マルチエージェントを用いた自律組織間診断システム：ENCORE

明石 修[†,☆] 菅原俊治[†,☆] 村上健一郎[†,☆]
丸山充[†,☆] 高橋直久[†,☆]

インターネットの安定運用を実現するためには，自律システム（AS）がインターネットに広報した経路情報の振舞いを理解し，自ASの意図が正しく反映されて伝わっていることを検証することは重要である．この機能を実現するため，他のAS内の環境において要求元ASに関する観測を行い，その結果を解析して返送するための協調動作の枠組みを示すリフレクタモデルを定義し，それを基本動作として自らの持つ診断知識に基づいて自律的に動作する，マルチエージェントによるAS間障害診断システムENCOREを提案する．本システムは，各ASに観測機能を持ったエージェントを配置し，エージェントが自らの持つ診断知識に基づいて他のAS中のエージェントと協調することにより，ネットワークの障害を解析する．

ENCORE: An Inter-AS Diagnostic System Using Cooperative Distributed Agents

Osamu Akashi,[†,☆] Toshiharu Sugawara,[†,☆]
Ken-ichiro Murakami,[†,☆] Mitsuru Maruyama[†,☆]
and Naohisa Takahashi[†,☆]

For reliable operation of the Internet, it is crucial that each autonomous system (AS) understands the implications of the routing information that it advertises, and verifies that its intentions are correctly propagated by viewing the information about itself from the point of view of other ASs. This paper proposes a network diagnostic system, ENCORE. It is based on the reflector model where intelligent agents in ASs cooperatively infer the dynamics of the Internet and analyze network faults. In ENCORE, an agent in an AS monitors routing information in its environment and shares the information with agents in other ASs. The agents then perform collective analysis based on diagnostic knowledge to identify problems.

1. はじめに

多くの自律システム（Autonomous System: AS）を相互接続するインターネットの安定運用を実現するためには，自ASが広報（アドバタイズ）した経路情報が，正しく意図を反映して全世界に伝わっていることを検証することは重要である．しかしながら，今日のインターネットではAS間経路制御は不安定であり[1),2)]，障害の早期発見や原因の特定は困難である．それは，インターネットが多くのASから構成されることによる複雑さに加えて，経路情報が空間的，時間的に変化するためである．

本研究の目的は，インターネットの経路情報の空間的，時間的な変化を観測することによりその振舞いを理解し，自ASの経路が正しく意図を反映して伝搬していることを検証し，障害があれば，それを検知，解析することが可能なAS間診断システムを提案することである．

経路情報の振舞いを理解するには，他のASにおいて経路情報を定常観測し，その解析を行う必要がある．また障害原因解析のためには，エキスパートの診断知識をシステムが持つ必要がある．従来のWEBで提供されている観測ツールは，その解析にネットワークに関するエキスパートの知識を前提としており，またトラフィックの観点から定常観測には問題がある．

本稿では，上記問題を解決するため，他のASからの視点で情報を収集するリフレクタモデル[3)]を定義する（2章）．またそれに基づいて動作するシステムを構築する際の要求条件をまとめ（3章），以下のような特徴を持つマルチエージェントによるAS間障害診断シ

† NTT 光ネットワークシステム研究所
NTT Optical Network Systems Laboratories
☆ 現在，NTT 未来ねっと研究所
Presently with NTT Network Innovation Laboratories

ステム **ENCORE**（Inter-AS Diagnostic Ensemble System using Cooperative Reflector Agents）[4),5)]を提案し，そのシステムにおける解決法を示す（4章）．最後に，実験環境下で動作確認を行った，エージェントの協調による診断の効果と有効性，および実行系の高速化について議論する（5章）．

(1) 観測・診断方法

診断知識と観測機能を持ったエージェントを各ASに配置し，定常観測は自己の知識に基づきローカルに行う．エージェントが障害発生の可能性を検知した場合，必要に応じて他のエージェントと観測結果を交換し，また新たな観測を依頼することにより，協調して障害の解析を進める．他のエージェントとの協調方法は，エージェントが自分自身の状態を映す鏡として動作するリフレクタモデルに基づく．

(2) 知識記述

システムの内部処理とは独立に，ネットワーク障害を解析するための知識をエージェントに与えるため，観測戦略と仮説検証フレームワークを記述する枠組みを提供する．また観測を通じて取得した値を，変数を通じて知識に反映することにより，外部環境であるAS固有の状況や時間的な変動を扱う枠組みを与える．

(3) 実行系の高速化

タビュレーション技法[6)]を用いた実行機構を提供し，過去の実行履歴に基づいた適応型の実行動作制御を行うことにより，値取得手続きの効率化を行う．

2. リフレクタモデル

2.1 AS間経路制御の現状

多くのASを相互接続するインターネットでは，BGP（Border Gateway Protocol）[7)]と呼ばれるプロトコルを広く用いて，AS間で経路情報を交換する．各ASは，自ASに関する経路情報を，BGP peerとなる隣接するASに広報する．経路情報は，順次各ASのポリシーに従って取捨選択されながら，インターネット全体に伝搬する．しかしながら今日のインターネットでは経路情報は不安定であり[1),2)]，障害の早期発見や原因の特定は困難である．それは，インターネットが多くのASから構成されることによる複雑さに加えて，以下のような経路情報制御自体が持つ性質が，その観測を困難にしている．

- あるASが広報した経路情報は，各ASにおいて各々のポリシーに基づき，経路選択などの情報の加工が行われ伝搬される．そのため，同じASから広報した情報が，伝搬経路上のASごとに空間的に変化する．
- 経路の選択と広報には，静的な選択規則のみでなく，動的なパラメータである物理的な回線やルータの障害，他のプロトコルとの相互作用が影響するため，経路情報は時間的に変動する．

これらの性質から，インターネットの経路情報の空間的・時間的な変化を把握し，自ASの広報した経路が意図を正しく反映していることを検証するためには，他のASに自ASに関する経路情報がどのように伝えられたのかを観測する必要がある．

2.2 リフレクタモデルと適用例

経路情報の空間的・時間的な変化を観測するため，各 AS_i ごとにエージェント R_i を配置し，自ASの視点から観測した他のASに関する様々な情報を，他のAS中のエージェントと交換する．R_i は AS_i で観測した情報に加えて，R_j $(j \neq i)$ から AS_j での AS_i に関する情報を得ることにより，AS_i に関する経路情報の変化を観測することが可能となる．このとき R_j は，AS_i の状態を映す鏡（リフレクタ）として動作する．このようなエージェント間の協調関係に基づく動作の枠組みを，リフレクタモデルと呼ぶ．

リフレクタとは，エージェント間の種々の協調形態の中で，特に要求元のエージェントに関する情報を，依頼を受けた側のエージェントの存在する環境において収集し，その解析結果を送り返す機能をいう．すなわち系全体を構成する各要素が発した情報が，系全体に変化しながら伝搬し，分散して存在する経路情報に対応するため，地理的に分散したエージェント間でお互いに相手に関する情報を観測する形態の協調を用いることにより，このような性質を持つ情報を観測することが可能となる．各エージェントの構造として，そのAS内のボーダルータへのアクセス権，ネットワーク診断ツールの実行権を持つことによるモニタ機構と，対話機能を通じて取得した情報を他のエージェントと交換するための機構が，分別して定義されている必要がある．

また経路情報の障害は，経路情報を広報した AS_i よりも，経路情報を受けとる他の AS_j $(j \neq i)$ の方が，その問題に早期に気付きやすい．それは伝搬経路における経路情報の空間的・時間的変動は，広報した側（AS_i）では観測が困難であり，伝搬経路である AS_j でのみ観測可能だからである．本モデルでは，R_j $(j \neq i)$ は AS_i に関する障害を検知した場合，エージェント R_i に通知し障害の発生を伝える．

図1に，リフレクタモデルの適用例を示す．この例では，AS_1，AS_2，AS_3 はIPレベルでトランジットす

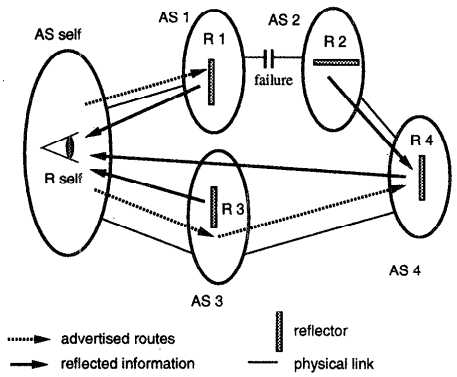


図1 リフレクタモデルの例
Fig. 1 Example of reflector model.

るASであり，AS_{self} が広報した経路情報は，AS_1，AS_2，あるいは AS_3 を通じて，AS_4 に到達している．

この例では，AS_1 と AS_2 の間で AS_{self} に関するフィルタ設定誤りを起こし，AS_{self} に関する経路情報が，AS_2 へ流れなくなった．これを従来の方法で，AS_{self} から診断ツールを使って調べても，AS_1 から先の接続が切れているという事実以上の解析は不可能である．すなわち，AS_1 と AS_2 の間の障害なのか，あるいは AS_2 中においてルータの静的経路設定の誤りがあるのか，他のASが誤って流した経路情報に起因する間違った経路情報が存在するための障害なのか，判定できない．

リフレクタモデルでは，AS_{self} 中のエージェント R_{self} は，R_1 と R_2 に自分が広報した経路情報などを送り返してもらう．その結果，AS_1 では AS_{self} に関する経路情報は意図どおりである．しかし AS_2 では，AS_1 に関する経路情報は存在するが，AS_{self} に関する経路情報は存在しないことが判明する．以上のことから，R_{self} は AS_1 と AS_2 の間に，AS_{self} に関するフィルタ設定誤りという障害が存在することを推論できるようになる．

3. システム構成上の要求条件

本章では，リフレクタモデルに基づく診断システムを実装するにあたり，システム構成上の要求条件を整理する．

エージェント間の協調は，依頼された処理を遠隔手続き呼び出しの手法に基づいて実行し，情報を送り返す単なる鏡としての動作ではない．情報の取得手段のプランニングや観測動作を，自身の持つ知識に基づいて自律的に決定し，動作する知的な鏡として動作する．すなわち，他のエージェントからのデータの要求に対しては，独自の知識に基づいて観測・診断動作をプランニングして必要な情報を収集し，必要に応じて統計処理や他のエージェントとのさらなる協調動作も行い，求められた結果を返す．

これらの動作を実現するためには，ネットワークに関するエキスパートの知識が必要である．この知識は，障害原因を解析するための診断に関する知識と，時間に基づいて行う観測動作に関する知識を含むことが必要である．診断知識は，観測した事象から可能性のある原因を推論する仮説と，その仮説を検証するためにルールを含まねばならない．これらの知識は，新たな障害事例の追加や，AS固有の状況に応じて柔軟に対応できるように，追加・変更が容易である必要がある．診断知識と観測動作は，明確化のため，独立した動作として記述する必要があるが，その相互作用を考慮したプランニングをする必要もある．

個々の知識の記述においては，経路数などの動的に変化する値や，観測を通じてのみ得られるAS固有のローカルな値を，知識の中に表現する必要がある．一方，診断のために必要な，ICMP（Internet Control Message Protocol）[8]を使った `ping` や `traceroute` などの診断ツールを用いた解析は，エージェントの配置されるASごとのツールの違いから影響を受けないようにするため，ツールの実体とその値の解釈部分を，診断知識とは独立にする必要がある．

実行系に関しては，診断を高速に実行するため，あるいは監視するシステムが，監視対象にかける負荷を減らし，影響を極小化するという観点から，観測負荷の高い観測や診断は効率的に実行することが必要である．また知識に基づいた様々な診断ルールをプランニングする場合，そのコストやあらかじめ与えた可能性のみでなく，過去の実行履歴に基づきローカルな環境に適応するように，フィードバックする機能も必要である．

4. AS間診断システム ENCORE

本章では，リフレクタモデルに基づいて動作する，ENCOREのシステム構成と知識処理アーキテクチャに関して述べ，要求条件に対する解決法を示す．

4.1 システム構成

ENCOREのシステム構成を図2に示す．ENCOREのエージェントは，ヘッダ部とボディ部に分割して構成し，両者で変数を共有することにより，実装依存部を分離し，環境の違いと時間的変化をパラメータ化する．

ヘッダ部は，診断・観測のための知識処理機能から成り，ボディ部は経路情報の観測や診断のためのツー

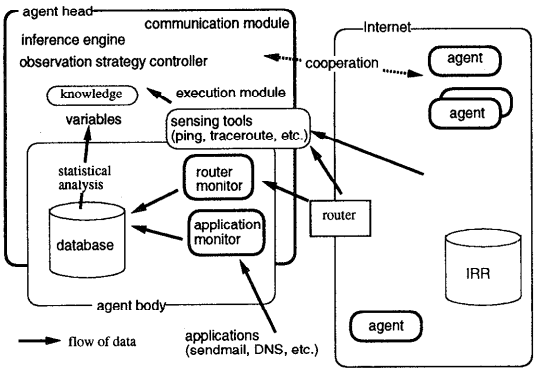


図2 システムの構造
Fig. 2 System structure.

ル，データベースなどから構成する．ヘッダ部とボディ部を分割することにより，特定のツールや観測手段の実装，得られる統計データといった環境独自の値と，共通な知識を分離し，ヘッダ部を様々な環境で共通に用いることができる．ボディ部は，共通化したアプリケーションインタフェース層を通すことにより，そのASごとに独自のツール，実装を用いることができる．

ヘッダ部は，以下の部分から成る．

- 推論エンジン ··· 観測された事象から仮説を取り出し，順次その検証のためのルールをプランニングする
- 観測制御エンジン ··· 観測動作の実行と，その観測戦略の選択・切換えを指示する
- 診断知識，観測動作戦略の記述
- 会話モジュール
- 実行モジュール ··· ボディ部のツールを起動し，その結果を知識として蓄える

他のASにおける観測や，診断ツールの使用による解析が必要な場合は，ヘッダ部でその依頼内容を決定する．次にその時点で協力して動作可能なエージェントを捜し，会話モジュールのエージェント間通信機能を用いて，仕事を依頼する．集められた結果に基づく判断や，さらなる診断動作プランニングもヘッダ部が行う．

ボディ部は，以下の要素から構成する．

- 診断ツール···ICMP を使ったツール，IRR (Internet Routing Registry)[9]データベースへの問合せ機能など．
- 観測ツール
  - ルータモニタ···BGP の経路情報の変動を監視し，統計処理したデータを提供．SNMP (Simple Network Management Protocol)[10]やルータにログインする機能を用いる．
  - アプリケーションモニタ···DNS (Domain Name System)[11]や sendmail の出力するログデータを監視．
- データベース
  - 観測の結果得られた情報
  - 環境固有の静的情報···システム立ち上げ時に読み込む，監視対象のボーダルータのアドレスや，そのアクセスパスワード

知識は，4.2.2 項に示すように，システムの実行系とは別途記述する．しかし環境ごとに異なる情報や，観測結果を統計処理し，時間的に変動する値を知識に反映するための変数を導入する．この変数の値を生成するため，ENCOREでは，値の取得手段，統計処理関数，閾値の設定，閾値を超えた場合の処理を定義するデータベースオブジェクトを提供する．このオブジェクトは，任意のルールから名前で参照可能である．

観測するデータの例として，そのASで観測される全経路数がある．変数の値は動的に変わるパラメータであり，ASごとの環境により，また時間の経過とともに変化するため，あらかじめ値を与えることができない．したがって，エージェントの存在する環境で定常的に観測し，変数参照時にその値を決定する機構が必要である．

4.2 ENCOREにおける知識処理

本節では，システム実行系とは独立した知識記述と，それに基づいた動作を実現するための，ENCOREにおける知識処理アーキテクチャについて述べる．

4.2.1 アーキテクチャの概要

エージェントのヘッダ部における知識処理アーキテクチャを図3に示す．エージェントの知識処理部は，診断知識，観測戦略を記述した静的データと，環境から取得した動的データを知識として持つ．エージェントはこの知識に基づいて，自律的に行動する．エージェント宛てのメッセージは，event monitorを通じて受け付ける．

Plannerは，検証動作をすることが可能な複数の仮説の中から，意図 (intention) に基づき次に検証する仮説の集合を選択し，実行するルールの部分集合を決定する[12]．この場合の意図とは，plannerが仮説を選択する場合の，コストや過去の履歴などに基づく動作基準を指す．Schedulerは，最終的にルールの集合を直列化し，executerが順次実行する．Evaluatorは，その結果を実行履歴データとして知識に反映する．他のエージェントとの協調は，coordinatorが仲介する．

4.2.2 知識記述

ENCOREでは，システムの実行処理とは独立に

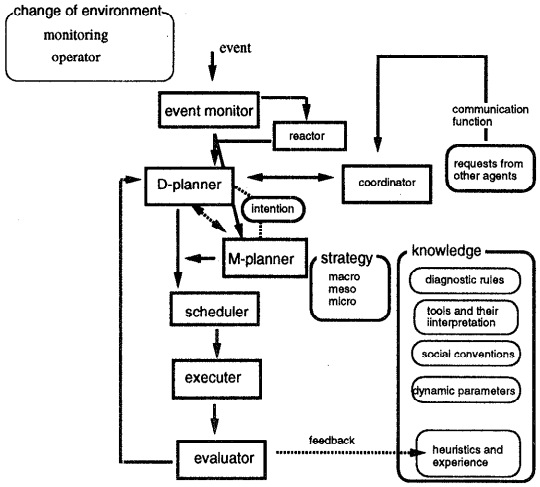


図3 知識処理アーキテクチャ
Fig. 3 Architecture for knowledge processing.

診断知識をエージェントに与えるため，図4に示すように仮説（**hypothesis**）と検証ルール（**rule**）を用いて宣言的に記述する枠組みを与える．観測戦略（**strategy**）は，実行するルール（**rule**）と実行頻度を記述する．

仮説は，その仮説が成り立つために必要な検証ルールを含み，仮説が真であるためには，各ルールの実行結果がすべて真である必要がある．仮説はさらに詳細な仮説を含み，入れ子構造を成す．また**trigger-event**として，イベント名のリストを記述し，そのイベントが通知された場合に，成り立つ可能性のある仮説であるとして検証を行う．仮説は**priority**値を属性として持ち，その初期値を記述する．この値は，実行履歴に基づき変更される（4.3節）．可能性のある仮説が複数存在する場合，その**priority**値に基づき，検証する順番を決定する．

検証のために用いるルールは，検証に必要なデータを取得するための手続き（**acquire-proc**）と，その結果を解釈し評価する手続き（**eval-proc**）の記述から成る．それぞれの手続き中では，呼び出す関数名とその引数を記述する．ルールは，ローカルに実行可能な手続きか，あるいは他のエージェントに依頼するかを示す**type**属性を記述する．なお，**eval-proc**中では，新たな診断・観測動作が起動可能である．

また同じゴールに対して実行される各ルール間でデータを共有し，不必要なシステム外部への呼び出しを最適化するため，ゴール内でのみ共通な値を持つことを保証する特殊変数を提供する．

観測戦略（**strategy**）の記述では，ルールの中で，4.1節で述べたデータベースオブジェクトの更新処理

```
goal ::= {hypothesis}+ | {strategy}+
hypothesis ::= {hypothesis}* | {rule}+
        [| trigger-event] [| priority]
strategy ::= {rule}+ | time-spec
rule ::= acquire-proc | eval-proc [| type]
```

図4 知識記述文法
Fig. 4 Syntax of knowledge description.

を実行する．

4.2.3 プランニング

ENCOREでは，診断知識と観測動作という，密接に関連するが，異なる2つの知識に基づいた動作をプランニングするため，plannerを，診断planner（D-planner）と，観測動作planner（M-planner）と呼ぶ2つの協調型plannerから構成する．

診断順序は，D-plannerが，半順序で記述された仮説のツリーを取り出し，**priority**値を参照することにより，順次検証する仮説の集合を決定し，その検証に必要なルールの集合を決定する．ルールが実行されることにより真偽が確定すると，D-plannerは可能性を絞り込み，次に検証する仮説の集合を選択する．

他のエージェントとの協調が必要なルールが存在する場合は，coordinatorモジュールが仲介し，他のエージェントを選択し，要求メッセージを送信する．Coordinatorを経由する理由は，協調して診断するエージェントを捜し，組織化する部分を知識と独立させ，他の組織化機構やプロトコルを独立して組み込み可能とするためである．現在組織化機構は，あらかじめトポロジ的に自ASの外側に存在するエージェントの候補を静的に記述しておき，その中から到達可能なエージェントを選択する方式を組み込んで用いている．もし相手エージェントからの返事がない場合，あるいは拒否メッセージが帰ってきた場合は，別のエージェントに依頼する．同時にその結果は，該当エージェントに対する知識としてcoordinatorにより知識として蓄えられ，次回の組織化のとき，協調相手の選択優先順位として反映する．

メッセージはUDP（User Datagram Protocol）/IP上に実装する，メッセージの種類は相手に要求する**request**と，その返事の**reply**，相手に情報を通知するための**inform**から成る．要求を受けたエージェントが，要求された処理を実行する手段を持たない場合や，負荷の観点から実行しない場合，**reply**メッセージに**reject**サブタイプを用いて通知する．

M-plannerは，タイマにより起動される動作を扱い，記述された観測動作，あるいはD-plannerからの要求

に基づき，記述された時間間隔に従って，観測のためのルールをプランニングする．

2つのplannerは，その実行結果により，相互に影響を及ぼしあう．すなわち，定期観測によりある異常の可能性のある事象を観測した場合は，M-plannerはD-plannerに観測状況をパラメータとして渡し，診断を依頼する．逆に，ある別のイベントで実行された診断の結果，D-plannerがより細かいレベルでの統計観測データをとることを必要と判断した場合，自エージェントの，あるいは他のエージェントのM-plannerに動的に定常観測を依頼し，その結果を受けとる．

4.3 ルール実行の高速化

診断を高速に実行し，監視するシステムが監視対象にかける負荷を減らすため，D-planner/M-plannerからの出力であるルールの集合の実行に際して，タビュレーション技法[6]を用いた実行機構を提供し，適応型の実行動作制御を行う．

すなわち，executerは，順次引数に値を束縛しルール中の関数を実行するが，その呼び出し仕様と結果値の組から成る表を作成しておく．表にデータを記入した後に同様の呼び出し要求が来た場合，実際には関数呼び出しを行わずに，表を参照し結果値を返す．ルール中の値取得関数は，外部のルータや他のエージェントへの問合せを行うコストの高い関数であること，その結果得られる値への参照が仮説検証における多くのルールに共通に含まれることから，特にENCOREのような診断処理システムには有効である．たとえば，自ASのルータへの問合せによりBGPデータのサマリーを取得し，その結果得られるBGP peerのアドレス，状態，全BGPエントリ数などは多くのルールにおいて参照される値であり，その有効利用は重要である．タビュレーションで用いる表の値は，有効時間を設定し，executerが値の消去処理を行う．

Evaluatorは，すべての検証規則が真になった仮説は`priority`値を上げることにより過去の実行履歴を反映する．これをplannerが次回の診断時に検証する仮説の集合を選択する際に参照し，`priority`値が高い仮説を優先的に選択して検証を先に進めることにより，過去の実行履歴を反映するようにフィードバックする．

4.4 実験環境

エージェントの知識処理部分は主にCommon Lisp（約6[Kline]），コミュニケーションとツールのテキスト処理部分は主にCとPerl（それぞれ約800[line]）を用いて，UNIX上で実装を行った．本システムでは，筆者らが管理する実験ネットワーク（独自のAS番号を持ち，海外および日本国内から専用線を経由して2つのASからフルルートをBGPで受け，インターネットにマルチホーム接続を行う）で観測された症例と，その解析経験を基に，これまでに約60の仮説を作成して組み込んだ．

診断動作実験は，実環境の変動するデータを観測するため，自AS内でフルルートをinternal BGPで受けるルータ（Cisco4500），および外部ASに設置してあるルータ（Cisco4000）を用いて，実ネットワーク上で観測を行い，エージェントはSun SparcStation20（メモリ98 Mbyte）上で動作させた．この外部ASとは1.5Mの専用線を介して接続する．しかしながら実環境では重大な障害を起こした診断実験は行えないため，別途実環境をシミュレートする環境も用意し，双方で実験を行った．シミュレーション環境では，3台のルータ（Cisco4500）と，5台のFreeBSDマシン（Pentium MMX200 MHz/メモリ64 Mbyte）上でGateDを用いてBGP接続を行い，マルチホーム接続を行う自ASと，4つの他のASから成る環境を構築した．各ASは，イーサネットで接続する．実環境で実際に観測された症例をシミュレーション環境で再現させて，システムに組み込まれたルールが正しく動作することを確認した．

5. 考　　察

本章では，リフレクタモデルに基づく診断と，ENCOREシステムに関して議論する．動作環境は，4.4節で述べたとおりである．

5.1 協調による診断機能

本節では，実際に観測された障害に対する本システムの診断動作手順を例に，本システムにおけるエージェントの協調の効果と有効性について考察する．この例では図5に示すように，AS_x と AS_y はBGP peerであり，AS_y からアドバタイズされた経路は AS_z に伝わっている．この障害は，AS_y は AS_x に通知しないで，突然自AS番号を変えてしまったことにより発生した．AS_x では，突如複数のASへの経路が変更になり，特に AS_y 経由しか経路の存在しないASへの接続性が失われる状況となった．この場合の診断動作は，以下のとおりである．

(1) AS_x 中のエージェント R_x は，定常観測の中で，全経路エントリ数の突然の減少を観測する．ルータからBGPのセッションの状態を取得することにより，あるいは該当データのないルータの場合は，過去のASごとの経路数の記録と比較することにより，減少した経路はある特定のAS（$= AS_y$）からの経路

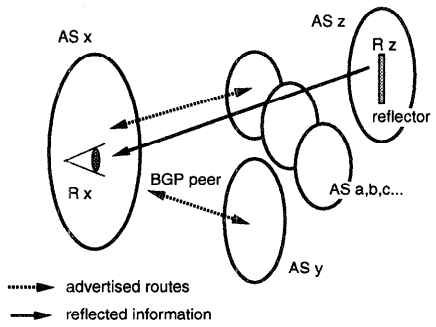


図5 診断動作例
Fig. 5 Example of diagnosis.

であることが判明する．M-plannerは診断依頼メッセージを，R_x のD-plannerに送る．

(2) R_x のD-plannerは，仮説リストから，可能性のある複数の仮説を取りだし，順次検証するためのルールをプランニングする．このとき，最初に AS_y のボーダルータへの到着性を検証するが，AS_x と AS_y の間のIPレベルの障害は検出されない．また，AS_x において，最近ルータの設定が変えられていないことも確認される．

この例では，AS_y にはエージェントは存在しない，として説明をする．もし存在した場合は，このエージェントとの協調により，仮説の検証はより容易になる．

(3) 自AS内の情報のみでは，もはやこれ以上の解析は不可能であるため，R_x は他のAS中のエージェント R_z と協調して，診断を行う．具体的には，AS_x 中のアドレス E_x，AS_y 中のアドレス E_y に関するBGP情報を調査するよう依頼する．

(4) R_z は，依頼された調査を独自の診断ツールを用いて行い，R_x に以下の結論を送る．

- AS_x をオリジンとし，AS_y を経由してアドバタイズされた経路 E_x は，AS_z の経路表にはない．
- 経路 E_y は，AS_z の経路表に存在する．しかし，オリジンAS番号は，$AS_{y'}$ に変わっている．

(5) R_x は R_z からの報告を受けとり，R_x は AS_y のAS番号が $AS_{y'}$ に変わったのではないかと，推論する．

このように自ASからのみの観測手段を用いて診断するだけではなく，他AS中のエージェントと協調して他のASにおいて観測・解析を行い，それらの情報を総合して診断を行うことにより，このような詳細な解析が可能となる．この例では，手順(2)終了時点で，自分自身の視点からではこれ以上の推論は不可能になる．そのため，他のエージェントの視点からの調査を手順(3)で依頼し，その結果を手順(2)までの調査で得た結果と統合し，最終的に障害の原因を推論する．このように，対象となる情報が自ASに来なくなる障害の場合，その原因を推測するためには，他のASに存在するエージェントと協調して診断を進める方法が有効である．

現在インターネットでは，55,000エントリを超える[13]BGPによる経路情報が流れており，全経路を常時観測することは困難である．ENCOREでは，この例で述べたように，観測負荷の軽い自AS内のルータの持つ全経路数やBGP peerの状態を示す情報から，障害の発生を予想する．その結果必要に応じて，観測負荷の高い特定の経路に関する詳細な調査を，順次プランニングする機構が有効であると考える．

5.2 診断知識の記述

4.4節で示したように，エージェントの診断知識は，海外および国内から専用線を経由して2つのASからフルルートをBGPで受け，マルチホーム接続をするASでの筆者の3年間ほどの運用経験と，そこで観測された障害に対する解析経験を基に，これまでに約60の仮説とその検証手順を組み込んだ．知識記述には，運用時に発生した障害と，それに対する解析手順を記録しておいたものを用いた．すなわち，筆者らが障害発生時に，その症例から複数の仮説をたて，順次情報を収集しながら，その結果を基に仮説を絞り込み，障害原因を追跡する手順を，そのまま4.2.2項の文法の仮説，その検証ルール，さらにそれが成り立った場合の詳細仮説，に対応させてそのまま記述した．本文法に基づいて与えた，ツール部分を除く知識の記述規模は，現時点で2[Kline]ほどである．本節の例で示したように，全経路数などの変動する値を定常的に観測し，BGP peerの状態やその可到達性から順次仮説を絞り込み，必要に応じて他のエージェントに観測を依頼することにより，半数以上の症例が記述できた．

現在ENCOREには診断知識自体を学習する機能を備えていないため，初期組み込み以外の知識を加える場合には，診断知識を4.2.2項の文法に基づいて与える必要がある．また必要に応じて，診断ツール，あるいは既存の診断ツールを用いる場合は，そのインタフェース関数を用意する必要がある．これらの知識記述に用いた宣言的知識の妥当性は，トポロジの異なるさまざまなネットワークでの検証が必要である．また動的パラメータの特性は，エージェントの存在するネットワーク環境ごとに固有の事象も含まれるため，今後のフィールド実験による環境ごとの妥当性の検証が必要と考える．

現在のENCOREのおいて対応可能な症例は，同一の原因に由来する症状が一定して観測される場合である．観測される症状自体がある周期で変動する事象である場合には，まずそれが同一原因に由来するのかどうかを切り分けなければならないという問題があり，知識の記述力に限界がある．また診断処理中に症例が変わらなかったかどうかを知る必要もある．現在の知識記述の枠組みに時間パラメータを用いて，ad hocに記述する方法は可能であるが，複雑さと記述量の観点から問題がある．また複数の障害事象が相互に干渉し合う症例の記述も，すべての組合せをあらかじめ記述することは困難である．仮説の検証記述に加えて，時間的に変動する事象をどのように処理するかを記述する部分を独立して定義し，メタなレベルでそれらの制御を組み合わせる手法が考えられるが，今後は記述法に関して検討していく予定である．

5.3 実行系の高速化の評価

本節では，連続して行われる複数診断中の複数ルール間の最適化と，1つの診断内で実行される複数ルール間での最適化について議論する．ルールは4.2.2項で述べたように，ルールの検証のために必要な値取得関数と，その結果からルール全体の真偽を決定する評価関数から成る．値取得関数は，ルータやIRRへの問合せ，他のエージェントへの仕事の依頼を含むため，その実行コストは大きく，実際の呼び出しの回数を抑えることは有効と考える．

表1に，タビュレーション技法による診断時間の短縮効果を，特定のアドレスのBGP経路情報が正しく伝搬していることを検証する仮説（check-entry）と，自ASのボーダルータにおけるBGPセッションの状態が正常であることを検証する仮説（check-BGP-peer）の実行時間（秒）を例に示す．これは，実環境系で実験を行った．IRRとしては，`whois.ra.net`を用いた．check-entryでは，自AS中のボーダルータの負荷を確認後，特定のアドレスのBGPエントリを自AS，および自AS外で確認し，IRRデータベースでどのAS中のアドレスかを確認する．check-BGP-peerは，自ASのボーダルータの負荷を確認後，各BGP peerのセッション状態を確認し，IPレベルの到着性を確認する．

表1 タビュレーション技法による最適化の例
Table 1 Example of optimization by tabulation technique.

hypothesis	no tabulation	using tabulation
check-entry	14 [s]	6.4 [s]
check-BGP-peer	15 [s]	8.4 [s]

この例では，比較のため，1) タビュレーション機能を用いない場合と，2) タビュレーション機能を用いた場合を示す．1) ではルール間で外部への問合せの結果が共有できないため，同様の問合せが複数回発生する．2) は，異なる種類の問合せのみが実際にexecuterにより実行された場合である．

従来タビュレーション技法は，計算処理要求が来た場合，その要求仕様と初回の計算結果値を保存しておくことにより，以降の同一の処理要求に対して再度の計算を省略し，計算の高速化を図るものであった．地理的に分散する情報を適用対象とするENCOREのようなAS間診断システムにおいて，診断動作のために外部への呼び出しは必須であり，そのコストの高い診断ツール呼び出し，ボーダルータへの問合せ，外部エージェントへの問合せなどの呼び出しを最適化することは，システムの動作速度の観点から重要である．本例では，その実行機構にタビュレーション技法を適用することによって，コストの高い外部への呼び出しを抑制することが可能となり，このように仮説検証時間を短縮する効果としてその有効性が明確に現れる．また，検証ルール中に記述する外部への呼び出しを，実行段階でタビュレーション技法により最適化するため，知識記述段階において，ルール内に同一の呼び出しを含むルール間の最適化や，同時に検証される複数仮説中に含まれる検証ルール間の最適化を意識しなくてもよい．同時に，診断システムの高速化や記述の容易さの観点に加えて，各ASのボーダルータやインターネットで共有されるIRRは重要な資源であり，監視を行うシステム自体が監視対象になるべく負荷をかけないという観点からも，重要な機能と考える．

また**表2**に，過去の診断の実行履歴に基づく適応型制御による診断時間の短縮効果を，同様にcheck-entry，check-BGP-peerの仮説の検証を例に示す．これも，実環境系で実験を行った．この値は，各仮説の検証において，`priority`値を用いないプランニングを行った場合のルールの呼び出し回数を基準値として，過去の実行履歴で真となった仮説のpriority値を上げ，優先的にプランニングした場合のルールの呼び出し回数との比率を表す．括弧内の分母は，適応型制御を行わなかった場合のルールの呼び出し回数，分子はpriority値を均等に割り振った初期状態から，連続し

表2 過去の履歴情報による最適化の例
Table 2 Example of optimization using historical data.

	check-entry	check-BGP-peer
rate	0.86 (=12/14)	0.81 (=6.5/8)

て診断を4回行った場合の平均ルール呼び出し回数を示す．

この例では，check-entryでは14%減，check-BGP-peerでは19%減と，検証のために実行されたルール数の減少が現れ，過去の履歴情報をプランニングに反映する手法の有効性が確認される．ただし，実環境においては，このように同じ診断事例が続くとは限らないため，直接この効果は反映されないが，環境固有の事象で繰り返し観測される事例には有効な手段になると考える．

5.4 関連研究との比較

WEBで提供されている“looking glass”[14]や“traceroute server”は，経路情報を自AS以外の地点で観測可能とするが，特定の情報をオペレータが取得するのを前提としたツールであり，その解析にはエキスパートの知識が必要となる．またトラフィックの観点からも，定常観測への適用は困難である．また，アクセス制御機能のあるルータと，そのルータからデータを集めてきて解析を行うサーバの組合せによるアプローチがあるが，トラフィックの観点から同様に適用は困難である．GDT[15]のように，リソースの依存関係から順次上流のエージェントに調査依頼をして，自動診断を行うアプローチもあるが，経路情報自体の障害を診断対象とする場合は，依存関係の構築が困難であるという問題がある．

従来のネットワーク診断と知識処理機構を持ったシステムの多くは，単一システムによるもので，ネットワークの地理的，機能的分散には十分な機能が発揮できない[16]．また，複数エージェントの協調を利用したシステムもいくつか提案されている[17]が，対象としている問題設定が本システムとは異なる．たとえば文献18)では，ネットワークの管理単位であるAS内を対象にしていた．したがって，管理主体の異なるAS間で変化しながら伝搬する経路情報を解析するために，情報を直接外部ASから取得したり，自ASに関する観測動作を動的に依頼，実行したりする枠組みは与えていない．

6. おわりに

各ASが広報した経路情報の振舞いを理解し，自ASの意図が正しく反映されて経路情報が伝わっていることを検証するため，他のASからの視点で情報を収集するリフレクタモデルを定義し，それに基づいて動作するAS間診断システムとして，エキスパートの知識を組み込んだマルチエージェントシステムENCOREを提案した．

本システムでは，AS間の協調においてのみ解析可能となる障害事例を診断知識として組み込み，トポロジーを限定した実験環境で複数のエージェントによる解析を行い，その環境下での有効性を確認した．また同様の環境で，連続して行われる外部への値参照をともなうルールの実行において，タビュレーション技法，および過去の診断履歴に基づく適応型制御による効率化の有効性を示した．

今後は，多くのASが複雑に接続する実環境下で，多くのエージェントを配置する形で適用実験を進め，診断知識の有効性，実行系の効率化の検証を進める予定である．また実環境における適用実験を通じて，観測動作戦略の切替えによる効率化，およびエージェントの組織化機構と協調に関する評価を進めていく予定である．

参考文献

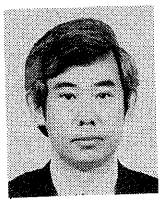
1) Labovitz, C., Malan, G.R. and Jahanian, F.: Internet Routing Instability, *Proc. ACM SIGCOMM* (1997).
2) Paxson, V.: End-to-End Behavior in the Internet, *Proc. ACM SIGCOMM* (1996).
3) 村上健一郎：リフレクタモデルに基づくインターネット管理アーキテクチャ，第52回情報処理学会全国大会論文集，Vol.1, pp.121–122 (1997).
4) 明石 修，菅原俊治，村上健一郎，丸山 充，高橋直久：リフレクタエージェントを用いた自律組織間診断システム，情報処理研究会報告，98-OS-77/98-DPS-87, Vol.98, No.15, pp.161–166 (1998).
5) Akashi, O., Sugawara, T., Murakami, K., Maruyama, M. and Takahashi, N.: Inter-Autonomous-System Diagnosis using Cooperative Reflector Agents, *2nd IEEE Int'l Conf. on Intelligent Processing Systems*, pp.227–232, IEEE (Aug. 1998).
6) Bird, R.S.: Tabulation Techniques for Recursive Programs, *ACM Computing Surveys*, Vol.12, pp.403–417 (1980).
7) Rekhter, Y. and Li, T.: A Border Gateway Protocol 4 (BGP-4), RFC1771 (1995).
8) Postel, J.: Internet Control Message Protocol, RFC792 (1981).
9) Routing Arbiter Project, http://www.ra.net/-RADB.tools.docs/overview.html.
10) Case, J., Fedor, M., Schoffstall, M. and Davin, J.: Simple Network Management Protocol (SNMP), RFC1157 (1990).
11) Mockapetris, P.: Domain Names - Implementation and Specification, RFC1035 (1987).
12) O'hare, G.M.P. and Jennings, N.R.: *Foun-*

dations of Distributed Artificial Intelligence, Wiley-Interscience (1996).
13) Internet Performance Measurement and Analysis Project, http://www.merit.edu/ipma/trends.
14) Kern, E., http://nitrous.digex.net.
15) Thaler, D.G. and Ravishankar, C.V.: An Architecture for Inter-Domain Troubleshooting, *Proc. IEEE Int'l Conf. on Computer Communications and Networks* (1997).
16) 菅原俊治：ネットワーク用エキスパートシステム，人工知能学会誌，Vol.9, No.1, pp.40–47 (1994).
17) Velthuijsen, H.: Distributed Artificial Intelligence for Runtime Feature-Interaction Resolution, *IEEE Computer*, Vol.26, No.8, pp.48–55 (1993).
18) Sugawara, T. and Murakami, K.: A Multiagent Diagnostic System for Internetwork Problems, *Proc. Int'l Networking Conf.*, pp.317–325 (1992).

(平成10年12月1日受付)
(平成11年 4 月1日採録)

明石 修（正会員）

1964年生．1987年東京工業大学理学部情報科学科卒業．1989年同大学院修士課程修了．同年日本電信電話（株）入社．以来，分散システム，コンピュータネットワーク，マルチエージェントシステムの研究に従事．現在，NTT未来ねっと研究所研究主任．ACM，日本ソフトウェア科学会各会員．

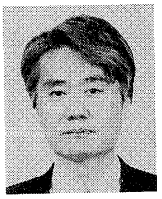


菅原 俊治（正会員）

1980年早稲田大学理工学部数学科卒業．1982年同大学院理工学研究科（数学専攻）修士課程修了．同年，日本電信電話公社入社（武蔵野電気通信研究所基礎研究部）．以来，知識表現，エキスパートシェル，学習，分散協調問題解決，マルチエージェントシステム，インターネット等の研究に従事．1992～1993年，マサチューセッツ大学アムハースト校客員研究員．現在，NTT未来ねっと研究所主幹研究員．博士（工学）．日本ソフトウェア科学会，IEEE，ACM各会員．

村上健一郎（正会員）

1955年生．1979年九州大学工学部情報工学科卒業．1981年同大学院修士課程修了．同年日本電信電話公社入社．現在，NTT研究所主幹研究員．インターネットパラダイム，記号処理計算機等の研究に従事．電子情報通信学会，ACM，インターネット学会，ソフトウェア科学会各会員．主な著書「はやわかりTCP/IP」（共立出版，共訳），「インターネット縦横無尽」（共立出版，共訳），「インターネット」（岩波書店）等．



丸山 充（正会員）

1983年電気通信大学電気通信学部応用電子工学科卒業．1985年同大学院修士課程修了．同年日本電信電話（株）入社．現在，NTT未来ねっと研究所主任研究員．主として，高精細画像情報提供システムの研究開発，ビデオ・オン・デマンドシステムの研究開発に従事．現在，コンピュータネットワークと実時間並列分散アーキテクチャの研究に従事．電子情報通信学会，日本ソフトウェア科学会，IEEE，ACM各会員．工学博士．

高橋 直久（正会員）

1951年生．1974年電通大学応用電子科卒業．1976年同大学院修士課程修了．同年日本電信電話公社武蔵野電気通信研究所入所．以来，機能分散型並列計算機，データフロー型計算システム，ソフトウェア工学，コンピュータ・ネットワーク等の研究に従事．現在，NTT未来ねっと研究所並列分散アーキテクチャ研究グループリーダ．工学博士（東工大）．電子情報通信学会，日本ソフトウェア科学会，ACM各会員．