社団法人　情報処理学会　研究報告
IPSJ SIG Technical Reports

2005－ＤＳＭ－37（16）
2005／5／13

機能群の調合とインターフェース触診による ネットワークリソース管理機能の抽象化および再構成

三好　優†　　木村　辰幸†

† 日本電信電話株式会社 NTT ネットワークサービスシステム研究所
〒 180-8585 東京都武蔵野市緑町 3-9-11
E-mail: †{miyoshi.yu.kimura.tatsuyuki}@lab.ntt.co.jp

あらまし　ネットワーク管理機能をビジネスロジックから直接扱うため，TMN などではネットワークを抽象化し，NML 以下を「リソース管理層 (RML)」と定義する管理モデルの検討が進められている．下位層を意識することのない高度なネットワーク管理機能実現のためには，実際には差異のある多様な NE の IF を均質的に扱うためのアダプタ開発が必要であり，開発コスト，導入スピードに問題を残している．そこで我々は，NE の IF に対する接続試験によって得られた応答と事前に収集した NE 固有のコマンドパターンを用いた IF の解析によって，抽象化した NM 層 IF から個別のアダプタ記述を自動生成することで NMS 開発を容易にする IF 触診方法を提案する．本稿では，アダプタ記述の分解，モデリング，調合など，我々が提案する IF 触診システムの機能要件について述べる．最後に提案している IF 触診によるアダプタ開発自動化の実現性を，実際の NMS のアダプタ記述を用いた実験により検証する．
キーワード　インターフェース触診，ネットワークリソース管理，MDA，オペレーションシステム機能開発

An Abstraction and a Reconstruction of Network Resources by the Interface Palpation and Preparation Technique

Yu MIYOSHI† and Tatsuyuki KIMURA†

† NTT Network Service Systems Laboratories, NTT Corporation
9-11, Midori-Cho 3-Chome Musashino-Shi, Tokyo 180-8585 Japan
E-mail: †{miyoshi.yu.kimura.tatsuyuki}@lab.ntt.co.jp

Abstract　To use the network management function directly from the business logic, a management model is discussed in TMN and so on. The management model which is considered about an abstraction of the network puts together the NM layer and the EM layer as "Resosurce Management Layer(RML)". To achieve advanced network management function which processes transparently difference of lower layer to operators, we should actually develop adaptors for handling NE interfaces which have different specifications. And service system development costs and a introduction speed are left as problems. And so we have proposed the interface palpation technique that supports development of an NMS. We examine the interface of NE, and obtain the response. Moreover, we use peculiar pattern of NE collected beforehand. In this paper, we describe necessary functions (adaptor descriptions decomposition, modeling and recomposition) for the interface palpation system that we are proposing. Finally, we verify the feasibility of interface palpation system which generates automatically the adaptor description for another NEs by the experiment that uses the adaptor description of actual NMS.
Key words　interface palpation, network resource management, MDA, OSF development

1. はじめに

ネットワークサービスに対する要求が年々多様化している．企業ではネットワークを介したビジネス (e ビジネス) が浸透し，利用サービスごとに異なる品質が詳細に要求されるようになった．また将来訪れるユビキタス社会においては，ユーザの環境が目まぐるしく変化するため，対応するネットワークサービスには頻繁で動的な設定変更が必要となることが想定される．

多様化し複雑化する要求に，これまでのようにネットワークサービスプロバイダのオペレータがきめ細かく対処していくことは難しい．そのため今後は企業オペレータや個人ユーザに対してネットワークの機能の一部を開放して制御を任せたり，自動処理の割合を高めるといった高度なオペレーション機能を持つネットワークが必要になる．

オペレーション機能に関わる標準を策定する TMN[1] では，次世代オペレーション機能開発のアプローチとして，ネットワーク管理層以下を「リソース管理層」と定義し，ネットワークを抽象的に取り扱うための管理モデルの検討が進んでいる．抽象化により，ネットワーク内部の詳細を意識することのないオペレーション機能実装が容易になると考えられるが，実際には多種多様なネットワーク機器 (NE) のインターフェース (IF) の仕様差異への対応など，実用性の担保が課題となる．

我々は，NE の IF に対して実際に接続試験を行い，得られる応答や NE 特有のコマンドパターンを利用して，各 NE に特化した動作コード（アダプタ）としてオペレーション機能を自動で再構成する "触診" と，NE とのインタラクション記述が持つ特徴を抽出して特定の NE に依存しないオペレーション機能記述の共通モデルを作り出す "調合" を提案し，方式と要素技術の検討を進めている．

我々の目的は，上位管理層（ビジネス管理層・サービス管理層）のユーザやオペレータが NE の IF の詳細を知らなくともネットワークを扱うことができるリソース管理機能を開発すること，そのためにまずネットワークサービスプロバイダの開発者が仕様の異なる NE のアダプタを容易に開発するためのサポートをすることにある．

本稿ではまず，提案している触診と調合の概要を述べる．次に，オペレーション機能の一部である既存アダプタ記述の分解，モデリング，記述差異の吸収，有効回答の抽出，触診用の機能記述調合，IF の触診（試験），新規 NE に対する既知のパターンを用いた自動アダプタ生成など，提案する IF 触診システムが具備すべき機能の設計について説明し，最後に IF 触診にて提案しているアダプタ開発自動化の実現性を，実際の NMS のアダプタ記述を用いた実験により検証する．

2. オペレーションシステム開発の動向と課題

2.1 次世代オペレーションシステム開発の動向

TMN では，従来よりオペレーションシステム機能 (OSF) をビジネス管理層 (BML)，サービス管理層 (SML)，ネットワーク管理層 (NML)，装置管理層 (EML) という階層に用途で分類し，各層の機能定義を行ってきた [1]．

そのなかで，e ビジネスや新しいサービスの要求に対応するといった目的から，NML・EML を単に「リソース管理層 (RML)」として表現し，上位のビジネスやサービスのロジックから直接ネットワークを扱えるようにすべきといったソリューションが提案されている (図 1)[2]．この提案では，NE への情報収集や設定といった個別の EML 機能は隠蔽される．そのため上位管理層のオペレーション機能は NE の細かい仕様に関与することなく経路やネットワークといった単位で管理対象を扱うことができるようになり，本来企業オペレータや個人ユーザに関連が深いビジネスやサービスという観点のオペレーション機能（例えば個別のポリシー設定や SLA の監視）の開発が容易となり，変化の激しい環境に対応した迅速なサービス導入に貢献できると考えられる．

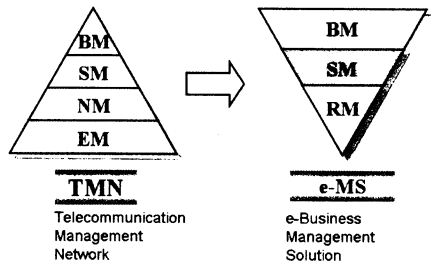

図 1　e ビジネスに対応する管理層モデル (文献 [2] より抜粋)

実現の鍵となる研究のひとつに，モデル駆動アーキテクチャ (MDA) をネットワーク管理システム (NMS) に適用するアプローチがある．[3][4] では，UML をベースにした DEN-ng と呼ばれる情報モデルを定義しており，DEN-ng モデルから自動で NE に対する処理コードを生成することを可能とした (図 2)．これによりネットワークサービスプロバイダの開発者は，NMS の機能とマッピングルールを定義するだけで NE に対する処理コード生成までの作業を自動化できるようになる．

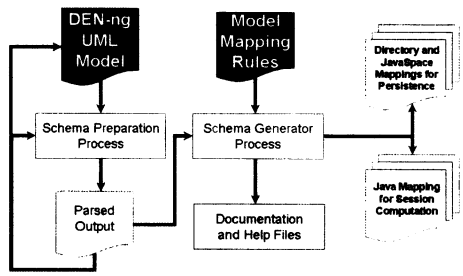

図 2　DEN-ng モデルからのコード生成フロー (文献 [4] より抜粋)

2.2 サービス導入時の課題

しかし，DEN-ng 定義と MDA による一連の処理フローによってオペレーションシステム機能の開発を自動化するにはいくつか大きな課題が存在する．

ネットワークサービスが新規に導入される場合，通常，新機能を持った NE と NE に対して情報収集や設定変更を行う NMS や各種サーバの機能開発が行われる．プロバイダが構築する網はコスト，要求機能，収容能力，地域性といった要因が複雑に絡み合うため，ベンダやプロダクトの異なる様々な NE で構成された系となっている．ベンダが異なれば，NE の IF 仕様はプログラミングモデル・言語・内包する意味までもが異なる．異なる IF を持つ多数の NE と接続した管理機能を実現するため，従来から我々は，管理システムを構成するソフトウェアコンポーネントを共通となる機能部と NE 依存のアダプタ部を切り離して開発してきた．これにより NE のリリースや NE の OS 更改により変更した IF に対してもアダプタ部の追加開

発のみで運用を継続することが可能である．

しかし，アダプタの用意だけでサービス開始までの開発期間を短縮できるわけではない．迅速なサービス導入のためには，NMS などのシステムが NE と同時にリリースされることが望ましい (図 3)．そのため NE の新 IF 設計と同時に NMS の機能設計を行って仕様を合致させる必要があるが，設計段階で NMS の開発者が NE の仕様を確認できることは少ない．仮に仕様を合わせることができたとしても，実際にはリリース直後に実機による IF 確認試験が必要であり，その結果次第ではシステムのアダプタの設計からコーディングまでの作業を再度やり直す必要がある．

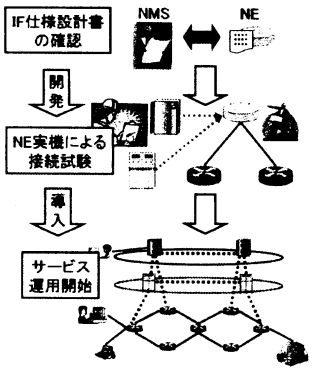

図 3　NMS の機能開発手順

またサービス開始後も，機能追加やバグフィックスによる NE の OS 更改，ネットワーク構成の変更といった要因で IF は頻繁に更改され，稼働中のシステムと整合をとれなくなることが少なくない．この場合もやはりリリースされた実機の IF 確認調査とアダプタの開発，アダプタ投入後のシステムとの接続試験が必要であり，その分新サービス導入が遅れることとなる．また，新規に導入する NE の IF を稼働中のシステムに対応させる場合には，NE に対応するアダプタの仕様設計，コーディングが必要となる．

以上のような状況を想定して先ほどの MDA の適用を検討すると，DEN-ng のような UML モデルから NE ごとのアダプタを生成するためには，まず命令を各 NE-IF が解釈できるコマンドへ変換するなどのルールが必要となる．MDA の例では図 2 におけるマッピングルールに実装されることとなる．NE ごとにマッピングルールを手作業で作成し試験するならば，開発者の作業負荷はアダプタの実装が変換ルールの実装に変わるというだけで抜本的な解決にならず，迅速なサービス導入には結びつくとは限らない．マッピングルール生成に関しては，実機試験を伴う何らかの自動化アプローチが必要になるだろう．

2.3　オペレーション機能記述の自動生成

DEN-ng などの UML モデルを利用する場合，その定義記述の作成時にも課題が存在する．これまでに述べたとおり，抽象度の高い共通情報モデルを構築することは上位管理層ユーザのオペレーション機能実装に利便性をもたらす．しかし，人手で記述したモデルが実際にユーザの目的を反映していること，NE に対する処理コードとして文法など動作要件を満たしていることを検証することは難しい．

一方異なる IF 仕様を持つ複数の NE への動作コードを先に用意し，その後共通のモデルへ変換するといったボトムアップアプローチならば，処理に沿った機能表現の抽象化がしやすい．この場合，開発者は対象となる複数の NE への処理の共通項を抽出してから UML などで機能モデルを定義する．しかし，このアプローチは先に NE ごとの動作コードの理解を必要とするため，本来の目的である容易な上位層オペレーション機能開発の実現と相反することとなる．また，NMS がいくつかの NE に対して既に機能を実装済みで，後から追加される NE 対応のアダプタを新たに実装するといったサービス運用中局面においては，モデル設計から作業を始めることは冗長である．このような上位層オペレーション機能の構築や既存アダプタの流用といった利用法の側面からは，具体的なアダプタ記述から共通の抽象記述に自動で変換する必要性が浮かんでくる．

3.　オペレーション機能群の “調合” とインターフェースの “触診”

3.1　今後のオペレーションシステム開発の要件

前章で述べた二つの課題から，今後のオペレーションシステム開発にはアダプタ記述などから上位層へ提供する共通的なオペレーション機能記述への変換 (具象→抽象) と，共通記述から各 NE で動作する特化したアダプタ記述への変換 (抽象→具象) の二点が重要になると考えられる．

従来の方法でモデルから実際の動作コード生成を行うためには，抽象から具象へ変換する緻密なマッピングルールが必要となるが，この場合，NE の OS アップデートによる仕様変更時には，合わせてマッピングルールを更改しなければならず，頻繁に OS の仕様変更がなされる状況ではマッピングルール作成に時間を取られ自動化のメリットが埋没する．さらに新規の NE 用のコード生成に関しては，既存のマッピングルールでは全く役に立たないことになる．

そこで我々は，オペレーション機能の開発を行う際に，マッピングルールを用いずに共通記述を特定の NE に対応したアダプタ記述へ変換するため，対象 NE の実機を試験環境に接続し，有効と考えられるコマンドの送信と応答の解析を繰り返すことで，実際に動作するアダプタ記述を自動で生成する IF “触診” 法を提案する．

IF 触診法では未知の NE から有効な応答を得る確率を高めるため，他ベンダのアダプタや既知のベンダ特性を既存のコマンドパターンとして蓄積し，共通記述を基にこれらの既存のコマンドパターンを組み合わせることでコマンドを多数自動生成し，試行錯誤を伴った試験を行う．仮に必要とする有効応答全てが得られた場合，変換ルールが存在しない場合でも NE に対する該当するオペレーション機能のアダプタを完全に自動生成することができる．また，試験環境にて NE の応答を求めながら作業を行なうため，アダプタの開発と実機試験を並行して進められるという利点がある．

一方，IF 触診法が機能するためには，試験環境にて可能性の

あるコマンドを多く生成し試行を繰り返すことが望ましい．そこで我々は機能モデルを一から作り出すだけではなく既存 NMS が持つアダプタ記述のパターンを利用し，組合わせや類似語，意味による変換を行って試行コマンドを多数生成する "調合" 法についても同時に検討を進めている．NMS が持つ既存のアダプタを調合することによって，ボトムアップアプローチにもかかわらず自動で機能の特徴が抽出されることとなるため，開発者の負担を軽減しながらも，より実用性の高い機能記述の生成が期待できる．図 4 に IF 触診法のイメージ図を示す．

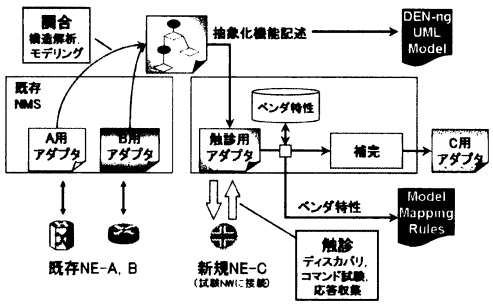

図 4　IF 触診法

調合結果は DEN-ng の UML モデルの構成情報として，触診により抽出するベンダ特性はマッピングルールとして展開することが考えられ，我々のアプローチは MDA によるオペレーション機能開発の自動化を補完することもできると考えている．

4. 機能要件

本章では，前章で述べた触診と調合を実現する IF 触診システムの機能要件について述べる．

4.1 機能記述の分解

本項では各 NE に対するアダプタのパターン解析による，オペレーション機能記述の分解機能について述べる．我々は機能記述の特定のパターンをシステムによって利用可能とするため，NMS が用いるアダプタ記述の論理的な解析を行った．

図 5 はアダプタの簡単な一例である．本アダプタは単体でも動作可能なスクリプト言語 Expect[5] で記述されている．Expect は応答の内容や状況に合わせた処理記述が可能のため，CLI による NE への処理や情報取得についてあらゆる機能の実装と表現に適している．

アダプタ記述は大抵の場合 NE に対して Telnet 接続し，CLI から複数のコマンドを送信し，応答から情報を取得し切断する構成で記述されている．また調査を行った結果，アダプタ記述が必ず以下の三つの部位に分けられる特徴を持っていることがわかった．

- 設定記述部：変数宣言や引数の定義設定を記述する部
- 接続記述部：接続・切断，ユーザ名やパスワード入力など，NE との接続に必須な処理を記述する部
- 命令記述部：情報の取得または設定など NE へのコマンドを記述する部

さらに接続記述部や命令記述部については，コマンド送信記述と該コマンドの応答解釈記述をセットにすることで論理的

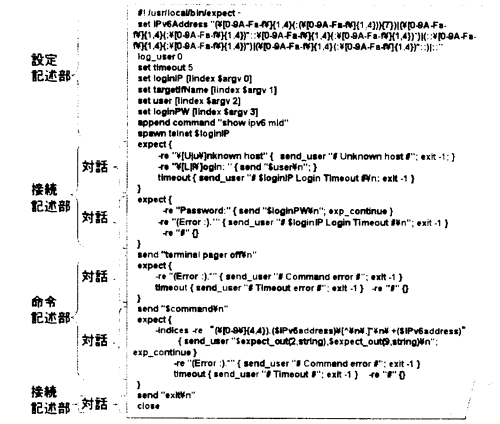

図 5　アダプタの記述例および分解例

な分割が可能であることがわかった．本稿ではコマンドと応答解釈の記述セットを対話と呼ぶこととし，以降この対話をオペレーション機能を表現する記述の最小粒度として検討を進めていく．

三つの記述部位，さらに対話単位での分解を自動的に行なうことで，既存アダプタ記述の再利用性が高まる．そこで，先に例示した Expect にて記述された複数のアダプタの分解を行う機能を試作したところ，与えられたキーワード (予約語は Expect に依存) を使うことで試した全てのアダプタを目的の粒度に分解することに成功した (図 6)．キーワードの変更は必要となるが，Expect 以外のスクリプト言語で記述された他のアダプタ記述についても，特別なルールを用いることなく分解可能である．

4.2 機能記述のモデリング

分解した対話や記述部を触診システムの DB に格納する際に，その後の触診用のアダプタ記述生成や共通記述生成に用いるための属性を付与する機能をモデリングと呼んでいる．我々は属性に，分解元のアダプタ記述の機能名や動作対象 NE のベンダ名やプロダクト名を用いることとした．本研究ではアダプタ記述を対話単位で分解していることから対話記述に関する属性も付与したが，我々はその属性として NE に送信されるコマンドの文字列を利用することとし，自動的に属性としてコマンドを別に格納している (図 6)．

コマンドの自動抽出についても Expect の予約語をシステムに用意することで実現可能だった．また，他の対話への参照関係についても自動抽出してプロセス情報として付与することとし，対話関係の検索やアダプタ記述の再構築に利用する．

4.3 機能記述の調合

指定した対象 NE に触診を行うための試験用機能記述を生成する機能である．格納したコマンドパターンを知識として最大限活かすため，コマンド文字列に対して様々な処理を加えて触診用の試行コマンドを生成し，そのコマンドを含めたアダプタを再構成する．現在は以下の整形処理を行うことで，コマンドを生成している．

- 他 NE アダプタが用いたコマンド（直接利用）
- 他 NE アダプタのコマンドを構成する単語順序入替え

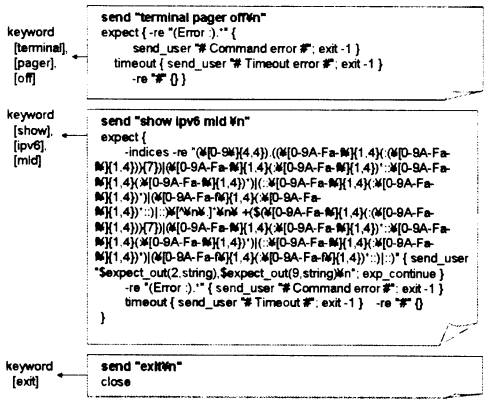

図 6　コマンドの抽出

- 特定文字列を類似語に変換
- 無効パターン排除
- 特定の NE に依存したコマンド

これらの処理は処理の意味が同じにも関わらず命令のために必要なコマンド文字列が異なる特徴を持つ，マルチベンダ NE 間のコマンド仕様の違いを吸収することを考慮している．特定の NE に依存したコマンド生成処理については次節にて説明する．

4.4　NE パターン処理

アダプタの対話記述には，対象としている NE のみに有効な，NE 依存の記述が存在する．例えば接続記述部のログイン処理は機能に関わらず NE のベンダに等しく固定である．このような記述については NE 特有のパターンとして別途記録することとし，以後の処理に優先的に用いることとする．パターンを用いるため対象 NE と接続した直後に NE のベンダやプロダクトを特定する必要があるが，その方法として，MIB-II の sysDesc を参照するディスカバリを行うのが一般的である．他にも調合によって生成した大量のコマンド群のなかで，全く可能性のないコマンドを削除するなどの処理ルールについてもベンダ依存の要素が大きいため，NE パターン機能により処理する仕様とした．

4.5　有効回答抽出

後述の IF 触診機能は，未知の IF に対して様々なコマンド送信を試行し，応答のなかから有効となる情報を探索する機能である．そのためには，実際に正解となる模範回答が必要である．模範回答は可能ならば収集の手間をなくすことが望ましい．そこで本システムでは，既存の NE 用アダプタを用いて有効な回答の抽出を行う機能を用意している．抽出例を図 7 に示す．

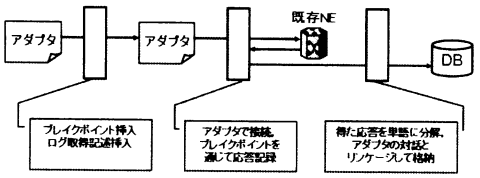

図 7　有効回答の抽出

本システムでは，抽出を自動化するために既存のアダプタの分解した対話ごとにブレイクポイントを挿入することで応答を取得している．応答文字列のなかから，同アダプタ記述内の対話以降で用いられている場合のみ，その文字列を有効として格納することとした．また，自動化には応答が無効であるとする判断も必要であるため，ベンダごとに定義されているエラー応答をパターンとして取り込んでいる．

4.6　IF 触診（試験）

IF 触診は，有効回答および無効回答をパターンとして得た後に，調合によって生成されたコマンドを送信し，回答を取得，有効回答を判定して対象 NE に対するアダプタを生成する機能である．NE の応答文字列と有効回答パターンを比較し，同一の文字列を検出することで，有効と判定する．このとき一つの有効回答を見つけるために複数のコマンド送信を行うことから，触診時にはコマンド送信後に変更された設定のロールバックや，設定ファイルのバックアップが必須の機能となる．

以上の要件を満たす，IF 触診システムの機能ブロック構成を図 8 に示す．

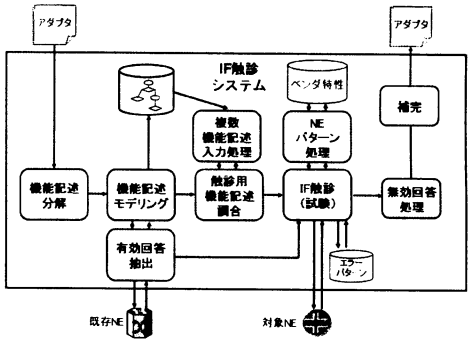

図 8　IF 触診システム

5.　検 証 実 験

本章では，提案した IF 触診法および IF 触診システムについて，その実現性の検証を行ったので報告する．

5.1　検 証 実 験

実験は，既存 NMS のアダプタから調合を用いて試験ネットワークに接続した実機 NE を触診した．実験のために，設計した機能から分解，モデリング，有効応答取得，試験用コマンド生成，IF 触診機能部を試作しており，対象 NE としてベンダの異なる 3 種の NE を，アダプタとして各 NE それぞれ 3 種類の機能記述 (Expect) 計 9 種類を用いている．

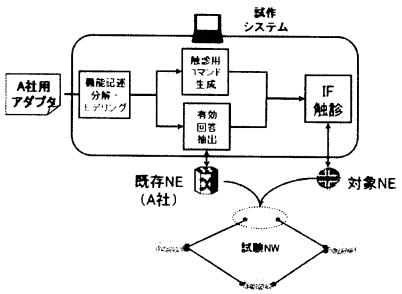

図 9　検証実験の概要

各 NE に対して触診を行うにあたり，他社のアダプタ記述および，機能に依存しない共通の記述については既知のパターンとしてシステムが蓄えているものとした．一方，対象 NE が実際に必要とするコマンドについてはパターンとして持たせることなく，触診によって探索できるかどうかを評価した．なお，実際にアクティブなネットワーク上でなければ NMS 機能の一部情報は取得できないため，実験を行うにあたり試験用ネットワークを構築して行っている．

5.2 実験結果と考察

実験結果について述べる．分解およびモデリングについては実験に用いたアダプタ全てで成功した．この結果は CLI 上で行われる全ての処理が一定のルールで論理的に解析できることを示していると考える．既存アダプタの解析においても，ブレイクポイントの挿入およびオペレーション機能が期待する有効応答の取得全てが成功した．

次に触診によるアダプタ再構成実験の結果について示す．

		1回目							2回目			
対象 NE	機種	触診開始コマンド数	再構成に必要な応答数	ベンダ固有の対話数	触診が必要な対話数	各特記事項	禁止入力	再構成率	各特記事項	禁止入力	再構成率
A	機種1	2741	5	3	2	1	1	1/2	4/5	2	0	2/2	5/5
	機種2	4757	4	2	2	1	1	1/2	3/4	2	0	2/2	4/4
	機種3	389	4	2	2	1	1	1/2	2/4	2	0	2/2	4/4
	小計	7887	13	7	6	3	3	3/6	10/13	6	0	6/6	13/13
B	機種1	418	6	4	2	1	1	1/2	5/6	2	0	2/2	6/6
	機種2	130	7	5	2	1	1	1/2	6/7	2	0	2/2	7/7
	機種3	82	6	4	2	1	1	1/2	5/6	2	0	2/2	6/6
	小計	630	19	13	6	3	3	3/6	16/19	6	0	6/6	19/19
C	機種1	5770	4	2	2	1	1	1/2	3/4	2	0	2/2	4/4
	機種2	298	5	3	2	1	1	1/2	4/5	2	0	2/2	5/5
	機種3	58	3	2	1	1	0	1/1	3/3	1	0	1/1	3/3
	小計	6126	12	7	5	3	2	3/5	10/12	5	0	5/5	12/12
合計		14643	44	27(61%)	17(39%)	9	8	9/17(53%)	36/44(82%)	9	8	17/17(100%)	44/44(100%)

図 10 触診によるアダプタ再構成実験結果

図 10 は各 NE 用に新たに開発するオペレーション機能を，NE 固有パターンの利用と，他 NE から得られるコマンドパターンによる触診でどれだけ再構成できるかを示した表である．

一度目の実験結果では，82 %以上の高い割合で有効応答を発見した．残りの検出失敗について解析した結果，現在のコマンドパターン定義の改良により検出できる見込みを得たため，二度目の実験では対象とした全ての有効応答抽出に成功した (100 %)．

再構成に必要な回答数はアダプタ分解結果であり，再構成を行う際に記述すべきコマンド数と解釈できる．今回の実験に用いたアダプタでは，ベンダ固有のパターンとして検出した対話数は全体の 60 %となり，残り 40 %が対象 NE について全く知識がない，触診などの別機能がなければ抽出できない対話だった．ベンダ固有のパターンと比べると少ないが，これは今回用いたアダプタで検出した全体の対話数が少なく，固定的に存在するベンダ固有パターンの率が相対的に多くなっているためで，多くの対話を必要とするアダプタにおいては触診が必要な対話の割合が増えていくと考えている．触診用に生成したコマンド数は正解一つに対して約 58〜5800 と大きくばらついているが，これは生成すべきコマンドが必要とする単語数や，パターンで得られる単語数が機能によって異なるためである．コマンド数は他 NE 用のアダプタ数が多いほど数が増えることになり効果があがるが，NE に対してありえないパターンのコマンドを削除することで触診の効率を高めることも必要となる．なお自動で触診を行うため，今回の実験では我々開発者の待ち時間に対するストレスはほとんどなかった．

6. おわりに

本稿では，NMS のオペレーション機能自動化を行うための機能要素として，IF 触診方法およびオペレーション機能記述の調合方法を提案した．また，提案した触診と調合について，その実現性を検証するため試作プログラムを用いて検証実験を行った．

触診については，現在の手法により未対応の NE からでも，高い割合で有効回答を取得できることを確認した．しかし現在の機能だけでは，全ての有効回答を得た場合でもアダプタを自動生成できるわけではないため，エラー処理や得た情報の文字列整形処理の補完といった機能によるソリューション化が必要となる．また，今回の実験で用いた NMS のアダプタは全て，コマンドに対して有効な応答を返信することを利用している．従って設定コマンドを含んだ対話の場合にはプロンプト確認など異なる処理手順が必要である．また未知のコマンドを自動で抽出するために，複数のベンダ NE で共通利用できるパターンを検出し，利用可能であることが証明できた．これは各々独自の IF 仕様でありながら，同様の機能を提供するために類似性を持つ実装に自然と収斂するためで，この特性を利用すれば未知の NE に対するオペレーション機能開発の自動化に実現性を見出せる．今後は，より定量的な結論を得るために，ベンダに固有のパターンを充実させたり，固有のパターンを可能な限り抑えるなど，様々な状況下で検証実験を重ねることが重要となる．

調合については，EML の差異を吸収し自動で共通記述を生成することができた．従って，目的の一つであるビジネスロジックから直接ネットワークのオペレーション機能を制御する次世代オペレーションの実現に一歩近づいたと考えている．今後は共通記述からポリシーなど上位管理層が用いるための記述への自動変換を，具体的なオペレーション機能を用いて検証していく．また，マッピングルールの自動生成や実機 NE の利用可能コマンドの全検出といった，他の用途への技術展開を進めていく．

文　献

[1] ITU-T Rec. M.3010, "Principles for a telecommunications management network," Feb. 2000.

[2] M. Ejiri and F. Birch : "ENABLING SECURE EBUSINESS BASED TELECOMMUNICATIONS MANAGEMENT a new Paradigm beyond TMN," Vol.1. ICETE Proceedings. Aug. 2004.

[3] J. Strassner : "A MODEL DRIVEN ARCHITECTURE FOR TELECOMMUNICATIONS SYSTEMS USING DEN-NG," Vol.1. ICETE Proceedings. Aug. 2004.

[4] J. Strassner : "Policy Based Management Thoughts and Observations from a Network Management Perspective," Policy Workshop 2004. http://www.policy-workshop.org/2004/slides/Strassner-Panel.pdf. Jun. 2004.

[5] The Expect Home Page : http://expect.nist.gov/

[6] 宮内 直人，登内 敏夫，高野 誠："管理システム開発技術，" 電子情報通信学会誌 Vol.87, No.12. 2004. pp.1036-1041, Dec. 2004.