

## 1 はじめに

ネットワークの高速化にともない、WDM (波長分割多重) 端局装置や SONET 光リング端局装置などが広く普及しているが、これらの端局装置は管理プロトコルとして ANSI (American National Standards Institute) 標準である TL1 (Transaction Language 1)<sup>[1]</sup> を具備するもの (以下、TL1 装置と呼ぶ) が多い。本稿では、通信キャリアの管理体系である TMN (電気通信管理網)<sup>[2]</sup> 環境にこれら TL1 を具備する装置を収容し、統合的な管理を可能とする TL1 対応 TMN Q アダプタを設計したので報告する。

2 TMN<sup>[2]</sup> と TL1<sup>[1]</sup> の比較と設計における課題

## 2.1 TMN と TL1 の比較

管理アーキテクチャ、管理情報モデル、ならびに管理操作プロトコルの観点からの TMN と TL1 の比較を次ページ表 1 に示す。また、管理操作の構成にも差異があり、以下図 1 に装置構成やサービス状態を取得する TL1 の管理操作要求 (RTRV-EQPT) の構成を例示する。

```
RTRV-EQPT:TOKYO-SYS201-1:HS1-PW2H:14195:....
①—②—③—④—⑤
```

①コマンドコード動作部(必須)②第1修飾子(選択)  
③システム識別子(選択)④アクセス識別子(選択)  
⑤相関タグ(必須)

図 1: TL1 の管理操作要求の構成例。

管理操作要求は、管理情報の取得や設定、ならびにクロスコネクットの接続や解放などの要求種別を表すコマンドコード (図 1 ① (動作部) と ② (第 1 修飾子))、要求送信先のシステム識別子 (図 1 ③)、操作対象のアクセス識別子 (図 1 ④)、ならびに応答との対応を取る相関タグ (図 1 ⑤) などのフィールドから構成される。

## 2.2 設計における課題

アダプタは、TMN の管理操作要求および TL1 の管理操作応答や通知を受信した時点でこれらを変換し、TL1 の管理操作要求および TMN の管理操作応答や通知を構成し、TL1 のエージェントや TMN のマネージャに送信する。

表 1 の比較により、アダプタ設計のためには、以下の課題を解決する必要がある。

## (課題 1) 管理操作および管理情報の対応付け

TL1 ではマネージャとエージェントで独自に管理操作を決定できるため、M-GET や M-SET をはじめとする TMN の管理操作の種別やこれに含まれる属性などのパラメータを解析し、解析結果に応じて各フィールドの値を設定し、TL1 の管理操作を構成して対応付ける必要がある。通知の場合も、各フィールドの値の解析結果に応じて TMN の通知に対応付ける必要がある。また、TMN の識別名と TL1 のアクセス識別子など、管理情報の対応付けも必要となる。

## (課題 2) アダプタのプログラム開発工数の削減

TL1 装置の管理情報は装置の種類毎に異なり、装置構成や管理情報を格納するデータベースのスキーマなどに依存する。統合的な管理のためには、様々な種類

Design of TMN Q Adapter for TL1  
Kiyohito YOSHIHARA, Keizo SUGIYAMA, Koji NAKAO  
and Sadao OBANA  
KDD R&D Laboratories Inc.

の TL1 装置の収容が必要となるが、各装置毎に異なるアダプタのプログラム (以下、アダプタプログラムと呼ぶ) を開発すると工数が増大する。このため、工数を削減したプログラム生成が必要となる。

## (課題 3) インスタンス管理の手間の削減

アダプタを介して TMN のマネージャから TL1 装置の管理を行うためには、TL1 装置上のインスタンスに相当する管理オブジェクト (以下、MO と呼ぶ) インスタンスの識別名をアダプタが把握する必要がある。しかしながら、インスタンスの数は一般に膨大なため、初期値などをマニュアルで登録、管理するのは実用的でなく、インスタンス管理の手間を削減する必要がある。

## 3 TL1 対応 TMN Q アダプタの設計

## 3.1 対応関係記法の拡張 (課題 1 への対処)

2 つの GDMO 定義を対応付けるための対応関係記法<sup>[4]</sup>を拡張し、TMN と TL1 の管理操作ならびに管理情報の対応付けの記述を可能とする対応関係記法を導入する (以下、拡張した上記記述を対応関係拡張記述と呼ぶ)。TL1 の管理情報モデルは規定がないため、TMN の管理情報と 1 対多または多対 1 に対応する場合があります。単一の TMN 管理操作に対して複数の TL1 管理操作を送信するなど、管理情報の集約・加工処理も記述可能とする (図 2(1))。また、TL1 装置の管理情報を対応関係拡張記述に含める。以下にその詳細を述べる。

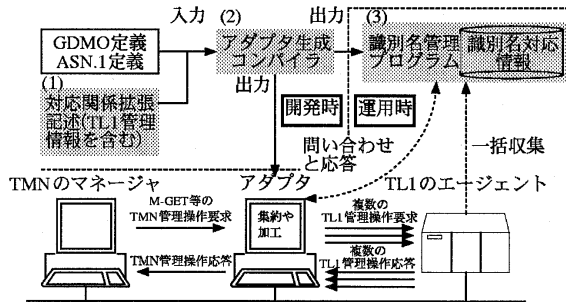


図 2: アダプタの開発と運用の概略。

## (1) 基本構成

M-GET や M-EVENT-REPORT など各 MO クラスの管理操作や通知毎に、これらに含まれる属性や文字列を使って TL1 や TMN の管理操作や通知を組み立てる処理を記述する。

## (2) 新たな予約語の導入

TMN の管理操作要求に応じて各フィールドの値を設定したり、TL1 の管理操作応答や通知を各フィールド毎に解析して管理操作や管理情報を対応付けるため、フィールドに対する値取得や設定、型変換、算術演算、ならびに管理情報の集約や加工処理等の記述を可能とする予約語を新たに導入する (表 2)。

## (3) 文字列操作のための関数の導入

例えば、識別名 (DN) から "HS1-PW2H" (図 1 ④参照) などのアクセス識別子を構成するため、ハイフンなどを区切り記号とする文字列操作が対応付けの際に必要である。このため、フィールドを "-" (ハイフン) や "." (コンマ) などの区切り文字毎にフィールドや文字列を文字列に分解したり、逆に、文字列からフィールドや文字列を合成する関数を新たに導入する (表 3)。

表 1: TMN と TL1 の比較.

	TMN	TL1
管理アーキテクチャ	マネージャとエージェント (注1)	同左
管理情報モデル	GDMO	規定なし (注2)
管理情報の定義	管理オブジェクト (MO) ごと	規定なし
管理情報の構造	名前木に基づく木構造	規定なし
管理情報のシンタックス	ASN.1	ASCII 文字列
管理対象の名前	MO クラスと識別名 (DN)	システム識別子とアクセス識別子
継承機能	あり (オブジェクト指向)	なし
管理操作プロトコル	CMIP	TL1
プロトコルスタック	OSI7 層	規定なし (注3)
コネクションの有無	あり	あり (ログイン・ログアウト)
管理操作	M-GET(読出し) や M-SET(設定) など. 各管理操作に要求と応答がある.	マネージャとエージェントで独自に決定 (注4). 各管理操作に要求, 応答, および確認がある.
通知	M-EVENT-REPORT	マネージャとエージェントで独自に決定
複数管理対象の一括操作	scope と filter パラメータ	アクセス識別子の指定により可 (例 "ALL")
同期機能	synchronization パラメータ	contingency フラグ

(注1) メディエーションなどもある. (注2) 装置構成や装置のデータベースのスキーマに依存. (注3) OSI, X.25, IP などを利用. (注4) 例えば, RTRV-ALM-ALL など, コマンドコードは動作部 (必須), 第1修飾部 (選択), ならびに第2修飾部 (選択) からなる.

表 2: フィールド処理のための予約語の例.

予約語	説明	適用例 (注)
VERB	動作部	RTRV
MOD1	第1修飾子	EQPT
MOD2	第2修飾子	なし
SID	システム識別子	TOKYO-SYS201-1
AID	アクセス識別子	HS1-PW2H
CTAG	関連タグ	14195

(注) 図1の管理操作要求受信時の値.

表 3: フィールド分解・合成のための関数の例.

関数	説明
POP(del, str, n)	文字列 str の先頭から (n-1) 番目 (n>1) と n 番目の区切り del の間の文字列.
APP(del, str1, str2)	文字列 str1 の直後に区切り del と文字列 str2 を連結した文字列.
POPTXT(text, n)	付加情報における n 番目の構成要素 (注).
DELETE(del, str)	文字列 str の両端の区切り del を削除した文字列.

(注) unquoted line, quoted line, または comment のいずれか.

(4) 対応関係拡張記述例

TL1 の通知 "REPT ALM COM" を ITU-T 勧告 X.721 における equipmentAlarm M-EVENT-REPORT に対応付けるための対応関係拡張記述例を図3に示す. ここでは, 先に導入した予約語や関数 (21, 22 行目) を使い, perceivedSeverity 属性 (20~28 行目) や notificationIdentifier 属性 (29 行目) を設定し, equipmentAlarm を送信 (30 行目) する.

3.2 プログラムの自動生成 (課題2への対処)

任意の GDMO 定義, ASN.1 定義, ならびに対応関係拡張記述を入力とし, これらを基にアダプタプログラムを自動生成するアダプタ生成コンパイラ (図2(2)) を提供し, プログラムの開発工数を削減する.

3.3 インスタンス管理の自動化 (課題3への対処)

TL1 装置上のインスタンスのアクセス識別子を一括収集し, これらを MO インスタンスの識別名と対応付け, 問い合わせに応じて MO インスタンスの識別名やアクセス識別子をアダプタに伝送する識別名管理プログラムを提供する. これにより, インスタンス管理の自動化を図り, インスタンス管理の手間を削減する. ここでは, 対応関係拡張記述を基にアダプタ生成コンパイラから自動生成する (図2(3)). 以下にその手順を示す.

(1) 例えば, 図3の15~17行目に示す"NAME\_PROC"節のように, MO インスタンスの識別名とシステム識別子やアクセス識別子との対応を対応関係拡張記述に記述し, これを基に識別名管理プログラムを自動生成する.

(2) アクセス識別子の値を "ALL" や空 (NULL) とする TL1 の管理操作要求を識別名管理プログラムから送信し, TL1 装置上で有効なすべてのアクセス識別子を一

```

1 networkElement MO_BEGIN{ /*MOクラス毎の対応記述*/
2 DECLARE{ /*MOクラス毎の対応記述内で有効な一時変数宣言*/ END_DECLARE
3 managedElementId{ /*属性毎の対応記述*/
4 GET_PROC{ /*属性に対する管理操作毎 (GET) の対応記述 (省略)*/ END_GET_PROC
5 SET_PROC{ /*属性に対する管理操作毎 (SET) の対応記述 (省略)*/ END_SET_PROC};
6 NTF_PROC{ /*通知の対応記述*/ equipmentAlarm: RELATED_TO : sysEqptAlm;
7 } END_NTF_PROC; } MO_END
8
9 PROC_DEF{ /*手続き部*/
10 sysEqptAlm{ /*手続き部の対応記述内で有効な一時変数宣言*/
11 DECLARE{ rdnsq (RDNSequence); fdn (ObjectInstance); out (AlarmInfo);
12 text (GraphicString); cause (GraphicString); severity (GraphicString);
13 } END_DECLARE
14 IF(VERB == "REPT" && MOD1 == "ALM" && MOD2 == "COM") THEN{
15 /* MO インスタンス識別名の対応付け */
16 NAME_PROC{ ringName(rdnsq); sysName(rdnsq); fdn = rdnsq;
17 } END_NAME_PROC
18 /* イベント情報の対応付け */
19 /* probableCause の対応付け (省略) */
20 /* perceivedSeverity の対応付け */
21 text = POPTXT(TEXT, 1); text = DELETE("text); text = POP(;text, 2);
22 severity = POP(;text, 1);
23 SWITCH(severity) {
24 CASE "CR": out.PerceivedSeverity = @Critical;
25 CASE "MJ": out.PerceivedSeverity = @Major;
26 CASE "MN": out.PerceivedSeverity = @Minor;
27 CASE "NA": out.PerceivedSeverity = @Warning;
28 CASE "CL": out.PerceivedSeverity = @Cleared; } END_SWITCH
29 /* notificationIdentifier の対応付け */ out.notificationIdentifier = [INTEGER] ATAG;
30 /* M-EVENT-REPORT の発行 */ emit_Notification(equipmentAlarm, fdn, GTIME, out);
31 } END_IF } /* 手続き sysEqptAlm() 終わり */ } END_PROC_DEF

```

図 3: 対応関係拡張記述例.

括収集する. 次いで, 収集したアクセス識別子, システム識別子, ならびに (1) の記述を基に, 識別名管理プログラムは MO インスタンスの識別名を構成し, これらの対応関係 (図2の識別名対応情報) を作成する. (3) アダプタからの問い合わせがある場合, 要求される MO インスタンスの識別名やアクセス識別子を伝送する. また, M-CREATE や M-DELETE などの管理操作により有効なアクセス識別子を変更する場合には, 識別名対応情報を更新する.

4 おわりに

本稿では, ANSI 標準の TL1 を具備する装置を通信キャリアの管理体系である TMN 環境に収容し, 統合的な管理を可能とするための TL1 対応 TMN Q アダプタの設計を述べた. 今後は設計方針にしたがった実装ならびに実環境評価が課題である. 最後に日頃御指導頂く (株) KDD 研究所村谷拓郎所長ならびに鈴木健二副所長に感謝します.

参考文献

[1] Bellcore. GR-831-CORE: Operations Application Messages - Language for Operations Application Messages, 1996.  
[2] ITU-T. Recommendation M.3010, Principles for Telecommunications Management Network, 1992.  
[3] IETF RFC 1157. A Simple Network Management Protocol (SNMP), May 1990.  
[4] 堀内他. 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. 情処論文誌, Vol.39, No.2, 1998.