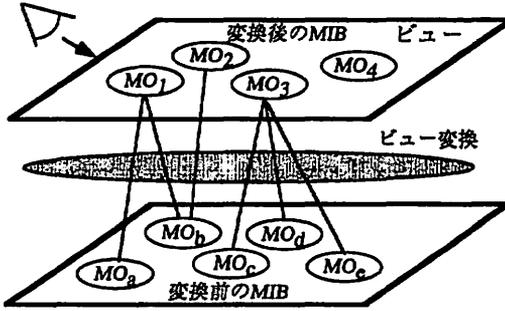


図1 ビュー変換の概念図

"MO_b"を加工して提供されることを示す。以下では、ビュー変換の必要性を示す。

(1) 同種NE装置に対する均質なビューの提供

管理するネットワークが大きくなるに従い、マルチベンダのNE装置が導入される。この際、同種のNE装置であってもベンダや標準によって、NE装置のMIB定義が異なる場合がある。例えば、ATM交換機では、標準のMIB定義^[9]と既存のベンダのMIB定義^[10]とを比較すると、使用VP(Virtual Path)の総数等のように標準のMIB定義の1属性が、ベンダのMIB定義における複数の属性で表現されたり、チャンネル識別子のように属性が同じでも、意味が異なる場合がある。また、ATM Forum^[9]やITU勧告^[11]といった標準のMIB間でも、管理オブジェクトクラス(MOC)、それに割り当てられるオブジェクト識別子(OID)等が異なる。このため、NW管理レイヤにおけるNE装置をまたがる統合的な管理には、これらのMIBが均質化された、すなわち、各NE装置のMIBの定義が同一なものとして扱えるビューを提供する必要がある。

(2) 上位レイヤの管理目的に応じたビューの提供

NW管理レイヤやサービス管理レイヤ等の上位レイヤにおける、性能や障害等の管理で扱うMIB定義は、一般に、NE装置のMIBの定義とは異なり、管理目的に応じたMIBの集約や加工を行ったビューを、それぞれのレイヤに提供する必要がある。図2にサービス管理レイヤにおけるビュー変換の例を示す。NE装置のMIB定義では、ATM Forum^[9]におけるVPの端点毎のIDや各種パラメータを持つMOC "vpCTPBidirectional"が定義され、また、その下位のMOC "upcNpcCurrentData"として、UPC(使用パラメータ制御)を違反した廃棄セル等の性能データが定義されている。一方、サービス管理レイヤのビューのMOC "vpsPerformance"

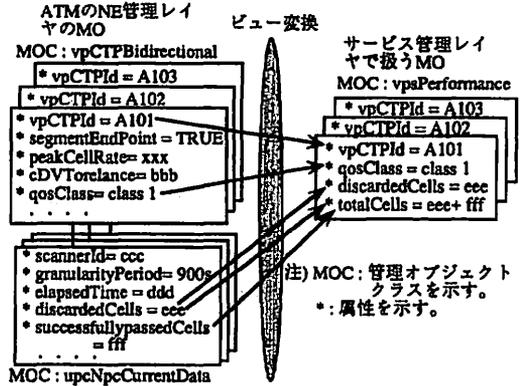


図2 サービス管理レイヤにおけるビュー変換の例

では、各VP毎の廃棄セル数を示す属性 "discardedCells"、および、総セル数を示す属性 "totalCells"等に集約・加工される必要がある。また、同様に、CNMにおいても、公衆網の顧客に提供するビューでは、NE装置のMIBに対する加工・集約を行う必要がある^[2]。

2.2 ビュー変換の利用形態

ビュー変換の機能は、図3に示すようにマネージャ、プロキシ、エージェントのいずれかに配置する利用形態がある。

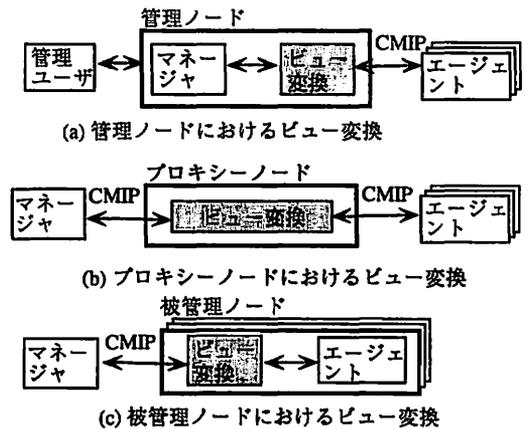


図3 ビュー変換の利用形態

3. ビュー変換の実現方式の提案

既存のMIBに対し、様々なビューを提供可能とするビュー変換の実現方式を以下に提案する。本方式では、変換前のMIB定義及びビュー変換後(以下、単に、変換後と呼ぶ)のMIB定義として、いず

れもGDMO^[10]で定義されたものを対象とする。

3.1 基本原理

ビュー変換の実現方式としては、① 変換前のMIBに対して、予め集約・加工の処理を行ったデータを保持する方法と、② 変換後のビューに対する管理操作を受信した時点で、変換前のMIBに対する属性の取得等の操作を行う方法、ならびに、③ ①と②の方法を共存させる方法の3つが考えられるが、本方式では③の方法とした。つまり、①の方法は、変換後のデータを予め保持するため、管理操作に対する応答時間が速いことを期待できるが、値が頻繁に変わるものを反映するのが困難なことや、管理操作の対象とならないものまで保持するため、メモリ容量等の無駄が多いことから、一定の期間値の変わらない、または、定期的に取得する必要がある属性に関してのみ①の方法を利用し、それ以外は②の方法を利用する。

3.2 ビュー変換方式の概要

様々なMIB定義やビューに柔軟に対応可能とするため、以下を行う。

(1) 対応関係記法の規定、導入

変換前のMIB定義と変換後のMIB定義間の対応関係を記述する方法(以下、対応関係記法と呼ぶ)を新たに規定、導入する(詳細は4章参照)。

(2) 対応関係の記述からのプログラム自動生成

ビュー変換を行うプログラム(以下、ビュー変換プログラムと呼ぶ)を、上記(1)を用いて記述した対応関係から自動生成させる(詳細は5章参照)。

3.3 ビュー変換に必要な機能

以下では、自動生成するプログラムの処理、および、それを生成するための対応記法の要求条件を明確化するため、ビュー変換に必要な機能を抽出する。

(1) MOCと属性の静的な対応付け機能

ビュー変換では、変換後MIBの各MOへの管理操作に対する処理を、変換前MIBのMOCや属性等を用いて実行する。変換前と変換後のMOと属性の対応は、1対1、N対1、1対NおよびN対Mの関係があり、それぞれ、表1に示す処理を行う。この際、変換後の属性への管理操作に対し、変換前の複数の属性に管理操作を発行したり、数値属性に対する算術演算の処理を行う。また、この際、属

性のシンタックスやそのメンバを扱うため、任意のASN.1データ型の値の代入、参照を行う。

(2) 動的な振舞いの対応付けの機能

上記(1)で示したMOCや属性の静的な対応付けだけでなく、以下に示すアクションや通知等の振舞いの対応付けを行う。

①M-SET：変換前のMOCへの値の設定のみならず、通知の発行。

②M-Action：変換前のMOCへのM-Actionの発行や属性の設定。

③M-Create/Delete：変換前のMOCの生成/削除。

④Notification：変換前のMOCからの通知に対して、通知発行。

(3) タイマを用いた定期的な属性の取得の機能

予め集約・加工の変換を行って保持しておく必要がある属性に対応するため、指定された周期で、変換前のMOの属性の取得。

4. 対応関係記法^[6]

3.3節で述べたビュー変換に必要な機能を記述可能とする、以下の記法を規定、導入する。なお、対応関係記法では、MOCに対する処理を記述し、MOIの識別名の対応付けは、識別名対応ライブラリを呼び出すこととしている(5章参照)。

4.1 概要

可読性を考慮し、記法はC言語に準じる手続き表現を採用しており、記述全体は、①定数宣言部、②MO対応付け部、③手続き部の三部分からなる。定数宣言部では、MO対応付け部で使用する整数、実

表1 ビュー変換に必要な処理

変換 前 後		MOC間	属性間
1	1	変換前MOCから、一部の属性を取り出す処理。	属性型OIDとシンタックスの変換や、定数の加算等の演算。
N	1	変換前の複数のMOCから、必要な複数の属性を集める処理。	複数属性による演算や、ASN.1構造形のメンバを各属性に割り当てる等の処理。
1	N	変換前の一MOCを、異なった集約方法で、複数MOCへ収容する処理。	一属性の構造形の要素を、それぞれ別の属性へ割り当てる等の処理。
N	M	上記1対NとN対1の場合との組み合わせ。	上記1対NとN対1の場合との組み合わせ。

数、文字列、オブジェクト識別子等の値の定数を定義する。MO対応付け部は変換後のMOCに対して発行される管理操作を変換前のMOCに発行するための規則をMOC毎に記述する。手続き部は、MO対応付け部から参照する各種手続きを記述する。

4.2 MOC対応付け部

MOC対応付け部のMOC毎の記述構成を図4に示す。なお、"[]"はオプション、"*"は0回以上の繰り返しを表す。

<MOC ラベル名>	
MO_BEGIN {	
[ATTRIBUTE規則]*	[ACTION規則]
[NOTIFICATION規則]	[CREATE規則]
[DELETE規則]	[TIMER規則]
END_MO_END	

図4 MOC毎の記述構成

ここで、"<MOCラベル>"はMOCを識別する。MOC毎の記述構成では、マネージャからの管理操作に対するビュー変換の処理を記述するため、M-GET/M-SET、M-ACTION、M-CREATE、M-DELETEの管理操作毎に、それぞれ、ATTRIBUTE規則、ACTION規則、CREATE規則、DELETE規則で記述する。また、エージェントからの通知、3.3節(3)のタイマを用いた定期的な属性の取得機能に対して、それぞれ、NOTIFICATION規則、TIMER規則を用意した。各規則の内容を表2に示す。

図5にATTRIBUTE規則を示す。"<属性ラベル>"は変換後の属性を識別する。一時変数宣言では、変換で使用する定数や変数を宣言する。また、ここでは、3.3節(3)で述べた予め集約・加工の変換を行って保持する値に対応するため、永続的に保持可能とするための変数(以下、永続変数と呼ぶ)も宣言可能とした(KEEP_VALUEにより記述)。変換前の属性に対して行なう管理操作を"GET_PROC"及び"SET_PROC"節内にそれぞれ記述する。なお、"管理操作規則"は図4に示される他の"規則"の中でも記述できる。

管理操作規則では、変換前と変換後のMOCラベル、属性ラベル、一時変数、さらに定数を組み合わせて変換前の属性に対する処理を記述

表2 各規則の内容

規則	内容
ATTRIBUTE規則	変換後の属性に対するM-GETまたはM-SETを変換前の属性に行なう管理操作。属性毎に記述。
ACTION規則	変換後のMOCに対するM-ACTIONを変換前のMOCに行なう管理操作。
NOTIFICATION規則	変換前のMOCからのM-EVENT-REPORTを変換後のMOCに発行する管理操作。
CREATE規則	変換後のMOCに対するM-CREATEを変換前のMOCに行ない、識別名手続きに変換前のMOCのMOIの生成を要求する管理操作。
DELETE規則	変換後のMOCに対するM-DELETEを変換前のMOCに行ない、識別名手続きに変換前のMOCのMOIの削除を要求する管理操作。
TIMER規則	指定された周期で、変換前の属性の値を取得する管理操作。

<属性ラベル> {
[-一時変数宣言]*
[GET_PROC { [管理操作規則]* } END_SET_PROC]
[SET_PROC { [管理操作規則]* } END_GET_PROC]
};

図5 ATTRIBUTE規則

する。ここでは、属性値やASN.1の構造形のメンバーの値の取得及び設定に加え、関数や手続きの呼び出し、算術演算(加減乗除、型変換等)、制御文を記述できる。また、GDMO中で使用されるASN.1の型や、ASN.1標準中の基本形は、C言語等で使用される型と同等に記述できる。

表3に、管理操作規則内で使用可能な主な関数の一覧を示す。これらの関数は、対応関係記述とMOIの識別名を対応づける際に使用する関数、ならびに、通知、生成/削除のCMIPプリミティブを発行するための関数である。また、制御文では、C言語と同様に条件文(IF THEN)、選択文(SWITCH CASE)、繰り返し文(WHILE)等を記述できる。

4.3 対応関係記法を用いた記述例

ベンダ(Fore社)独自のMIB定義(以下Fと呼ぶ)をもつATM交換機を、標準的なMIB定義であるM4インタフェース(以下M4と呼ぶ)のビューに対応させる対応関係記述を、構成情報、性能情報、通知の対応付けの点から記述する。

(1) 構成情報の対応付け

図6にM4の"atmAccessProfile" MOCの属性

表3 管理操作規則における主な関数

関数名	機能等
GET_INDEX GET_INDEX_NEXT	変換前または後のMOCを引数とし、対応するMOI全てを抽出し、その一つを順に返す。
GET_SUPERIOR_FDN GET_SUPERIOR_CLASS	名前木上で上位のMOIの識別名やMOCを返す。
SELF_FDN SELF_MOC	ビュー変換を実行しているMOI自体の識別名やMOCを返す。
FDN_COMPARE	引数で指定された2つの識別名を比較する。
CORRESPOND	変換前MOIを引数として、対応する変換後MOIを返す。
emit_Notification emit_Create emit_Delete	マネージャへの通知発行、エージェントへのMOI生成/削除操作発行。
CREATE_M_INSTANCE DELETE_M_INSTANCE	変換後MOIの生成/削除

"maxNumActiveVCCsAllowed"の対応付けの一部を示す。この属性は一つのポートのVCC(Virtual Channel Connection)の最大数を表す。Fには一つのVCCの最大個数を表す"pathEntry" MOCの"pathMax-Channels"属性がある。一つのポートはいくつかのパスを収容し、一つのパスはいくつかのチャネルを収容する。これより"atmAccessProfile" MOCの一つのMOIは"pathEntry" MOCの複数のMOIに対応する。例は、a)対応するFの複数のMOIを探し(12行目から22行目)、b)そのMOI毎に属性値を取得し(15行目から18行目)、c)これらの総和を求める(19行目)ことを表す。なお、変換前の属性ラベルおよび変換後の属性ラベルは、他の属性との識別や属性のシンタックスのASN.1構造形のメンバへの操作が可能のように、"<MOCラベル名>%<属性名>[.ASN.1メンバ]*"、"%<MOCラベル名>%<属性名>[.ASN.1メンバ]*"と記述する。

上記a)で1対NのMOIの対応付けを行なうため、GET_INDEX(12行目)とGET_INDEX_NEXT(20行目)を使用している。また、N対1のMOIの対応付けをするために、記号"#"(15及び18行目)を使用する。"#"の左には属性ラベル、右にはMOIを置く。これは左の属性が右のMOIに束縛されることを表す。1対Nに対応する属性の対応付け例として、一数值属性を複数の数值属性と対応付ける(19行目)。対応する変換前の属性値の取得のため"<<"(18行目)を使う。これはM-GETに対する管理操作を変換前の属性に行ない、取得した右辺の値を左辺に代入する。

```

1 atmAccessProfile MO_BEGIN {
2   atmAccessProfileId { /* 省略 */ }
3   maxNumActiveVCCsAllowed { /* 省略 */ }
4   maxNumActiveVCCsAllowed { /* 省略 */ }
5   DECLARE { /* 一時変数宣言 */
6     fdn (FDN);
7     port (INTEGER);
8     id (INTEGER);
9   }
10  END_DECLARE
11  GET_PROC {
12    %maxNumActiveVCCsAllowed = 0;
13    fdn = GET_INDEX( pathEntry );
14    WHILE( fdn != NULL ) DO {
15      id = $atmAccessProfile$atmAccessProfileId.numericName;
16      port = GET ( pathEntry$pathEntryId.pathPort$fdn );
17      IF( port == id )
18        THEN ( %maxNumActiveVCCsAllowed
19              <<- pathEntry$pathMaxChannels#fdn
20              + %maxNumActiveVCCsAllowed;
21              fdn = GET_INDEX_NEXT( pathEntry );
22            )
23      }END_IF }END_WHILE
24  }END_GET_PROC
25  SET_PROC { /* 省略 */ }END_SET_PROC ;
26  } MO_END

```

図6 構成情報の対応付け例

```

1 cellHeaderProtocolCurrentData MO_BEGIN
2 { administrativeState { /* 省略 */ }
3   discardedCellsInvalidHeader {
4     DECLARE { tmp (INTEGER); }END_DECLARE
5   }
6   GET_PROC {
7     tmp <- atmLayerEntry$atmReceivedCells - aa4Entry$aa4ReceivedCells
8     - aa5Entry$aa5ReceivedCells + aa4Entry$TransmittedCells
9     + aa5Entry$TransmittedCells - atmLayerEntry$atmTransmittedCells ;
10    %discardedCellsInvalidHeader = tmp - discardedCellsInvalidHeader_former ;
11  }END_GET_PROC ;
12  TIMER_PROC INTERVAL = 15M ;
13  DECLARE {
14    KEEP_VALUE discardedCellsInvalidHeader_former (INTEGER) = 0;
15    discardedCellsInvalidHeader_latter (INTEGER);
16    discardedCellsInvalidHeader_differ (INTEGER); /* 省略 */
17  }END_DECLARE
18  discardedCellsInvalidHeader_latter
19  <- atmLayerEntry$atmReceivedCells - aa4Entry$aa4ReceivedCells
20  - aa5Entry$aa5ReceivedCells + aa4Entry$TransmittedCells
21  + aa5Entry$TransmittedCells - atmLayerEntry$atmTransmittedCells ;
22  discardedCellsInvalidHeader_former = discardedCellsInvalidHeader_latter ;
23  /* 新旧値の入れ替え */ /* 省略 */
24  }END_TIMER_PROC /* 省略 */
25  }END_MO_PROC

```

図7 性能情報の対応付け例

(2) 性能情報の対応付け

図7にM4の"cellLevelProtocolCurrentData" MOCの属性"discardedCellsInvalidHeader"の対応付けの一部を示す。この属性は、一つのポートにおける不正なヘッダのため廃棄したセル数の15分毎の積算値を示す。Fでは、不正ヘッダによる廃棄セル数は、交換機起動時からの積算値を表わすとともに、送受別々に積算している。このため、Fの6属性から送受の廃棄セルの積算値を取得し(6行目から8行目)、前の15分毎の性能データ(以下、事前データと呼ぶ)からの差分を計算してM4の属性値を得る(9行目)。事前データは、永続変数として保持され(13行目)、TIMER規則により15分毎(11行目)に、17行目から21

```

1 tcAdaptorTTPBidirectional MO_BEGIN
2 /* 省略 */
3 NTF_PROC {communicationsAlarm: RELATED_TO
4           InternetAlarm : Internet;
5 }END_NTF_PROC /* 省略 */
6 }MO_END /* 省略 */
7 Internet ( in InternetAlarmInfo ) {
8   DECLARE ( out (AlarmInfo);
9   }END_DECLARE
10 IF ( in.probableCause == {1 3 6 1 4 1 3 2 6 2 2 0 0} )
11 THEN ( /* asxSwLinkDownwon受信 */
12   out.probableCause = {arfProbableCause 27};
13   out.perceivedSeverity = @major; /* 省略 */
14   emit_Notification( communicationsAlarm, KEEP, out );
15 }END_IF
16 };

```

図8 通知の対応付け例

行目の処理により更新される。

(3) 通知の対応付け

図8にM4の"tcAdaptorTTPBidirectional" MOCから発行する通知の規則の一部を示す。3行目から4行目のNOTIFICATION規則では、Fからの"InternetAlarm"通知に対して、M4の"communicationsAlarm"通知を発行し、その際の手続きはInternetであることを示す。"InternetAlarm"のパラメータである"probableCause"のパラメータが条件に合致すれば(10行目)、"communicationsAlarm"のパラメータを設定し(12, 13行目)、通知を発行する(14行目)。

5. ビュー変換プログラム自動生成

ビュー変換生成コンパイラは、GDMOと対応関係記述からビュー変換プログラムとクラス対応情報(5.2節参照)を自動生成する(図9)。また、識別名対応ライブラリは、クラス対応情報を使用して、変換前MOIと変換後MOIとの関係を示す識別名対応情報を生成する(詳細は5.2節参照)。

5.1 ビュー変換プログラム

ビュー変換プログラムは、図10に示すように、①制御モジュール、②タイマモジュール、③インスタンス管理モジュール、複数の④変換処理モジュールから構成する。また、変換後のMIBへの操作や、変換前のMIBへの操作は共有メモリを介して、各モジュールに渡される。

(1)制御モジュール

制御モジュールは、M-GET要求等の変換後のMIBへの操作、変換前のMIBからの通知、ならびに、タイマモジュールからのタイムアウトを受信

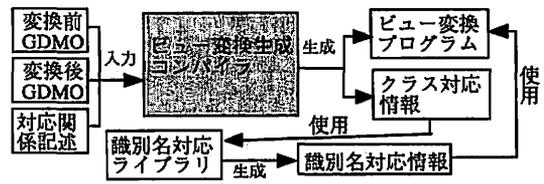


図9 ビュー変換生成コンパイラ

すると、変換処理モジュールへ振り分ける。

(2)タイマモジュール

4.2節のTIMER規則で記述された間隔で、複数のタイマを起動し、MOIの識別名を付加してタイムアウトを制御モジュールへ知らせる。

(3)インスタンス管理モジュール

インスタンス管理モジュールは、変換前と変換後のMOIの対応を示す識別名対応情報を一括して管理する。これらの情報は、変換処理モジュールならびにタイマモジュールから参照される。

(4)変換処理モジュール

ビュー変換の処理を行うモジュールであり、制御モジュールから受信した操作が、変換後のMIBからの操作、既存MIBからの通知、あるいは、タイムアウトかにより、以下の処理を行う。

- ①変換後のMIBからの操作: 操作がM-GETまたはM-SETの場合には、変換後MOCの判別、属性の種類判別、識別名情報の取得の後に該当するATTRIBUTE規則を実行する。操作がM-CREATE, M-DELETE, M-ACTIONの場合には、変換後MOCの判別、識別名情報の取得の後に該当する規則を実行する。
- ②変換前のMIBからの通知: 変換前MOCの判別、識別名情報の取得の後に該当するNOTIFICATION

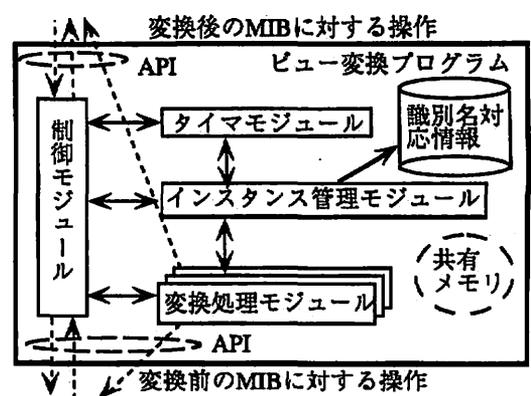


図10 ビュー変換プログラムの構成

規則を実行する。

③ タイムアウト: 変換後MOCの判別、識別名情報の取得の後に該当するTIMER規則を実行する。

5.2 インスタンスにおける識別名の対応付け

識別名対応情報は以下の手順で生成する。

(1) ビュー変換に必要な変換前MIBのMOC抽出
ビュー変換コンパイラは、対応関係記述から、クラス対応情報として、①変換後MIBの全MOCに対して、対応付ける変換前MOC、ならびに、②変換後MOCの名前属性に対応する変換前MOCの属性を抽出する)。

(2) 変換前MOIの識別名と属性値の収集

上記(1)で抽出した変換前MOCに属する全インスタンスを求め、それらに対する①識別名及び②他の必要な属性値を求める。

(3) MOIの対応付け

上記(2)で取得した変換前MOIの識別名と属性値を、上記(1)のクラス対応情報を利用して対応付けて、識別名対応情報を得る。この際、変換後MOIの一つに対して、複数の候補が存在する場合は、キーとなる属性の値を指定して特定するか、ユーザがマニュアルで指定する。

6. 評価と考察

6.1 対応関係記法の記述力

4.3節で示したATM交換機に対応関係記述を適用した結果、M4インタフェースで定義されるATM特有の27種類のMOC全てを記述できた。但し、ATM特有の属性54種類のうち、属性cellHeaderAbnormalityType等のように、ベンダ独自のMIBに対応するものがなく、任意の値を設定するのが困難な属性が7種類あった。また、属性maxEgressBandwidth等のように、M4で属性値の変更が許されるが、Fでは許されておらず、M-GETへの対応規則は記述できるが、M-SETへの処理が対応付けられない属性が7種類あった。

以上、サポートできない属性が若干あったが、MOCレベルでは全て対応付けられ、ATM交換機を管理する上では、特に問題無いと考

えられる。

6.2 対応つかない属性への対処

変換前のMOIでサポートされていない属性は、新たなビューを提供する際に、変換する方法がないので対応できない。また、逆に、変換後のビューで、集約や加工では変換前のMOから導出できない、サービス管理やCNMにおける顧客情報等のようなMOもある。この場合には、対応関係中で、値を設定する手続きを用意したり、変換後MOIのためのMIB用データベースに別途登録し、これと連携することにより、対応可能と考えられる。

6.3 ビュー変換生成コンパイラ

様々なビューに対応するため、本方式では、対応関係記法に加え、コンパイラ形式を採用した。コンパイラ形式の他にインタプリタ形式も考えられるが、本ビュー変換では様々なビューを提供するため、既存MIBへの複数の管理操作の発行や制御文の記述等の複雑な処理が必要な場合も多く、インタプリタ形式より高速な処理が可能なコンパイラ方式を採用した。

6.4 トランザクション処理への対応

ビュー変換後の一つの属性に対して、変換前の複数の属性が対応付けられることがある。このため、属性の値の設定等において、複数の属性の値の変更の一貫性を保証する必要がある。このため、対応関係記述では、トランザクション処理の開始と終了を明示的に記述可能とした。これに対するビュー変換プログラムの実現では、エージェントがどの程度トランザクション処理をサポートしているかにより、実現方法が異なる。例えば、複数のMOIを含む単一の管理操作におけるトランザクションは、CMIPの"synchronization"パラメータの"atomic"の値をサポートしていれば、それに対応付けられる。サポートしていない場合には、ビュー変換プログラムで、全てのM-SETが成功しない時には、事前の値に設定し直す等の処理を行う必要がある。

7. おわりに

本稿では、既存のMIBに対し、様々な異なるビューを容易に提供するためのビュー変換の実現方法を提案した。本方式では、既存のMIBと新たに提供するビューとの対応関係を記述する対応関係記法を新たに導入し、その対応関係記述からビュー変換

のプログラムを自動生成した。対応関係記法は、MOCや属性の静的な対応付けだけでなく、動的な振る舞いやタイマを用いた定期的な属性の取得を記述可能とする。また、対応関係記法をATM装置の対応関係記述に適用し、その有効性を確認した。現在、本方式に基づく実装を行っている。

最後に日頃ご指導頂くKDD研究所 村上所長、ならびに、ご討論いただいた(株)KDDコミュニケーションズ 黒木担当課長に感謝します。

参考文献

- [1]: ITU-T Rec. M.3010 "Principles for Telecommunications Management Network", 1992.
- [2]: ITU-T Rec. X.160 "Architecture for Customer Network Management Service for Public Data Networks", 1995.
- [3]: S. M. Klerer, R. S. Cohen "Distribution of Managed Object Fragments and Managed Object Replication : The Data Distribution View of Management Information", Integrated Network Management II, IFIP, 1991.
- [4]: J. R. Haritsa, M. O. Ball, N. Roussopoulos, A Datta, J. S. Baras, "MANDATE: Managing Networks Using DAtabase TEchnology", IEEE JSAC, Dec. 1993.
- [5]: 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.
- [6]: 櫻村他, "管理情報転送サービス統合機構の検討", 信学技報 IN94-113, Nov. 1994.
- [7]: 堀内, 黒木, 吉原, 杉山, 小花 "TMNにおける管理情報ベース(MIB)におけるビュー変換方式の提案", 情処第53回全大, 10-06, Sept. 1996.
- [8]: 吉原, 堀内, 黒木, 杉山, 小花 "TMNにおける管理情報ベース(MIB)のビュー変換のための対応関係記法", 情処第53回全大, 10-07, Sept. 1996.
- [9]: ATM Forum af-nm-0027, "CMIP Specification for the M4 Interface, Sep. 1995.
- [10]: Fore Systems "Fore System MIB", 1994.
- [11]: ITU-T Rec. I.732 "ATM Management of the network elementview", Jan. 1996.
- [12]: ITU-T Rec. X.722 "Guidelines for the Definition of Managed Objects", 1992.