ネットワーク管理のための管理情報ベース（MIB）に対する柔軟なビュー提供方式

堀内浩規† 吉原貴仁† 杉山敬三†
小花貞夫† 鈴木健二†

電気通信管理網（TMN）において管理の対象となるネットワーク要素（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．ネットワーク管理システムの構築では，NE装置のMIBを集約・加工し，実際のMIBとは異なった見え方を与えるビューを提供する機能の実現が重要である．本論文では，既存のMIBに対し，様々なビューを提供するプログラムを効率的に開発可能とするビュー提供方式を提案する．提案方式は，既存のMIBと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を提供するために必要なプログラムを自動生成することを特徴とする．対応関係の記述法では，1）管理操作受信時に集約・加工を行う静的な情報の対応付け機能，2）動的な情報の対応付け機能や，3）周期的に集約・加工を行う機能を記述可能とする．提案方式を実装し，ATM交換機のビュー提供に適用して，自動生成したビューを提供するプログラムの処理時間やプログラム開発効率等の観点から提案方式の有効性を実証した．

Realization Method for Providing Flexible View of Management Information Base (MIB) in Network Management

Hiroki Horiuchi,† Kiyohito Yoshihara,† Keizo Sugiyama,†
Sadao Obana† and Kenji Suzuki†

In Telecommunications Management Network (TMN), managed Network Elements (NEs) are represented as Management Information Base (MIB) which is a collection of managed objects. In network management systems, it is necessary to provide views which aggregate and process existing MIBs, suitable for management purposes. This paper proposes a realization method for providing flexible views of MIBs. In the method, notation of mapping rules between an existing MIB and a view is introduced and programs for providing view are generated from GDMO definitions of a MIB and the mapping rules described. Furthermore, we implemented a proxy based on the method and evaluated its performance and efficiency of program development.

1. はじめに

電気通信管理網（TMN）[1]の標準化が進捗し，これに基づくネットワークの管理が普及しつつある．TMN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の実現のための課題は，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現へと移りつつある．また，管理ドメイン間の協調という観点より，顧客から公衆網の網管理情報への監視や制御を可能とする顧客網管理（CNM）[2]の実現も課題である．

TMNの管理対象である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装置でもベンダごと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）に見せる必要がある．この場合には，NE装置のMIBの定義を揃えるため，実際のMIBの定義とは異なった見え方を与えるビューが必要となる．ビューは，NW管理レイヤ等の上位のレイヤにおける，管理目的に応じた下位のレイヤのMIBを集約・加工する際にも必要となる．さらに，CNMでも，顧客に提供するビューは公衆網内のMIBの定義とは多くの場合異なる．この

† 国際電信電話株式会社研究所
KDD R&D Laboratories

ため，ネットワーク管理システムの構築では，既存のMIBに対してビューを提供する機能の実現が重要となる．

ビューの提供に関する従来の研究では，ビュー提供の必要性やアーキテクチャを提案したものがある[3),4)]が，具体的な実現方法については示されていない．また，SNMP（Simple Network Management Protocol）を実装したNE装置のMIBにTMNのビューを提供するための方式が提案されている[5),6)]が，提供するビューやNE装置が限定されており，前述のビューの提供の目的には適用できない．

本論文では，既存のMIBに対し，様々なビューを提供するプログラムを効率的に開発可能とするためのビュー提供方式を提案する．ここでは，既存のMIBと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を提供するためのプログラムを自動生成する．さらに，この方式を実装し，生成プログラムの処理時間やプログラムの開発効率等の観点からの評価を行う．以下，2章でMIBの概要とビュー提供の必要性を述べる．3章でビュー提供方式の提案を行い，4章と5章で，それぞれ，対応関係の記述法とビューを提供するプログラムの自動生成について述べる．6章では実装と評価を通じて，提案方式の有効性を示す．

2. MIBの概要とMIBにおけるビュー提供の必要性

2.1 MIBの概要

TMNにおけるNE装置等の管理対象は，管理オブジェクト（MO）の集合であるMIBとして表現される．同じ性質を持つMOの種類のことをMOクラスと呼び，MOクラスに属するMOの実体のことをMOインスタンスと呼ぶ．MIBに含まれるMOクラスは，属性型，許される管理操作，振舞い，ならびに，MOクラス間の包含関係を示す名前結合等がGDMO（Guidelines for the Definition of Managed Objects）と呼ばれる記法を用いて定義される（以下，このようなMIBの定義をMIB定義と呼ぶ）．

MIBでは，MOインスタンスの集合を名前木と呼ばれる木構造として保持し，各MOインスタンスは識別名により一意に識別される．また，MIBに対する管理操作には，属性値の取得（M-Get）と設定（M-Set），MOインスタンスの生成（M-Create）と削除（M-Delete），動作（M-Action），ならびに，取得操作の取消し（M-CancelGet）とMIBから発行する通知（M-EventReport）がある．

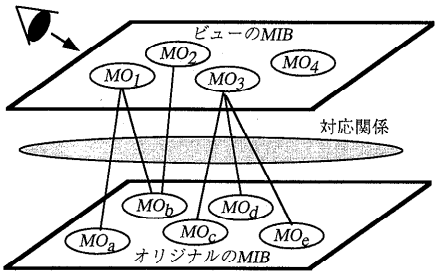


図1 ビューの概念
Fig. 1 View of MIB.

2.2 MIBにおけるビュー提供の必要性

図1に，ビューの元となる既存のMIB（以下，オリジナルのMIBと呼ぶ）に対する異なった見え方，すなわちビューを提供する概念を示す．たとえば，ビューのMIBの“MO_1”は，オリジナルのMIBの“MO_a”と“MO_b”を加工したものであることを示す．以下では，ビュー提供の必要性を述べる[3),4),7),8)]．なお，ビュー提供の実現形態には，マネージャまたはエージェントの装置に組み込む形態や，マネージャとエージェントの間に外付けの装置（プロキシー）として実現する形態がある．

(1) 同種NE装置を均質に見せるビューの提供

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MIB定義が異なる場合がある．たとえば，ATM交換機では，ATM Forum標準のMIB定義[9)]とベンダ独自の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定義における1つの属性型が，ベンダのMIB定義における複数の属性型で表現されたり，チャネル識別子のように，それぞれのMIB定義で属性型の名前やシンタックスが同じでも，意味が異なる場合がある．また，ATM ForumやITU勧告[11)]といった標準のMIB定義間でも，MOクラスや，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が均質化された，すなわち，各NE装置のMIB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定義とは異なり，管理目的に応じてMIBの集約や加工を行ったビューを提供する必要がある．図2にNW管理レイヤにおけるビュー

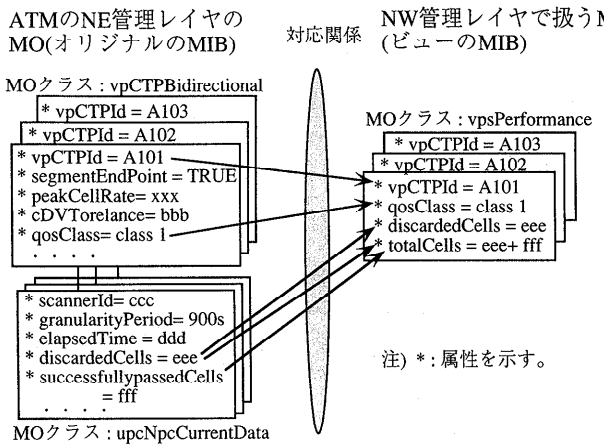


図 2 NW 管理レイヤにおけるビュー提供の例
Fig. 2 Example of view in network management layer.

提供の例を示す．ここでは，ATM Forum の MIB 定義[9]をオリジナルの MIB 定義としている．MO クラス “vpCTPBidirectional” では，VP の端点ごとの ID や各種パラメータを持つ VP の構成情報が定義され，その下位の MO クラス “upcNpcCurrentData” では，使用パラメータ制御（UPC）を違反した廃棄セル等の性能情報が定義されている．一方，NW 管理レイヤで提供するビューの MIB 定義における MO クラス “vpsPerformance” では，各 VP ごとの総セル数を示す属性型 “totalCells” 等に集約・加工する必要がある．また，図 2 の NW 管理レイヤのビューをオリジナルの MIB 定義として，さらに上位のサービス管理レイヤのビューを提供する必要がある場合がある．同様に，顧客網管理（CNM）[2]においても，公衆網の顧客に提供するビューでは，公衆網内の NE 装置や上位レイヤの MIB に対して集約・加工を行う必要がある．

以上，(1)，(2) で示したように，ネットワーク管理システムの構築では，既存の MIB に対して異なる見せ方を与える様々なビューの提供が必要となる．

3. 柔軟なビュー提供方式の提案

2 章で述べた様々なビューを提供するために，ビューごとに個別にプログラムを開発するのは効率的でない．そこで，オリジナルの MIB に対し，様々なビューを提供するプログラムを効率的に開発可能とするビュー提供方式を以下に提案する．ここでは，オリジナルの MIB 定義およびビューの MIB 定義は，いずれも GDMO[12] で定義されたものとする．

3.1 基本原理

ビュー提供方式としては，①オリジナルの MIB の情報に対して，あらかじめ集約・加工の処理を行ってビューの MIB の情報として保持し，ビューの MIB に対する管理操作は保持した情報に対して実行する方法と，②ビューの MIB に対する管理操作を受信した時点で，オリジナルの MIB に対する管理操作を実行する方法が考えられる．①の方法は，管理操作の応答時間が速いことを期待できる反面，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，つねに保持しなければならないという問題点がある．そこで，提案方式では，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情報に関してのみ①の方法を利用し，それ以外は②の方法を利用する方法とした．

3.2 ビュー提供方式の概要

オリジナルの MIB 定義に対し，様々なビューを提供するプログラムを効率的に開発可能とするため，以下を行う．

(1) 対応関係記法の導入
オリジナルの MIB 定義とビューの 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
ビューを提供するプログラム（以下，ビュー提供プログラムと呼ぶ）を，上記 (1) を用いて記述した対応関係を利用して自動生成させる（詳細は 5 章参照）．

3.3 ビュー提供プログラムの機能

以下では，ビュー提供プログラムを生成するのに必要となる対応関係記法の要求条件を明確にするため，

表 1 静的な情報の対応付けのための処理
Table 1 Processing for mapping of static information.

対応		MO間	属性間
ビュー	オリジナル		
1	1	オリジナルのMOから,一部の属性を取得する.	属性型OIDとシンタックスの変換や,定数の加算等の演算,ASN.1の符号化／復号等を行う.
1	N	オリジナルの複数のMOから,必要な属性を取得する.	オリジナルの複数の属性に対する演算や,ASN.1構造形メンバのオリジナルの各属性への割り当て,ASN.1の符号化／復号等を行う.
N	1	オリジナルの一つのMOを,異なった集約方法で,複数MOへ割り当て,その値を取得する.	オリジナルの属性を,ビューの複数の属性へ割り当て,ASN.1の符号化／復号等を行う.
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.
1	0	オリジナルのMOに対応付かない属性を保持し,その値を取得する.	オリジナルの属性以外から値の導出を行う.

注）オリジナル：オリジナルのMIBを示す.

表 2 動的な情報の対応付けのための処理
Table 2 Processing for mapping of dynamic information.

管理操作と通知	処理内容
属性値設定操作 (M-Set操作)	①ビューとオリジナルの属性値設定(注1),② 値変更を通知するためのビューからの通知発行の処理等.
MOインスタンス生成操作 (M-Create操作)	①ビューとオリジナルのMOインスタンスの生成(注2),②対応するオリジナルの属性値の設定(注3),③ 生成を通知するためのビューからの通知発行の処理等.
MOインスタンス削除操作 (M-Delete操作	①ビューとオリジナルのMOインスタンスの削除(注2),②対応するオリジナルの属性値の設定(注3),③ 削除を通知するためのビューからの通知発行の処理等.
動作操作 (M-Action操作)	動作の内容に合わせて,①オリジナルのMOへのM-Action発行(注2),②オリジナルの属性値設定,③ビューの属性値設定,④ビューのMOインスタンスの生成および削除,⑤オリジナルのMOインスタンスの生成および削除の処理等.
通知 (M-EventReport通知)	①ビューからの通知発行(注2),②ビューの属性値設定の処理等.

(注1) 表1のMO間と属性間の両方の処理に対応する.
(注2) 表1のMO間の処理に対応する.
(注3) 表1の属性間の処理に対応する.
(注4) ビューとオリジナルは,それぞれ,ビューとオリジナルのMIBを示す.

まず，ビュー提供プログラの機能を抽出する．

(1) 管理操作受信時に集約・加工を行う機能

(a) 静的な情報の対応付け機能

属性値取得操作（M-Get 操作）受信に伴った管理操作の実行では，ビューまたはオリジナルの MIB への副次的な管理操作発行等の波及効果が生じない静的な情報の対応付けとなる．この際のビューとオリジナルの MIB における MO 間と属性間の対応は，1 対 1，1 対 N，N 対 1，N 対 M，および，1 対 0 の関係があり，**表 1** に示す処理を行う．

(b) 動的な情報の対応付け機能

M-Get 操作以外の属性値設定（M-Set 操作）等の管理操作や通知（M-EventReport 通知）受信時には，ビューまたはオリジナルの MIB に波及効果を与える動的な情報の対応付けとなる（**表 2**）．ここでは，まず，上記 (a) の静的な情報の対応付けの場合と同様

に，ビューとオリジナルのMIBにおけるMO間と属性間の対応は，1対1，1対N，N対1，N対M，および，1対0の関係を持って，表1の処理を対象となる管理操作に置き換えて実行する．この際，M-Set操作はMO間と属性間の両方，MOインスタンス生成操作（M-Create操作）と削除操作（M-Delete操作）はMO間または属性間のいずれか一方，動作操作（M-Action操作）とM-EventReport通知はMO間のみの関係を持った処理を行う．さらに，上記に加え，M-Set操作における属性値変更にともなう通知の発行等の波及効果の処理を行う．

(2) 周期的に集約・加工を行う機能

ビューのMIBに対する管理操作の受信に関係なく，指定した周期でオリジナルのMIBの属性値を取得し，その属性値を集約・加工してビューのMIBの情報として保持する．

4. 対応関係記法

4.1 要求条件の明確化

3.3節のビュー提供プログラムの機能を記述するための対応関係記法の要求条件を以下に示す．

[条件1] MOクラスと属性型ごとの記述

MOインスタンスとそれに包含される属性値の個数は一般に膨大であり，MOインスタンスごとに対応関係を記述するのは現実的でない．このため，本記法ではMOクラスと属性型ごとでビューとオリジナルのMIBの対応を記述する．ビューとオリジナルのMIBのMOインスタンスごとの対応は，別の処理により識別名対応情報（5章参照）として求める．

[条件2] 管理操作と通知の種類ごとの記述

MOクラスごとの記述中では，3.3節(1)管理操作受信時に集約・加工を行う機能を記述するため，M-Get操作とM-Set操作はMOクラスが包含する属性型ごと，M-Action操作はMOクラスに定義される動作の種類ごと，M-Create操作とM-Delete操作はMOクラスごと，M-EventReport通知はMOクラスに定義される通知の種類ごとに記述する．

[条件3] 静的な情報の対応付けの記述

3.3節(1)(a)の機能を記述可能とするため，条件2で示した記述中で，MOクラス名と属性型名を使用し，属性型名を被演算子とする算術演算の記法を導入して対応付ける．

[条件4] 動的な情報の対応付けの記述

3.3節(1)(b)の機能を記述するため，条件3に加え，ビューとオリジナルのMIBに対する管理操作発行やビューのMIBからの通知発行を記述可能とする．この際，ビューのMIBに対する管理操作やオリジナルのMIBからの通知に含まれるパラメータから管理操作の対象となるビューのMOインスタンスの識別名や設定する値等を抽出する処理を記述可能とする．

[条件5] 周期的に起動される処理の記述

3.3節(2)の機能を記述するため，周期的にオリジナルのMIBの属性値を取得する処理を記述可能とする．

[条件6] ASN.1データ型の記述

条件3~5の算術演算において，属性のシンタックスを扱えるようにするため，任意のASN.1データ型に対する変数の宣言，値の代入ならびに参照を可能とする．

[条件7] 永続変数の記述

条件4のビューのMIBへのM-Set操作等の処理結果や，条件5の周期的に取得した属性値をビューのMIBの情報として保持するため，永続的に保存する変数（以下，永続変数と呼ぶ）を宣言可能とする．

4.2 概　　要

対応関係記法は，①定数宣言部，②MOクラス対応付け部，③手続き部の3部分からなる．定数宣言部では，MOクラス対応付け部で使用する整数，実数，文字列，OID等の値の定数を定義する．MOクラス対応付け部は，ビューのMIBに対して発行された管理操作をオリジナルのMIBに対する管理操作に変換するための規則をMOクラス単位に記述する（条件1）．手続き部は，MOクラス対応付け部から参照する各種手続きを記述する．

4.3 MOクラス対応付け部

MOクラス対応付け部の文法を**図3**(a)に示す．MOクラスごとにビューのMIBへの管理操作や通知の種類に応じて，Attribute規則，Action規則，Create規則，Delete規則，Notification規則を記述し（条件2），周期的に起動される処理をTimer規則で記述する（条件5）．

図3(b)にAttribute規則の文法を示す．一時変数宣言では，定数や変数に加え，永続変数（条件7）を宣言する（予約語KEEP_VALUEにより宣言）．ビューのMIBの属性型に対する管理操作を“GET_PROC”および“SET_PROC”節内にそれぞれ記述する．なお，“管理操作規則”は図3(a)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，条件3～7に示す処理に加え，型変換，関数や手続きの呼び出し，固定値の割当て，および，制御文を記述できる．オリジナルとビューにおけるMIBのMOクラスと属性型は，他のMOクラスと属性型との識別，算術演算，属性型のシンタッ

表 3 管理操作規則における主な組込み関数
Table 3 Built-in functions in management operation rules.

関数名	機能等
GET_INDEX () GET_INDEX_NEXT ()	オリジナルまたはビューのMOクラスを引数とし,対応するMOインスタンス全てを抽出し,その一つを順に返す.
GET_SUPERIOR_FDN () GET_SUPERIOR_CLASS ()	オリジナルまたはビューの名前木上で1つ上位のMOインスタンスの識別名やMOクラスを返す.
SELF_FDN () SELF_MOC ()	処理を実行しているビューのMOインスタンス自体の識別名やMOクラスを返す.
FDN_COMPARE ()	引数で指定された2つの識別名を比較する.
CORRESPOND ()	オリジナル(ビュー)のMOインスタンスとビュー(オリジナル)のMOクラスを引数として,対応するビュー(オリジナル)のMOインスタンスを返す.
emit_Notification () emit_Create () emit_Delete () emit_Action ()	通知発行,オリジナルのMOインスタンスの生成および削除操作,M-Action操作の発行.
CREATE_VIEW_INSTANCE () DELETE_VIEW_INSTANCE ()	ビューのMOインスタンスの生成および削除を行う.
CREATE_ORG_SET () DELETE_ORG_SET ()	ビューのMOインスタンスの生成および削除に伴って,オリジナルの属性値を設定する.
SYSID_SET ()	複数のビューのMIBを集約・加工する際に,対象とするMIBを特定する.

(注) ビューとオリジナルは,それぞれ,ビューとオリジナルのMIBを示す.

```
<MOクラス対応付け部> ::=
  <MOクラス名>
    MO_BEGIN    {
      [ <Attribute規則> ]*   [ <Action規則> ]
      [ <Notification規則> ]   [<Create 規則> ]
      [ <Delete規則> ]        [ <Timer規則> ]
    }MO_END
```

(a) MOクラス対応付け部の文法
(a) Syntax for MO class mapping part.

```
<Attribute規則> ::=
  <属性型名>   {
    [ <一時変数宣言> ]*
    [ GET_PROC { [ <管理操作規則> ]* } END_GET_PROC]
    [ SET_PROC  { [ <管理操作規則> ]* } END_SET_PROC]
  };
```

(b) Attribute規則の文法
(b) Syntax for Attribute rule.

```
<オリジナルのMIBの属性> ::=
  <MOクラス名>%<属性型名>[ . <ASN.1メンバ> ]*
<ビューのMIBの属性> ::=
  [$<MOクラス名> ]%<属性型名>[ . <ASN.1メンバ> ]*
```

(c) 管理操作規則中の属性の文法
(c) Syntax for attribute in operation mapping rule.

注) <>：非終端記号を表わす, []：オプションを表わす,
*：0回以上の繰り返しを表す.

図 3 対応関係記法の文法の一部
Fig. 3 Syntax for mapping rules.

クスの ASN.1 構造形メンバへの操作が可能なように,図 3 (c) に示す文法で記述する（条件 3，条件 6）.

表 3 に，管理操作規則内で使用可能な主な組込み関数の一覧を示す．これらの関数は，ビューとオリジナルの MO インスタンスの識別名を求める関数，オリジナルの MIB に対する MO インスタンス生成/削除等の管理操作を発行する関数（条件 4），ならびに，複数のオリジナルの MIB を集約・加工する際に対象の MIB を特定する関数等である．属性値取得操作と属性値設定操作は頻繁に使用することと算術演算を記述しやすくするため，組込み関数でなく，それぞれ，“<--”，“->>” の記号を用いることとする．制御文では，条件文（IF THEN），選択文（SWITCH CASE），繰返し文（WHILE）を記述できる．また，1 対 N の対応付けの際に，オリジナルの MIB における複数の属性値の変更の一貫性を保証する必要があるため，トランザクション処理の開始（TRANS_BEGIN）と終了（TRANS_END）を記述する．

4.4 対応関係記法を用いた記述例

対応関係記法による記述例として，ベンダ独自の MIB 定義（オリジナルの MIB）を持つ Fore 社製 ATM 交換機に，標準の M4 インタフェースの MIB 定義に従うビューを提供させるための構成情報，性能情報，通知の対応関係記述を以下に示す．

(1) 構成情報の対応付け（静的な情報の対応付け例）

図 4 にビューの MIB の MO クラス “atmAccessProfile” の属性型 “maxNumActiveVCCsAllowed” の対応付けの一部を示す．この属性型は 1 つのポートの VCC（Virtual Channel Connection）の最大数を表す．オリジナルの MIB では，MO クラス “pathEntry” の属性型 “pathMaxChannels” が 1 つの VCC の最大個数を表す．1 つのポートはいくつかのパスを収容し，1 つのパスはいくつかのチャネルを収容する．

これより，ビューの MIB の 1 個の MO インスタンス “atmAccessProfile” は，オリジナルの MIB の複数の MO インスタンス “pathEntry” に対応する．例は，(a) 対応するオリジナルの MIB の複数の MO インスタンスに対して（12 行目から 22 行目），(b) その MO インスタンスごとに属性値を取得し（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．

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

図 4 構成情報の対応付け例
Fig. 4 Example of mapping in configuration information.

上記 (a) で 1 対 N の MO インスタンスの対応付けを行うため，組込み関数 GET_INDEX（12 行目）と GET_INDEX_NEXT（20 行目）を使用する．また，N 対 1 の MO インスタンスの対応付けをするために，記号 “#”（15 および 18 行目）を使用する．これは “#” の左の属性型が右の MO インスタンスに束縛されることを表す．対応するオリジナルの MIB の属性値を取得するためには，“<--”（18 行目）を使う．これは M-Get 操作に対する管理操作をオリジナルの MIB の属性値に行い，取得した右辺の値を左辺に代入することを示す．

(2) 性能情報の対応付け（あらかじめ集約・加工を行う例）

図 5 にビューの MIB の MO クラス “cellLevelProtocolCurrentData” の属性型 “discardedCellsInvalidHeader” の対応付けの一部を示す．この属性型は，1 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 分ごとの積算値を示す．オリジナルの MIB 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し，また，送受別々に積算している．このため，オリジナルの MIB の 6 つの属性値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 分ごとの性能データからの差分を計算してビューの属性値を得る（9 行目）．15 分ごとの性能データは，永続変数として保持し（13 行目），Timer 規則により 15 分ごと（11 行目）に，17 行目から 21 行目の処理

```
1  cellHeaderProtocolCurrentData MO_BEGIN
2  { administrativeState {  /* 省略 */ }
3   discardedCellsInvalidHeader {
4    DECLARE {  tmp    (INTEGER); } END_DECLARE
5    GET_PROC {
6     tmp <-- atmLayerEntry%atmReceivedCells - aal4Entry%aal4ReceivedCells
7          - aal5Entry%aal5ReceivedCells + aal4Entry%TransmittedCells
8          + aal5Entry%TransmittedCells - atmLayerEntry%atmTransmittedCells ;
9     %discardedCellsInvalidHeader = tmp - discardedcellsinvalidheader_former ;
10   } END_GET_PROC   } ;
11 TIMER_PROC INTERVAL = 15M {   /* Timer規則 */
12    DECLARE {
13     KEEP_VALUE discardedcellsinvalidheader_former (INTEGER) = 0;
14     discardedcellsinvalidheader_latter (INTEGER);
15     discardedcellsinvalidheader_differ (INTEGER); /* 省略 */
16   } END_DECLARE
17   discardedcellsinvalidheader_latter
18     <-- atmLayerEntry%atmReceivedCells - aal4Entry%aal4ReceivedCells
19          - aal5Entry%aal5ReceivedCells + aal4Entry%TransmittedCells
20          + aal5Entry%TransmittedCells - atmLayerEntry%atmTransmittedCells ;
21   dicardedcellsinvalidheader_former  = dicardedcellsinvalidheader_latter ;
22   /* 新旧値の入れ替え */    /* 省略 */
23 } END_TIMER_PROC  /* 省略 */
24 } MO_END
```

図 5 性能情報の対応付け例
Fig. 5 Example of mapping in performance information.

```
1 tcAdaptorTTPBidirectional MO_BEGIN
2 {
3   m3100alarmStatus {
4    DECLARE { KEEP_VALUE tc_alarmstatus  (INTEGER) = @Mcleared ;
5    } END_DECLARE
6    GET_PROC {
7       %m3100alarmStatus = tc_alarmstatus ;  } END_GET_PROC
8  } ;    /*  省略  */
9  NTF_PROC {communicationsAlarm: RELATED_TO InternetAlarm : Internet ;
10  } END_NTF_PROC /*  省略  */
11 } MO_END  /*  省略  */
12 Internet ( in InternetAlarmInfo ) {
13  DECLARE { out      (AlarmInfo);
14  } END_DECLARE
15  IF ( in.probableCause == {1 3 6 1 4 1 326 2 2 0 0} )
16  THEN { /* asxSwLinkDown受信 */
17      out.probableCause = {arfProbableCause 27} ;
18      out.perceivedSeverity = @major ;
19      tc_alarmstatus = @ActiveReportable_Major ; /* 省略 */
20      emit_Notification( communicationsAlarm, KEEP, out ) ;
21   } END_IF
22 } ;
```

図 6 通知の対応付け例
Fig. 6 Example of mapping in notification.

により更新する．

(3) 通知の対応付け（動的な振舞いの例）

図 6 にビューの MIB の MO クラス “tcAdaptorTTP-Bidirectional” から発行する通知の規則の一部を示す．9 行目から 10 行目の Notification 規則では，オリジナルの MIB からの通知 “InternetAlarm” に対して，ビューの MIB の通知 “communicationsAlarm” を発行し，その際の手続きは “Internet” であることを示す．通知 “InternetAlarm” のパラメータ “probableCause” が条件に合致すれば（15 行目），通知 “communicationsAlarm” のパラメータを設定し（17，18 行目），通知を発行する（14 行目）．この際，ビューの MIB の属性型 “m3100alarmStatus” の Attribute 規則（3 行目から 8 行目）で定義された永続変数（4 行目）に値を設定して（19 行目），障害状態の保持を行う．

5. ビュー提供プログラムの自動生成

ビュー提供プログラム生成コンパイラは，様々なビューを柔軟に提供するため，**図 7** に示すように，①オリジナルの MIB 定義，②ビューの MIB 定義および，③これらの定義間の対応関係を 4 章の記法に従って記述した対応関係記述を入力とし，C 言語のビュー提供プログラムを生成する．また，識別名対応情報ジェネレータは，上記①～③に加え，④オリジナルの MIB の名前木と⑤ビューの MIB 独自の識別名に用いる属性（以下，名前属性と呼ぶ）の値からビュー提供プログラムが利用する識別名対応情報（5.2 節参照）を生成する．

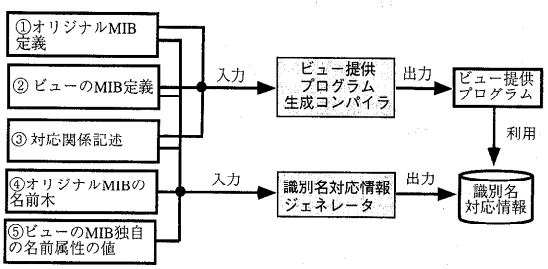


図 7 ビュー提供プログラム生成コンパイラ
Fig. 7 Compiler for generation of view program.

5.1 コンパイラが生成するビュー提供プログラム

ビュー提供プログラムは**図 8** に示すように，(1) 制御モジュール，(2) 変換処理モジュール，(3) インスタンス管理モジュール，(4) タイマモジュールから構成する．

(1) 制御モジュール

ビューの MIB に対する管理操作およびオリジナルの MIB からの管理操作応答や通知はすべてこの制御モジュールが受信する．制御モジュールは，ビューの MIB に対する管理操作やオリジナルの MIB からの通知を受信した際に，新たな変換処理モジュールを割り当てて管理操作を受け渡す．さらに，オリジナルの MIB からの管理操作応答を該当する管理操作に割り当てた変換処理モジュールに振り分ける．また，タイマモジュールからのタイムアウト通知を受信すると変換処理モジュールにタイムアウトの処理を依頼する．

表 4 ビュー提供プログラムの規模
Table 4 Program size of the view program.

ビュー提供プログラム	規模(Kstep)
(1)制御モジュール	4.9
(2)変換処理モジュール	対応関係記述に依存 (注)
(3)インスタンス管理モジュール	8.4
(4)タイマモジュール	0.6

(注)27個のMOクラスを持つM4インタフェースをビューとし，Fore社製ATM交換機をオリジナルのMIBとした場合の対応関係記述は8.5Kstepであった．自動生成したプログラムの大きさは30.0Kstepになる．ただし，プログラムの大きさにGDMOコンパイラとASN.1コンパイラが生成するプログラムは含まない．

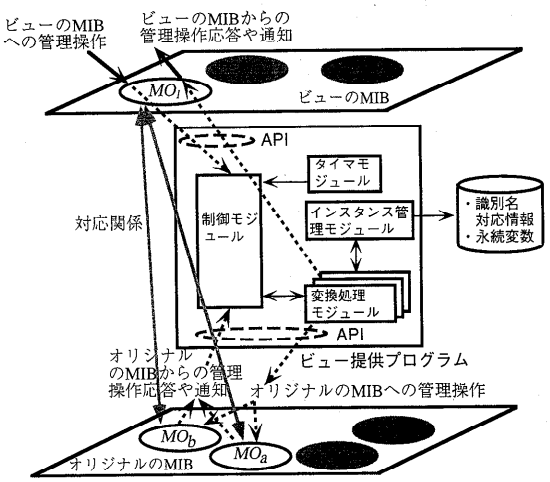


図 8 ビュー提供プログラム構成
Fig. 8 Configuration of view program.

(2) 変換処理モジュール
変換処理モジュールは，対応関係記述に従った管理操作の変換の処理を行う．ここでは，対応関係記述には記述されないが変換処理で必要となる，①MO クラスや属性型の OID の変換，②属性値の符号化と復号のための関数呼び出し，③ビューとオリジナルの MIB における MO インスタンス間の変換や，④オリジナルな MIB の属性値の設定や取得のための管理操作呼び出し等をコンパイラが生成する． また，ビューの MIB に対する管理操作，オリジナルの MIB からの通知，タイムアウトの通知を同時に受信した際でも変換処理を並列に実行可能とするため，それぞれに対し 1 つの変換処理モジュールを割り当てる．

(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，識別名対応情報や永続変数を管理する．ビューまたはオリジナルの MIB の MO インスタンスの生成および削除にともなった識別名対応情報の変更や，変換処理にともなった永続変数の値の更新や参照は，すべてインスタンス管理モジュールを経由して行う．

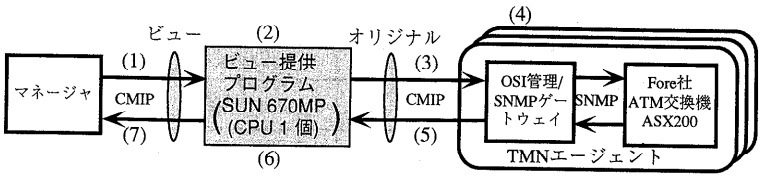
(4) タイマモジュール
対応関係記述で指定されたタイマのタイムアウトを制御モジュールへ知らせる．タイマは，変換処理モジュールが周期的にオリジナルの MIB の属性値を取得し，ビューの MIB の属性として提供する等のために使用する．

5.2 ジェネレータが生成する識別名対応情報

識別名対応情報は，ビューの MIB の名前木と，その名前木に対応するオリジナルの MIB の MO インスタンスの識別名の組を保持する．ビューの MIB における名前属性の値は，オリジナルの MIB の属性値から直接導出する場合と，独立に割り当てられる場合がある．このため，識別名対応情報ジェネレータは，前者の場合にはオリジナルの MIB の名前木と名前属性の型の対応関係記述を用いて導出して保持する． 後者の場合には，前者の処理を行った後，ユーザの定義した識別名として保持する．

6. 評価と考察

提案方式の有効性を実証するため，ビュー提供プログラム生成コンパイラおよび識別名対応情報ジェネレータを実装し，ビュー提供プログラムを生成させた．生成プログラムは，2.2 節で述べた 3 つの実現形態（マネージャ，プロキシー，エージェント）のうち，今回プロキシーの形態に対応するプログラムとした． また，ここでは各モジュールをプロセスとし，CMIP プロトコル処理には既存の CMIP ボード [14] を使用した．表 4 に 5.1 節のビュー提供プログラムの各モジュールの規模を示す．なお，ビュー提供プログラム生成コン



- (1), (3), (5)および(7) はそれぞれマネージャとビュー提供プログラム間またはビュー変換プログラムとエージェント間の通信時間.
- (2) と (6) はともにビュー提供プログラムの処理時間.
- (4) は OSI管理/SNMPゲートウェイの処理時間を含むエージェントの処理時間.

図 9 試験構成と観測点
Fig. 9 Test configuration and its measurement points.

表 5 ビュー提供プログラムの処理時間
Table 5 Processing time in the view program.

測定項目	処理時間 (msec) (注1)
(a) 属性型vpCTPIdの値取得 (MOインスタンスが1対1に対応)	105.5 (236.6)
(b) 属性型maxNumActiveVCCsAllowedの値取得 (MOインスタンスが1対17に対応)	829.8 (1308.7)
(c) 属性型cellScramblingEnabledの値取得 (属性型が1対2に対応)	156.3 (290.2)
(d) 属性型m3100alarmStatusの値取得 (注2)	83.4 (124.4)
(e) 属性型ingressCDVToleranceの値設定	221.1 (314.2)
(f) 通知communicationsAlarmの受信	114.6 (168.5)
(g) MOインスタンスatmCrossConnectionの生成 (注3)	184.2 (499.9)

(注1) 数値は図9の(2)と(6)の和. 括弧内の数値は(1)から(7)までのすべての和. (f)の数値は(6)の値. (f)の括弧内の数値は(5)から(7)までの和.
(注2) この属性値はビュー提供プログラムに永続変数として保持されるため, 括弧内の値は図9の(1), (2), (6), および(7)の和.
(注3) このMOインスタンス生成は, ビューのMOインスタンス生成に加えて, オリジナルへの3個の属性値設定のための管理操作送受とオリジナルからの1個の通知受信に対応付けられる.

パイラの規模は約 73.1 Kstep であった.

6.1 ビュー提供プログラムの処理時間

ベンダ独自の MIB[10] を持つ Fore 社製 ATM 交換機をオリジナルの MIB とし, 標準の M4 インタフェース[9] のビューとして提供するビュー提供プログラムを対応関係記述から自動生成した. マネージャから, プロキシーを介して ATM 交換機に管理操作を発行したときの処理時間を計測した. マネージャ, プロキシーならびにエージェントは LAN (Ethernet) で接続した. なお, ATM 交換機の MIB は SNMP の MIB であるため, OSI 管理/SNMP ゲートウェイ[13] を使用して OSI 管理の MIB に変換させた. **図 9** に試験構成と観測点を示し, **表 5** にビュー提供プログラムの処理時間を示す. 表 5 のどの測定項目においても実用的な処理時間を達成している.

6.2 ビュー提供プログラムの開発効率

図 10 に対応関係記述の規模と生成されるプログラムの規模の関係を示す. 生成されるプログラムの規模は入力となる対応関係記述の規模の約 3 倍である. つまり, 本方式により, プログラムのコーディング量が人手で開発した場合の約 1/3 に削減され, 開発効率が向上する. これは 5.1 節 (2) で述べた対応関係記述には記述しなくて済む処理が生成されるためである.

6.3 対応関係記法の記述能力

Fore 社製 ATM 交換機の MIB に対して M4 インタフェースのビューを提供するための対応関係を記述した.

静的な情報の対応付け機能では, ビューの MIB で定義される MO クラス "atmAccessProfile" 等の ATM 交換機固有の 27 種類の MO クラスをすべてオリジナ

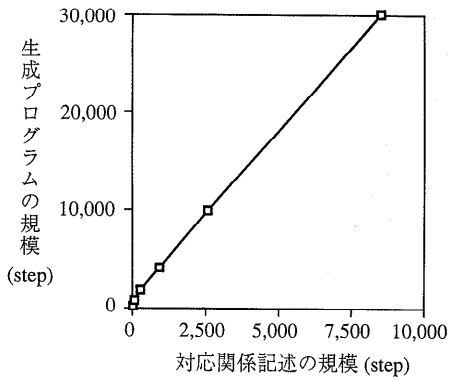


図 10 対応関係記述に対する生成プログラム規模
Fig. 10 Size of generated program to size of mapping rules.

ルの MIB の MO クラスに対応付けることができた．また，ビューの MIB の 54 種類の属性型の中で，直接 1 対 1 に対応する属性型が 14 種類，直接 1 対 1 には対応しないが，4. 3 節で述べた型変換や算術演算等を使って集約・加工を行うことにより対応付けることで，さらに対応付け可能となる属性型が 33 種類あった．一方，属性型 “numReceivedOAMCells” 等のようにまったく対応付けられない属性型が 7 種類あった．

動的な情報の対応付け機能では，通知や MO インスタンスの生成や削除は運用上考えられるすべてに対応付けられ，動作は 8 種類のうち OAM セル関連を除く，MO クラス “atmFabric” の動作 “m4connect” 等の 6 種類を対応付けられた．

また，周期的に集約・加工を行う機能では，MO クラス “upcNpcCurrentData” 等の 3 カ所で定義し，永続変数は 19 カ所で定義して対応付けた．

以上，対応付けられない属性型や動作が若干あるが，MO クラスではすべて，属性型や動作の主なものではすべてが対応付けられ，ATM 交換機を管理するためには実用上大きな問題とはならない．

6.4 ビュー提供方式の複数段適用

提案方式を複数段繰返し適用することにより，管理目的に合った様々なビューを効率的に提供できる．

たとえば，サービス管理レイヤのビューを提供する場合には，まず，NE 装置の MIB に対して NW 管理レイヤのビューを提供し，そのビューをさらに加工してサービス管理レイヤのビューを提供するといった提案方式の 2 段の適用で実現できる．

また，同種の NE 装置でもベンダの違いにより互いに MIB 定義が異なる NE 装置を統合して，ビューを提供する場合，提案方式の 1 段の適用でも実現できるが，オリジナルとビューの MIB の MO クラスや識別名による選択文等の制御文が多重な入れ子構造になり対応関係記述が複雑化し，可読性や保守性の点から好ましくない．このような場合には，まず MIB 定義が同じ NE 装置ごと（ベンダごと）に提案方式を適用して均質なビューを提供した後，さらにこれらのビューを加工して統合したビューを提供するという 2 段の適用により容易に実現できる．

7. おわりに

本論文では，既存の MIB に対し，様々なビューを提供するプログラムを効率的に開発可能とするビュー提供方式を提案した．提案方式では，既存の MIB と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，それを用いた対応関係記述からビュー提供のプログラムを自動生成させることを特徴とする．対応関係記法は，(1) 管理操作受信時に集約・加工を行う静的な情報の対応付け機能，(2) 動的な情報の対応付け機能や，(3) 周期的に集約・加工を行う機能を記述可能とする．

提案方式を実装し，ATM 交換機のビュー提供に適用して，自動生成したビュー提供プログラムの処理時間やプログラム開発効率等の観点から提案方式の有効性を実証した．

今後，ネットワークの高度な運用・管理を可能とするため，TMN のネットワーク管理レイヤやサービス管理レイヤ，ならびに，顧客網管理（CNM）の実装が進むことが予想される．管理目的に応じて既存の NE 装置の MIB を集約・加工した新たなビューを提供可能とする提案方式は，これらの実装に有効な方式である．

謝辞 日頃ご指導いただく国際電信電話（株）研究所村上仁己所長に感謝します．また，ご討論いただいた（株）KDD コミュニケーションズ黒木哲也担当課長に感謝します．

参考文献

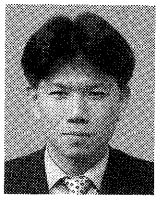
1) ITU-T Rec. M.3010: Principles for Telecommunications Management Network (1992).
2) ITU-T Rec. X.160: Architecture for Customer Network Management Service for Public Data Networks (1995).
3) Klerer, S.M. and Cohen, R.S.: Distribution of Managed Object Fragments and Managed Object Replication: The Data Distribution View of Management Information, *IFIP Trans. Integrated Network Management*, II, pp.763–774 (1991).
4) 木原，山室，磯部：管理オブジェクト情報集約機構の構成法，電子情報通信学会技術研究報告，

IN93-12 (1993).
5) 宮内，中川路，三上，水野，青野，檜山，曽我：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, pp.1426–1440 (1993).
6) Kalyansundaram, P. and Sethi, A.S.: An Application Gateway Design for OSI-Internet Management, *IFIP Trans. Integrated Network Management*, III, pp.389–401 (1993).
7) 堀内，吉原，杉山，小花: TMN 管理情報ベース（MIB）のためのビュー変換の実現方式，情報処理学会マルチメディア通信と分散処理ワークショップ論文集，pp.57–64 (1996).
8) 吉原，堀内，杉山，小花：TMN 管理情報ベース（MIB）のためのビュー変換方式の実装と評価，第 54 回情報処理学会全国大会論文集，2U-07 (1997).
9) ATM Forum af-nm-0027: CMIP Specification for the M4 Interface (1995).
10) Fore Systems: Fore System MIB (1994).
11) ITU-T Rec. I.732: ATM Management of the network element view (1996).
12) ITU-T Rec. X.722: Guidelines for the Definition of Managed Objects (1992).
13) 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，電子情報通信学会技術研究報告，IN94-85 (1994).
14) 加藤，堀内，井戸上，鈴木：パソコン用 CMIP ボードの開発，1992 年電子情報通信学会秋季全国大会，B-414 (1992).

(平成 9 年 5 月 19 日受付)
(平成 9 年 11 月 5 日採録)

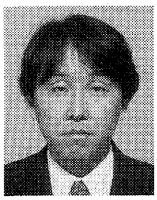
堀内 浩規（正会員）

昭和 35 年生．昭和 58 年名古屋大学工学部電気工学科卒業．昭和 60 年同大大学院情報工学専攻修士課程修了．同年国際電信電話（株）入社．現在，同社研究所ネットワーク管理グループ主任研究員．この間，ネットワークアーキテクチャ，OSI プロトコル実装方式，通信プロトコルの形式記述技法，ネットワーク管理の研究に従事．平成 4 年度電子情報通信学会学術奨励賞，平成 8 年度情報処理学会全国大会大会優秀賞を各受賞．電子情報通信学会会員．



吉原 貴仁（正会員）

昭和 45 年生．平成 5 年東京工業大学情報工学科卒業．平成 7 年同大大学院理工学研究科情報工学専攻修士課程修了．同年国際電信電話（株）入社．現在，同社研究所ネットワーク管理グループ主任．この間，ネットワーク管理，ネットワークアルゴリズム，分散処理の研究に従事．電子情報通信学会会員．



杉山 敬三（正会員）

昭和 37 年生．昭和 60 年京都大学工学部情報工学科卒業．昭和 62 年同大学院修士課程修了．同年国際電信電話（株）入社．現在，同社研究所ネットワーク管理グループ主査．この間，OSI プロトコル実装，ネットワークアーキテクチャ，分散処理，ネットワーク管理，EDI の研究に従事．平成 6 年度電子情報通信学会学術奨励賞受賞．電子情報通信学会会員．

小花 貞夫（正会員）

昭和 28 年生．昭和 51 年慶應義塾大学工学部電気工学科卒業．昭和 53 年同大大学院修士課程修了．同年国際電信電話（株）入社．現在，同社研究所ネットワーク管理グループリーダ．工学博士．この間，パケット交換方式，ネットワークアーキテクチャ，OSI プロトコル実装，データベース，ビデオテックス，分散処理，ネットワーク管理の研究に従事．電子情報通信学会会員．

鈴木 健二（正会員）

昭和 20 年生．昭和 44 年早稲田大学理工学部電気通信科卒業．昭和 44 年から 45 年までオランダのフィリップス国際工科大学に招待留学．昭和 51 年早稲田大学大学院博士課程修了．工学博士．同年同年国際電信電話（株）入社．現在同社研究所副所長．この間，磁気記録，パケット交換方式，ネットワークアーキテクチャ，高速・分散処理の研究に従事．平成 4 年度電子情報通信学会業績賞，平成 7 年度科学技術庁長官賞を各受賞．平成 5 年度より電気通信大学大学院情報システム学研究科客員教授．本会理事．電子情報通信学会，IEEE 各会員．