

オブジェクト指向に基づく管理情報ベース

2V-3

石場 淳 檜垣 伸俊 井崎 智子 水野 治展

松下電器産業(株) 情報通信研究センター

1. はじめに

マネージャエージェント方式で実現されるネットワーク管理システムにおいて、管理プロトコルを意識せず管理対象にアクセスするためには、管理情報ベース(MIB)は、通信システムの機能とデータベースの機能を兼ね備え、且つ、拡張性とすばやい応答性が要求される。そこで本稿では、これらの要件を満足するMIBをオブジェクト指向に基づき開発したので、その実現方式について報告する。

2. MIB設計の要件

MIB構築時に必要な基本的な設計要件は、以下のとおりである。

(1) 通信システムとしての要件

- ・管理対象とのリンク機能
- ・異なる規格^[1-2]に基づく管理対象の一元管理

(2) データベースとしての要件

- ・複雑な情報構造^[3]をもつ管理対象の統一的管理
- ・利用者の目的に合致した柔軟なインタフェース
- ・すばやい応答性
- ・障害時の一貫性保証

市販のRDBやOODBは、これらの面で一長一短があり、MIB構築に適しているとは限らない。

3. MIBの機能

3.1 MIBアクセスの抽象化

管理の対象となるネットワークの包含木を図1に示す。

インスタンスfddiNetworkで識別される幹線のネットワークに接続されている支線のネットワーク状態を表示するために、必要なデータ項目(属性)を並べた表1に示すビュー(管理情報の見え方)を定義する。また、支線全体の状態よりもさらに詳細な状態を表示したい場合には、表2に示すような支線に接続されているHubの状態を表示するためのビューを定義する。さらに、表2で定義したビューに対して、pana01というローカル名のHubの状態を検索せよ等という要求が行えるMIBインタフェースがあれば、表2のデータ項目をすべて埋めずに検索できる。このように、ビューを定義することによ

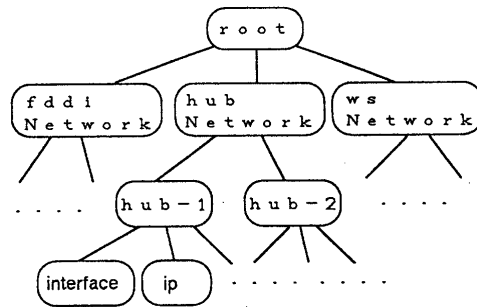


図1 包含木の例

表1 支線状態表示ビュー

ノード番号	ホート番号	ネットワーク名	状態	接続情報		
				タイプ	名前	
10	1	Ether-1	0 1 1	hub	pana01	.
.

表2 Hub状態表示詳細ビュー

hub名	hub状態	接続元情報 ポート番号	IF情報	IP層情報
pana01	OK	10	.	1.2.3.4 . . .
pana02	OFF	20	.	1.2.3.5 . . .
.

り、利用者の目的に合わせた管理情報の見え方で処理することができ、MIBアクセスの自由度が広がる。さらに、本MIBによれば、利用者は、各ビューのデータ項目それぞれがどのような手段(検索のための管理プロトコル等)によって操作できる管理情報なのか、どこにどのような構造で格納されているのか等を意識せずMIBにアクセスすることができる。

本MIBは、利用者が自由に作成したビューとそれに対する条件をオブジェクトに対する管理操作要求として受け付ける外部スキーマを備えている。実装したスキーマについてはあとで簡単にまとめる。

An Object Oriented Implementation of Management Information Base

Atsushi ISHIBA, Nobutoshi HIGAKI, Tomoko IZAKI, Harunobu MIZUNO

INFORMATION & COMMUNICATION RESEARCH CENTER,
MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.

3.2 スレッドによるメッセージ処理

インタフェースに柔軟性をもつMIBを構築するためにオブジェクト指向のアプローチを試みた。MIB利用者からのメッセージの処理をスレッドによって行う方式を図2に示す。ここでいうMIB利用者は、例えばMMI処理を行うプロセス等を指す。

MIB利用者は、作成したビューに対して条件を加えた管理操作要求を含むメッセージを外部スキーマのオブジェクトに送る。このオブジェクトは親スレッドを生成し、実際の操作を行う物理スキーマのメソッドをもつ複数のオブジェクトに処理を依頼する。例えば外部スキーマのオブジェクト1は、物理スキーマのオブジェクトA-1とB-1に対して振り分けた要求をメッセージとして送っている。メッセージを受けた物理スキーマの各オブジェクトは子スレッドを生成し、メッセージに含まれている要求を自分のメソッドを使って処理し、外部スキーマのオブジェクトに対して操作完了のメッセージを送り、子スレッドを終了させる。外部スキーマのオブジェクトは、すべての物理スキーマのオブジェクトから操作完了のメッセージを受けると、結果を収集して、利用者に管理操作の応答を返し、親スレッドを終了させる。

図3にオブジェクトの処理シーケンスの例を示す。外部スキーマのオブジェクト1とオブジェクト2は、スレッドを使うことによって物理スキーマのオブジェクトとは独立に処理できるので、図3のように要求1のオブジェクトA-1の処理に時間がかかる場合でも、要求2の処理のほうが早ければ、先に応答を返すことができる。

このように、外部スキーマによる要求をオブジェクトとし、必要なだけ物理スキーマを実行するオブジェクトに振り分けて、処理を依頼することによって管理操作を行うことができる。本MIBでは、外部スキーマは値の参照しか行わないので、複数のビューからの要求を受けつけて並行して処理することができる。

4. 実装

4.1 スキーマ構成

各オブジェクトは3層構造をもつ以下のスキーマの各メソッドで処理を行う。

- (1) 外部スキーマ…ビュー操作、レコード操作、カーソル操作等の処理を行う。
- (2) 内部スキーマ…メッセージのフォーマットを統一する処理を行う。
- (3) 物理スキーマ…実際に値の操作を行う処理で、
 - ・CMIP物理スキーマ
 - ・SNMP物理スキーマ
 - ・値管理の物理スキーマ
 等からなる。

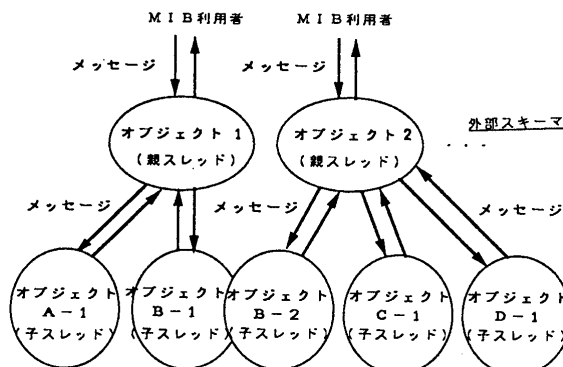


図2 スレッドによるメッセージの処理

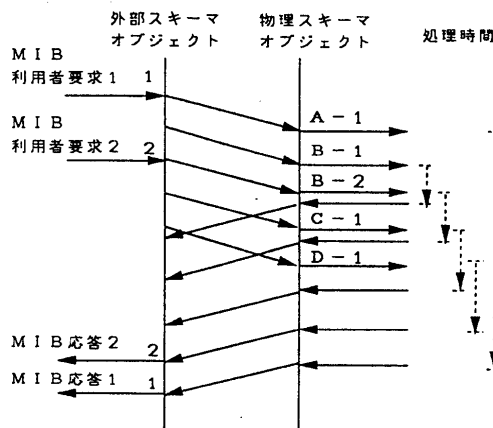


図3 オブジェクトの処理シーケンス

4.2 共通処理

本MIBは次の共通処理の機能をサポートしている。

- ・オブジェクト管理
- ・メッセージ管理
- ・メソッド管理
- ・スレッド管理

5. おわりに

本稿は、MIB内のオブジェクトに対するメッセージの処理をスレッドによって実現する方式について述べた。この方式は、マネージャMIBだけではなくエージェントMIBにも適用できる。今後は、種々の実システムで評価を行うと共に、スキーマの機能の充実を図っていく予定である。

参考文献

- [1] ISO/IEC 9595, 9596 CMIS/CMIP(1990).
- [2] RFC 1157 SNMP(1990).
- [3] ISO/IEC 10165 SMI(1991).