

TMNに基づくネットワーク管理へのWWW技術の 適用方式の提案とGDMO/HTML変換ソフトの実装

杉山敬三 吉原貴仁 堀内浩規 小花貞夫

(株)KDD 研究所

和文抄録

Web-based Management は、従来、個々の装置やシステムを対象として検討が行われてきたが、通信事業者のような大規模ネットワークを管理するための TMN(電気通信管理網)に対して、WWW 技術を体系的に適用する方式は論じられていない。本稿では、TMN に基づくネットワーク管理への WWW 技術の体系的な適用方式を提案する。ここでは、TMN の機能階層モデルに従い、一つの機能階層における複数の機能ブロックの統合(水平統合)と、機能階層間にまたがった機能ブロックの統合(垂直統合)を行う。また、提案方式に既存の装置やシステムを収容するために実装した GDMO/HTML 変換ソフトウェアについて報告する。

A Proposal on Applying WWW Technology to TMN-based Network Management and an Implementation of GDMO/HTML Conversion Software

Keizo SUGIYAMA Kiyohito YOSHIHARA Hiroki HORIUCHI Sadao OBANA
KDD R&D Laboratories Inc.

Abstract

Existing Web-based management only deals with individual equipment and systems. No methods have been discussed to apply WWW technology to TMN-based network management. This paper proposes a new method of applying WWW Technology to TMN. In the method, vertical and horizontal integration is made according to the TMN functional hierarchy model. We also describe the implementation of GDMO/HTML conversion software in order to accommodate the existing equipment and systems which support TMN Q interface.

1. はじめに

近年、インターネットのWWW(World Wide Web)を用いた情報発信や情報交換が急速に普及し、WWW技術を用いたネットワーク管理、いわゆるWeb-based Managementが注目されている。従来、ネットワーク管理システムで

は、ネットワーク管理用のプラットフォームを有する専用コンピュータに、固有のアプリケーションを開発することが主流であった。Web-based Managementでは、インターネットやイントラネットを介してWWWブラウザが動作する既存のパソコンやワークステーション上に容易にGUIを提供でき、またWWW

ブラウザ上で動作するJavaプログラムであるJavaアプレットのように、Java言語を用いて開発したアプリケーションはプラットフォームに依存しない。実際のハブやルータ等の機器には、各種の設定をWWWブラウザから行えるものが出現している。

これまでのWeb-based Managementでは、個々の管理対象やシステムでの検討がされてきた^{[1][2]}。また、通信事業者のような大規模なネットワークを管理するTMN(電気通信管理網)への適用例^{[3][4]}も存在するが、特定の装置やシステムあるいはサービスを対象としており、WWW技術の体系的な適用方式については論じられていない。そこで本稿では、TMNに基づくネットワーク管理へのWWW技術の体系的な適用方式を提案する。また、提案した適用方式に既存の装置等を収容するために実装したGDMO/HTML変換ソフトウェアについて報告する。

2. TMN の概要

2.1. TMN のアーキテクチャ

TMN(電気通信管理網)は、巨大化・複雑化する通信網を効率的に保守・運用することを目的としてITU-Tで提案されたネットワーク管理のアーキテクチャ^[5]であり、TMNを実現するのに必要な機能ブロックや管理プロトコル/管理情報が規定されている。機能ブロックの一覧と物理的な構成例を各々表1と図1に示す。

図1の構成要素で、OSはオペレーションシステム(網管理システム)、DCNはLANやWAN等のネットワーク、NEは交換機や伝送装置等の通信装置、WSは表示・入力用端末、MDやQAは管理プロトコル/情報変換のためのゲー

表1 TMN の機能ブロック

OSF (オペレーションシステム機能)	網管理のための情報処理や制御を行う機能
NEF (網要素機能)	端局等網管理の対象となる通信を提供する機能
WSF (ワークステーション機能)	TMNの情報をユーザに提供する機能
MF (メディエーション機能)	OSFとNEF間での管理プロトコル/情報変換機能
QAF (Qアダプタ機能)	非TMN装置をTMN環境に収容する機能

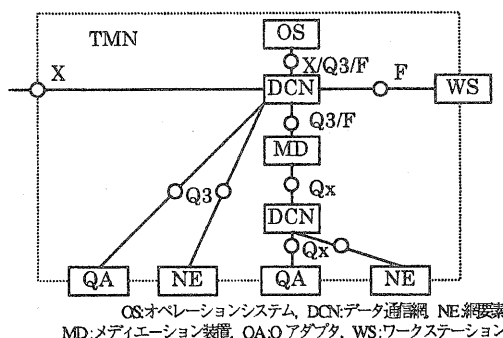


図1 TMN の物理構成の例

トウェイに相当する。図1の○は参照点(インタフェース)を示しており、Qインタフェース(Q3/Qx)にはOSI管理の管理プロトコル(CMIP: 共通管理情報プロトコル)や管理情報定義(GDMO: Guidelines for the Definition of Managed Objects)を使用する。

2.2. TMN の機能階層モデル

TMNの管理機能を運用の側面から階層化したモデルを図2に示す。図2で、NELは網要素自身を示す。EMLでは個々のNEに対する管理を行い、NMLではNEの集合体としてのネットワークを管理する。また、SMLではカスタマに提供される通信サービスとしての側面を扱い、BMLでは企業としてのトータルなビジネスの側面を扱う。

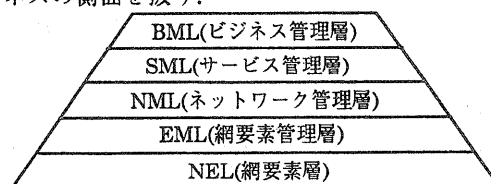


図2 TMNの機能階層モデル

3. TMN に基づくネットワーク管理へのWWW 技術の適用方式の提案

これまで、TMNに基づくネットワーク管理へのWWW技術の適用方式^{[3][4]}が幾つか報告されている。ここでは、従来の適用方式の問題点を挙げ、それらを解決する新たな適用方式を提案する。

3.1. 従来の適用方式とその問題点

従来の適用方式のTMNアーキテクチャにおける位置付けを図3に示す。

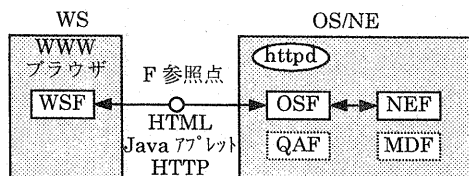


図3 従来の WWW 技術の適用方式

WWWブラウザは図1のWSに相当し、WWWブラウザとOSやNEの間はF参照点として位置付けられる。F参照点では、HTTP(Hyper Text Transfer Protocol)やHTML(Hyper Text Markup Language)等を用い、網管理情報の授受を行う。OSとNEでは各々OSFとNEFが必須の機能であり、MDFやQAFはオプションである。また、OSとNEには、WWWサーバ機能であるhttpdデーモンがある。WWWブラウザはOSやNEからHTMLやJavaアプレットをダウンロードし、OSやNE固有の画面や管理機能を持つ。

この従来方式では、ワークステーションやパソコンを用いて任意の場所からアクセスできたり、OSやNE側の管理機能に変更があってもWWWブラウザには影響しないという利点はあるものの、以下の問題点が存在する。

- ・(問題1)各OSやNE固有のプログラムを動作させるため、操作性が統一されず、それらを連携させることが困難である。
- ・(問題2)WWWブラウザの動作するパソコンやワークステーションを表示(WSF)にのみ使用しており、OSやNE側との負荷分散が行われず効率的でない。

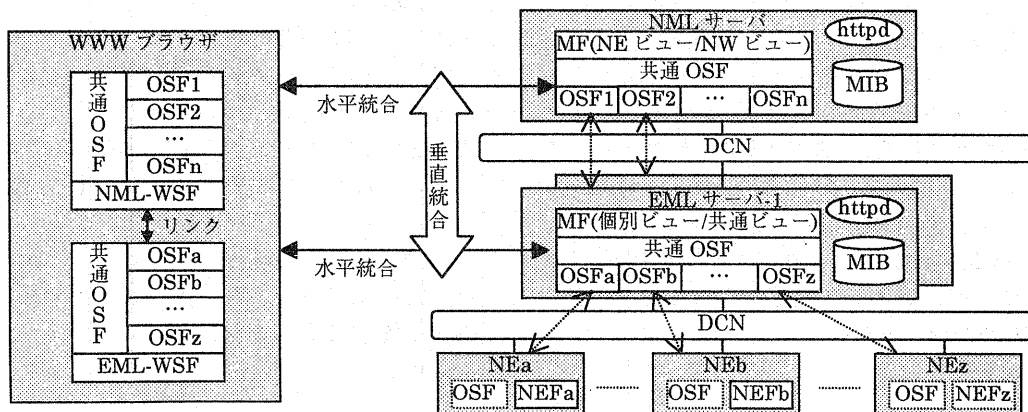


図4 提案する WWW 技術の適用方式

3.2. 提案する適用方式

WWWでは、リンクをたどることで複数のシステムに容易にアクセスできたり、Java言語を利用することで、各種のソフトウェアコンポーネントから容易に所望のソフトウェアを構成することができるという特長がある。そこで、これらの特長を用いて、3.1で挙げた従来の適用方式の問題点を解決するWWW技術の適用方式を提案する。

提案する適用方式では、図2に示したTMNの機能階層モデルに従って、一つの機能階層における複数の機能ブロックを扱う水平統合と、機能階層間にまたがった機能ブロックを扱う垂直統合を行う。そのため、従来のNMLやEMLのOSの代わりに、それらに対応するNMLサーバとEMLサーバを設ける。

提案方式の概念を図4に示す。以下ではNML以下を対象に示すが、SMLやBMLに対しても同じ手法が適用できる。

●水平統合

各機能階層において、従来方式のようにその管理対象である個々のOSやNEに固有の管理を行うのではなく、複数の異種OSFに対するソフトウェアをコンポーネントとして組み合わせ連携させることで、水平方向の統合管理を行う。ここでは、共通的な処理を行う共通コンポーネント(共通OSF)と個々のOSやNE固有の処理を行う個別コンポーネント(OSF_x)を設ける(問題1の解決)。このソフトウェアコン

ポーネントは、WWWブラウザとEMLサーバ/NMLサーバの間での機能分担を行うことで、サーバ側の負荷軽減を図る(問題2の解決)。

例えば、各OSの保持する性能情報にアクセスして同一グラフ上に表示したり、一つのコマンドにより複数のNEに対し同時に終端点の監視状態を設定可能とする。

●垂直統合

従来のようにEMLやNMLといった機能階層毎に独立に管理を行うのではなく、TMNの機能階層の垂直方向にまたがってNMLサーバとEMLサーバ間でリンクを確立することで連携させ、統合的に管理する。このために、HTMLのアンカータグやクリッカブルマップ、あるいはJavaアプレット上の座標やボタンとリンク先のURL(Uniform Resource Locator)を事前に登録し対応付ける(問題1の解決)。

例えば、NMLサーバにアクセスしてネットワークの状態を表示させている時に特定のNEで発生した警報の詳細を知りたい場合、画面上の該当箇所をクリックする等によりEMLサーバにアクセスし、詳細を別ウインドウに表示可能とする。

3.2.1. 機能ブロックの配置

負荷分散を実現するためのTMN機能ブロックの配置を以下に示す。

(1)WSF(ワークステーション機能)

WSFはWWWブラウザ上で実行する。

(2)OSF(オペレーションシステム機能)

OSFは、WWWブラウザ上のJavaアプレットとEML/NMLサーバ間で機能分担する。Javaアプレットはセキュリティ上ローカルマシンのファイルにアクセスできないため、永続性が要求される情報はMIB(管理情報ベース)を有するEML/NMLサーバ側で扱い、その情報を加工する処理はWWWブラウザ側で行う。例えばEML/NMLサーバ側で性能情報を収集し、グラフ表示等に必要の集計等の処理はWWWブラウザ側で行わせる。

(3)MF(メデイエーション機能)

MFには、①NE毎の情報モデル(個別ビュー)と同種のNEに対する共通の情報モデル(共通ビュー)間での変換機能、②NEに対する情報モデル(NEビュー)とネットワークに対する情報モデル(NWビュー)間での変換機能、がある。ここでは、EMLサーバで①を扱い、NMLサーバで②を扱う。

3.2.2. ソフトウェアコンポーネントの実現

ソフトウェアコンポーネントの実現方法に関連して、これまでにJavaアプレット間を連携させる方法が提案されている¹⁰⁾。しかしながら、JavaアプレットはWWWブラウザ上で動作する単独のプログラムであるため、コンポーネントの粒度が大きく、組み合わせで所望のプログラムを構成するのが容易ではない。そこで、WWWサーバ側でも動作可能であり、より小さい粒度のコンポーネントを扱えるJavaBeans¹¹⁾を用いてプログラミングする。

(1)共通コンポーネントの粒度

TMNでは、イベントをロギングしたりフィルタリングする等のシステム管理機能(SMF)のために各OSやNEで共通的に使われる管理オブジェクトが存在し、管理仕様に応じて必要な機能が用いられる。そこで個々のSMFを共通コンポーネントとして扱う。また、操作性の統一を図るため、WSFでは各種のGUI部品等を共通コンポーネントとして共有させる。

(2)コンポーネント間のインタフェース

JavaBeansでは、イベント発行元(イベントソース)にイベント受信先(イベントリスナー)を登録し、イベントソースがイベントリスナーのメソッドを呼び出すことでコンポーネント間の通信が行われる。SMFに対応するコンポーネントとのインタフェースは、管理オブジェクトに対する操作や通知と考えることができ、CMIPに相当するメソッドを提供する。また、共通OSFから複数の個別OSFを同時に制御する場合には、マルチキャスト型のイベントとする。

4. GDMO/HTML 変換ソフトウェアの実装

3章で提案した適用方式において、Qインタフェースをサポートする既存のOSやNEを収容するため、GDMO/HTML変換ソフトウェアをWindowsNT上で、C言語とJava言語を用いて実装した。

4.1. ソフトウェア構成

本ソフトウェアは、WWWサーバ上でWWWブラウザ/サーバ間のプロトコルとQインタフェースの変換処理を行う。変換処理では、①WWWブラウザからのアクセス時にWWWサーバが変換機能を起動するCGI(Common Gateway Interface)方式と、②WWWサーバ上に変換機能を常駐させJavaアプレットが変換機能とのコネクションを確立する方式がある。CGIではアクセス時にプロセスが起動されるためTMNで重要な実時間のイベント通知ができず処理性能も劣るため、②の方式を採用した。

図5に、ソフトウェア構成を示す。管理操作/情報変換モジュールは、①Javaアプレットとの間でコマンドを授受するコマンド解析/応答部、②ASN.1(抽象構文記法1)基本符号化されたGDMOの情報とC言語の構造体間の変換を行うGDMOインタプリタ⁸⁾、③OSやNEとの間でCMIPプリミティブを授受するプリミティブ作成/解析部からなる。CMIPプロトコル処理には既存のCMIPボード⁹⁾を用いた。

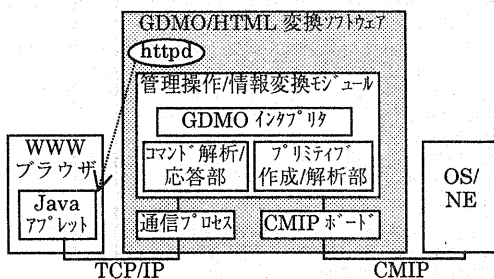


図5 GDMO/HTML 変換ソフトウェア構成

4.2. 管理操作/情報変換

管理情報は、GDMOやASN.1に従って、表示や入力を行うようにした。図6にインスタンス名と属性値の表示例を示す。

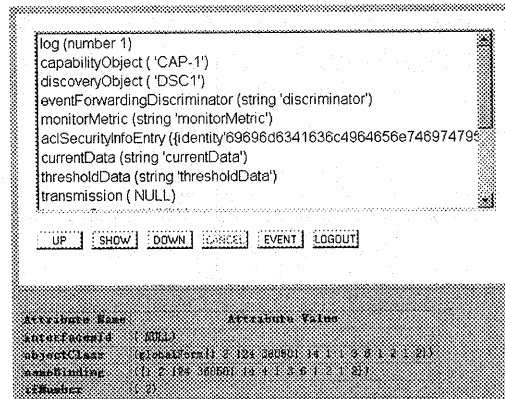


図6 インスタンス名と属性値の表示例

4.2.1. 管理操作変換

指定したインスタンスに対し、それを基点オブジェクトとしてM-GET等の管理操作を発行するボタンや、それを起点とした名前木の上下移動を行うためのボタンを設けた。インスタンスの指定では、Scopeを1stLevelOnlyとしたM-GETを繰り返して発行しインスタンス名の一覧を表示し選択させる。Javaアプレット上でボタンが押下されると、ボタンに対応する文字列をコマンド解析/応答部が受信し、プリミティブ作成/解析部がそれに対応するCMISプリミティブを発行する。

また、M-EVENT-REPORT受信時にはイベントを受信した旨をコマンド解析/応答部が送信し、Javaアプレット上ではそれをボタン点滅により表示する。コマンド解析/応答部では受信したイベントの一覧を保持しており、通知情報の詳細を見る場合にはそれをJavaアプレット上に表示して選択させる。

4.2.2. 管理情報変換

M-GETcnfやM-SETcnfにおける属性値の変換では、GDMOインタプリタの出力する属性値を文字列であるASN.1の値記法に変換し、GDMOで定義された各属性名と属性値のペアをHTMLのTABLEタグに対応付ける。

M-EVENT-REPORTindにおける通知情報やM-ACTIONcnfにおける動作応答も、属性値と同様TABLEタグに対応付ける。

またM-SETreqにおける属性値やM-

ACTIONreqにおける動作情報の入力では、属性や動作情報のASN.1要素に応じ、表2に示すようにASN.1の単純型毎に値入力のためのJavaアプレットの画面と対応付ける。

表2 画面とASN.1要素の対応

画面	ASN.1要素
テキスト入力	INTEGER, BIT STRING(2進入力), OCTET STRING / ANY(16進入力), OBJECT IDENTIFIER, REAL(Mantissa/Base/Exponent), CHARACTER STRING (IA5String等), UTCTime, GeneralizedTime
チェックボックス	名前付き BIT STRING, ENUMERATED, 名前付き INTEGER
ラジオボタン	BOOLEAN, CHOICE, SEQUENCE OF/SET OF の継続

5. 評価と考察

5.1. 提案方式について

従来方式はWWWブラウザを単なる表示端末として用いているため、ワークステーション画面のX端末上への表示と大差なく、Xプロトコルに比べ通信量が削減される程度にすぎない。提案方式は、WWWブラウザが動作するマシンの処理能力を有効利用でき、共通のソフトウェアコンポーネント(共通OSF)を流用することで開発効率の向上が期待できる。

今後全てのOSやNEが本提案方式のようなWWW技術を採用すれば、Qインタフェースの代わりにHTTPやCORBAのIIOP(Internet Inter-ORB Protocol)などWWWに適した通信プロトコルを用いることができる。一方、管理情報モデルについては、Web-based Managementの業界標準であるJMAPI^[10]やWBEM^[11]の管理情報モデルに見られるように、GDMOのようなオブジェクト指向的な管理情報モデルが必要である。

5.2. GDMO/HTML変換ソフトウェアについて

実装したGDMO/HTML変換ソフトウェアの性能を評価するため、処理時間を測定した。測定にはCPUがPentium 133MHz、メモリが32MBのパソコンを用い、管理操作/情報モジュールがWWWブラウザから要求を受けてからM-GETreqを発行するまでの時間と、M-GETcnf(8属性)を受信してからWWWブラウザに属性値を転送するまでの時間を測定した。

その結果、前者が約20ms、後者が約200msであった。これにより、パソコン上でWWWブラウザ・サーバ間のプロトコルとQインタフェースの変換処理を実用的な性能で実現できることがわかった。

6. おわりに

本稿では、TMNに基づくネットワーク管理へのWWWの体系的な適用方式として、TMNの機能階層モデルに従って、一つの機能階層における複数の機能ブロックの統合(水平統合)と、機能階層間にまたがった機能ブロックの統合(垂直統合)を行う方式を提案した。水平統合では、異種オペレーションシステム機能(OSF)を統合的に扱うためにソフトウェアコンポーネントを用い、またWWWブラウザ側にOSFの一部を分担させた。垂直統合では、網要素管理層とネットワーク管理層などの機能階層間でのリンクを確立することで連携させた。さらに、Qインタフェースをサポートする既存のOSやNEを本提案方式に収容可能とするGDMO/HTML変換ソフトを実装し、実用的な性能を有することを確認した。

最後に日頃ご指導頂く(株)KDD研究所村谷拓郎所長、鈴木健二副所長に感謝します。

参考文献

- [1]:M.C.Maston, "Using the World Wide Web and Java for Network Service Management", Proc. of IFIP/IEEE IM'97.
- [2]:B.Reed et al. "Distributed systems management on the web", Proc. of IFIP/IEEE IM'97. 1997
- [3]:Z.Canella et al. "Integrating Web-Based User Interfaces in TMN Systems", Proc. Of IFIP/IEEE NOMS'98
- [4]:J. R. Horowitz, "Bringing Java to TMN", Proc. of IFIP/IEEE NOMS'98
- [5]:ITU-T Rec. M.3010, "Principles for a Telecommunications Management Network", 1992
- [6]:江頭,福井「ネットワーク管理ソフトウェア部品の連携機能」,1997年信学総大,B-7-17
- [7]:<http://java.sun.com/beans/>
- [8]:K.Sugiyama et al. "Implementation and Evaluation of MIB Tester for OSI Management", Proc. of IFIP/IEEE IM'97
- [9]:A.Idoue et al. "Design and Implementation of OSI Communication Board for Personal Computers and Workstations", Proc. Of 11th ICC
- [10]:<http://java.sun.com/products/JavaManagement/>
- [11]:<http://www.microsoft.com/management/wbem/de fault.htm>