5N－6

ＯＳＩ管理情報支援ツールＭＩＮＴの設計

中川路哲男、宮内直人、勝山光太郎、水野忠則

三菱電機(株)　情報電子研究所

１．はじめに

ＩＳＯ及びＣＣＩＴＴでは、異機種のシステム間での相互接続を可能とする為に、ＯＳＩ（開放型システム間相互接続）の標準化を行なっている。ＯＳＩの需要の高まりと共に標準化も順調に進捗し、各地で接続実験が行なわれる状況である。各種のシステムをネットワークに接続して運用するようになると、ネットワークも大規模化し、その管理が重要な課題となってくる。ＯＳＩでは、安全で且つコストにあった品質の通信を利用者に提供するために、ネットワーク管理（ＯＳＩ管理）の標準化を進めている。ネットワーク管理の標準化は、管理情報の内容とそれを転送する管理プロトコルの二つの側面から行なわれている。

我々は、管理情報を支援するツールＭＩＮＴ(Management INformation support Tool)を設計したので、本稿ではその機能・構成について報告する。

２．管理情報とテンプレート

ネットワーク管理は、管理されるシステム（エージェント）側で管理対象毎に管理情報を収集してＭＩＢ（管理情報ベース）に蓄積し、管理するシステム（マネジャ）との間で、管理プロトコルを使用してその内容を検索、設定、報告することにより行なわれる。

管理情報は、管理対象毎に規定されるものであり、管理対象によって多種多様のものが想定される。そのため、その体系や、汎用的な定義方法（テンプレート）がＩＳＯで検討されている[1]。テンプレートは、その管理対象が持つ属性の内容、生成・消去の条件、通知・動作などの振舞いの内容、他の管理対象との関係などを定義するためのものである。情報形式の規定には、ＡＳＮ．１を使用している[2]。テンプレートの例を図１に示す。

テンプレートによる管理情報の定義の内、エンティティやコネクションなど一般的なものはＯＳＩ管理の標準の一部として、各プロトコルに固有のものはプロトコル規格の一部として標準化されるが、その多くが未だ定義されていないのが現状である。

３．ＭＩＮＴの概要

ＯＳＩ管理を実現するためには、テンプレートで規定された管理情報を、管理対象毎に実システムで収集できる情報構造として定義し、システム運用中に収集・蓄積し、更にマネジャからのアクセスやマネジャへの通知時に管理プロトコルで転送できる形式に変換する必要がある。管理情報は各層あるいは各プロトコル毎に存在するが、プロトコル処理ソフトウェアや管理アプリケーションの処理を考慮すると、統一的な方法でアクセスできることが望ましい。

我々は、管理情報モデルがオブジェクト指向の概念で設計さ

```
オブジェクトクラステンプレート
ExampleObjectClass MANAGED OBJECT CLASS
  DERIVED FROM      ISO/IEC 10165-2:top
  CHARACTERIZED BY:
    BEHAVIOUR DEFINITIONS     CommunicationErrorBehaviour;
    ATTRIBUTES        QOS-Error-Cause GET;
    OPERATIONS        CREATE with-automatic-instance-naming;
                      DELETE deletes-contained-objects;
    NOTIFICATIONS  CommunicationError;
REGISTERED AS {ObjectClass 1};

属性テンプレート
QOS-Error-Cause ATTRIBUTE
  WITH ATTRIBUTE SYNTAX      AttributeModule.QOSErrorCause;
  MATCHES FOR                Equality;
  BEHAVIOUR                  QOSErrorBehaviour;
REGISTERED AS {AttributeID 2};

振舞いテンプレート
CommunicationErrorBehaviour BEHAVIOUR
  DEFINED AS  The CommunicationError notification is generated by ....

通知テンプレート
CommunicationError     NOTIFICATION
  BEHAVIOUR  CommunicationErrorBehaviour;
  WITH DATA SYNTAX       EventModule.ErrorInfo;
  WITH RESULT SYNTAX     EventModule.ErrorResult;

関連ＡＳＮ．１モジュール
AttributeModule.DEFINITIONS ::= BEGIN
QOSErrorCause ::= ENUMERATED {
  responseTimeExcessive(0), queueSizeExceeded(1),
  bandwidthReduced(2), retransmissionRateExcessive(3)}
END
EventModule.DEFINITIONS ::= BEGIN
ErrorInfo ::= SET {
      [0] ProbableCause OPTIONAL,
                :
END
```

図１　テンプレートの例

れていることから、ＭＩＢをオブジェクト指向データベースとして実現することを検討している。即ち、管理対象の定義がクラスの定義に相当し、実際の管理対象がそのインスタンスであり、管理情報がそのインスタンス変数に相当する。更に、管理情報の継承も、サブクラスの概念を用いてそのまま実現可能である。このように実現することで、プロトコル処理ソフトウェア及び管理アプリケーションは、管理対象に相当するインスタンスを生成し、それにアクセスするメッセージを送れば良いことになり、モデル通りの簡明な処理となる。

ＭＩＢを実現するソフトウェアは以下の三つの部分から構成される。

- 各管理対象に相当するオブジェクト：管理対象毎の管理情報を蓄積し、アクセスに対して情報を提供する。
- 管理対象の名前木を管理するオブジェクト：各管理対象の名前を管理し、識別名の解析や自動ネーミング機能を提供する。
- フィルタやスコープなどの解析を行なう汎用関数群：管理対象や管理情報に依存しない処理を提供する。

ＭＩＮＴは、テンプレートで記述された管理対象の定義を解析し、管理対象に相当するオブジェクトのクラス定義を自動生成するツールである。使用するオブジェクト指向言語は、過去に我々が開発したｓｕｐｅｒＣ[3]である。ＭＩＮＴを使用することにより、管理対象をテンプレートで規定されたままの形で

Design of the Management Information Support Tool "MINT"
Tetsuo NAKAKAWAJI, Naoto MIYAUCHI, Kotaro KATSUYAMA , Tadanori MIZUNO
Information Systems and Electronics Development Laboratory
MITSUBISHI ELECTRIC CORPORATION

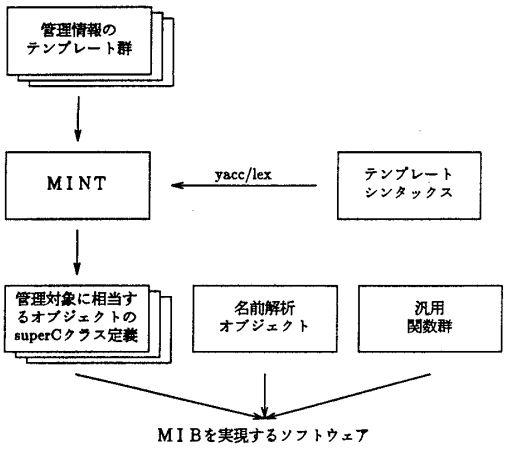

図2　MINTの位置付け

扱うことができ、その内部的な実現方法について意識する必要がなくなる。また、MINTは、テンプレートのシンタックス定義自体をyacc/lexを利用して解析するため、テンプレートのシンタックスの変更・拡張にも柔軟に対応できる。図2に、MINTの位置付けを示す。

4．MINTの機能

(1) テンプレートの解析

MINTは、以下のテンプレートを入力とする。

・オブジェクトクラステンプレート：管理対象の定義であり、その管理対象が持つ属性、操作、通知などを規定する。MINTは、この単位で処理を行ない、superCによるクラス定義を出力する。このテンプレートとクラス定義が1対1に対応する。

・条件テンプレート：管理対象の定義の内、オプショナルな属性、操作、通知を規定したものである。

・属性テンプレート：属性のシンタックスなどを定義するテンプレート。MINTの出力であるクラス定義のインスタンス変数定義に利用される。インスタンス変数は、属性のシンタックスがASN．1で規定されているので、具体的には我々が過去に開発したASN．1ツールAPRICOT[4]によるデータ構造へのポインタに相当する。

・振舞いテンプレート：通知や動作の内容を定義するテンプレート。自然言語で記述されるが、MINTが処理できるように、属性と通知や動作の関係を定義するための簡単なシンタックスを追加定義した。

・動作テンプレート：動作に関する情報シンタックスを定義するテンプレート。

・通知テンプレート：通知に関する情報シンタックスを定義するテンプレート。

また、名前の属性を定義する名前テンプレート、特定エラーテンプレート、グループ属性テンプレートは今回の支援の対象としない。

(2) クラス定義の出力

MINTはオブジェクトクラステンプレートを中心とする上記のテンプレート群を入力として、管理対象のsuperCによるクラス定義を出力する。出力されるクラス定義の内容を図3に示す。管理アプリケーションは、このクラスのインスタンスを生成することで管理対象を生成し、そのインスタンスに対して属性アクセスや動作メッセージを送ることで、管理操作を行なう。

```
クラス名
スーパークラス名
インスタンス変数
  属性へのAPRICOTポインタ
  通知関数へのポインタ
  動作関数へのポインタ
    :
メソッド
  生成／消滅
  初期化
  属性へのアクセス
  通知関数の登録
  動作関数の登録
    :
```

図3　出力されるクラス定義の内容

インスタンスの生成に当たっては、まず名前解析オブジェクトに名前の解析を依頼して、該当するクラスを得る必要がある。インスタンスは、管理対象の存続時期とアクセス性能の観点から、メモリ上・ファイルのいずれに生成することも可能である。ファイルにインスタンスを生成した場合には、ロード操作を行なうことにより、インスタンスがアクセス可能な形となる。また、通知や動作については管理対象から自律的に行なわれる操作なので、通知・動作発生時に呼ばれる管理アプリケーションの関数を、インスタンス生成時に登録する形とした。

属性に対するアクセスは、属性毎に用意されたアクセスメッセージをそのインスタンスに送ることで行なう。属性は、ASN．1による転送構文の形で蓄積されており、それをそのまま属性情報として扱うことで通信処理との整合を図った。振舞いテンプレートで、属性の変更が通知などに影響を及ぼすことが定義されている場合は、その振舞いが発生するか否かのチェックも行なう。

動作については、実際にプロトコル処理を行なう部分で動作を行なう必要があるので、属性に影響のある処理を行なった後、その動作が行なわれる関数を呼び出す。

通知については、通知発生時に呼ばれる管理アプリケーションの関数を呼び出す。

5．おわりに

管理情報を支援するツールMINTについて報告した。今後は、具体的にプロトコルソフトウェアと結合するなどして、その実用性を検証していく予定である。

<参考文献>

[1] ISO DP 10165-4: Structure of Management Information Part 4: Guidelines for the Definition of Managed Objects (1989).

[2] ISO 8824/8825: Abstract Syntax Notation One(ASN.1) (1988).

[3] 勝山、佐藤、中川路、水野: 通信ソフトウェア向けオブジェクト指向言語superC, 情報処理学会論文誌, Vol.30, No.2, pp.234-241 (1989).

[4] 中川路、勝山、宮内、水野: OSI抽象構文記法支援ソフトウェアAPRICOTの開発と評価, 電子情報通信学会論文誌 (D-I), J73-D-I, No.2 (1990).