Vol.2010-DPS-142 No.5
Vol.2010-CSEC-48 No.5
2010/3/4

能動化された知識の組織化による ネットワーク障害管理支援方式

高橋優介[†] 三杉大輔[†] 高橋晶子[††]
笹井一人[†††] 阿部 亨[††††] 木下哲男[††††]

筆者らは，ネットワーク管理者に要求される知的負担や作業負担の軽減を図るため，能動的情報資源（AIR）の概念に基づくネットワーク管理支援システム（AIR-NMS）の実現を目指した研究を進めている．AIR-NMS では，ネットワークの様々な状態情報や，管理者が有する多様なネットワーク管理知識を情報資源とみなし，それらの活用を支援する知識と機能をソフトウェアエージェントとして実装・付加することで，各情報資源へ能動性を与えている．これら能動化された情報資源が組織化・協調することにより，AIR-NMS は，ネットワークの様々な状態情報に基づき，多様な障害への具体的な対策案を管理者に提示することが可能となっている．本稿では，AIR-NMS が必要とする多数のネットワーク管理知識を効果的に獲得・利用する方法を提案し，その効果を実験により評価した結果を述べる．

Network Fault Management Support Method by Organization of Activated Knowledge

Yusuke Takahashi[†] Daisuke Misugi[†] Akiko Takahashi[††]
Kazuto Sasai[†††] Toru Abe[††††] Tetsuo Kinoshita[††††]

To reduce the intellectual and time-consuming loads imposed on network administrators, we pursue our research for realizing AIR-NMS, which is the network management system (NMS) based on Active Information Resource (AIR). In AIR-NMS, various information resources (e.g., status information of network, practical knowledge of network management) are combined with software agents which have the knowledge and function for supporting the utilization of them, and thus individual information resources are given activities as AIRs. Through the organization and cooperation of AIRs, according to various status information of the network, AIR-NMS provides the administrators with practical measures against a wide range of network troubles. In this article, we propose the method for achieving the effective acquisition and utilization of the network management knowledge needed by AIR-NMS, and explain the results of evaluation experiments on our proposal.

1. はじめに

近年，コンピュータネットワークの大規模化・複雑化に伴い，ネットワーク管理には，高度で幅広い管理知識や複雑・煩雑な処理が必要となってきており，その結果，ネットワーク管理者に課される知的負担や作業負担が増大している．これに対し，これまでに多くのネットワーク管理支援手法･システム [1-4] が提案されてきた. しかし，従来の手法・システムでは，例えば，ネットワークの障害管理を行う場合，ネットワークの様々な情報の総合的判断や，多様な障害への具体的対策案の導出が管理者に委ねられているため，管理者（特に管理の知識・経験が乏しい初級管理者）に対する多様かつ高度な支援が期待できなかった．

これら従来のネットワーク管理支援手法・システムが持つ問題の効果的な解決を図るため，筆者らは，能動的情報資源（Active Information Resource: AIR）[5] の概念に基づくネットワーク管理支援システム（AIR-based Network Management Support System: AIR-NMS）[6] の実現を目指した研究を進めている．AIR-NMS では，ネットワークの様々な状態情報（機器やアプリケーションの構成や設定情報，システムやトラフィックのログ情報，等）や，ネットワーク管理者が有する様々な管理知識を情報資源とみなし，それらの活用を支援する知識と機能をソフトウェアエージェントとして実装・付加することで，各情報資源の能動化（AIR 化）を行う．これら能動化された情報資源が組織化を行い，組織された中で協調を図ることにより，AIR-NMS は，ネットワークの様々な状態情報に基づき，多様な障害への具体的な対策案を管理者に提示することが可能となっている．

能動化された知識の組織化によるネットワーク障害管理支援方式により，ネットワークの様々な情報を総合的に判断し，多様な障害への具体的対策案を導出するためには，ネットワーク管理者が有する多数の管理知識を能動化し，蓄積・利用する必要がある．そこで本稿では，実用的な AIR-NMS の実現を目指し，管理知識の効果的な獲得を図るための管理知識の類別・表現方式，及び獲得された管理知識の効率的な利用を図るための能動化された管理知識の協調方式の提案を行い，各方式の効果を実験により評価した結果を述べる．

† 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University
†† 仙台高等専門学校情報工学科
Department of Computer Science, Sendai National College of Technology
††† 東北大学電気通信研究所
Research Institute of Electrical Communication, Tohoku University
†††† 東北大学サイバーサイエンスセンター
Cyberscience Center, Tohoku University

Vol.2010-DPS-142 No.5
Vol.2010-CSEC-48 No.5
2010/3/4

2. 関連研究

2.1 従来のネットワーク管理支援手法・システム

ネットワーク管理における障害管理の作業の流れは，以下のように考えられる．

作業１　障害症状の発見／検知
作業２　障害原因の想定
作業３　想定された原因の診断
作業４　対策案の導出
作業５　対策案の適用

ネットワーク管理支援に関する従来技術の多くは，ネットワークの障害の検知（**作業１**）[1,2] や，障害原因の診断（**作業３**）の一部である，機器やネットワークの情報の収集・提示 [3,4] などを対象とし，これらの支援に限定されるものが多い．また，従来技術の多くは，特定の障害や状況に焦点を絞っているため，対象とする管理知識の種類は限定的なものとなり，その結果，障害原因の想定（**作業２**）は限定的な範囲で行われる．しかし，実際のネットワークの障害管理においては，収集・提示される多種多様な情報からの状況の総合的な判断による様々な障害原因の診断（**作業３**），及びそれに対する具体的な対策案の導出（**作業４**）が必要となり，これらの作業（推論）には高度で幅広い管理知識が要求される．従って，管理作業の一部のみを支援し，対象とする管理知識の種類が限定的である従来技術では，管理者に対する充分な管理支援が期待できない．

2.2 AIR の概念に基づくネットワーク管理支援システム（AIR-NMS）

筆者らは，前述した従来技術が有する問題の効果的な解決を目指し，AIR の概念に基づくネットワーク管理支援システム（AIR-NMS）の実現に向けた研究を進めている．AIR は，情報資源に対する検索，加工，統合，分析などの一連の処理を，情報資源側で能動的に代行・支援させることで，利用者に要求される情報資源の利用負担の軽減を図る概念として提案されている．AIR は，情報資源の活用を支援するための「利用支援知識」と「利用支援機能」を，エージェント，あるいはマルチエージェントとして実装し，情報資源に付加することで実現される．これを情報資源の能動化と呼ぶ．この AIR の概念をネットワーク管理支援に適用したものが AIR-NMS である．

AIR-NMS の概念を図１に示す．AIR-NMS は，機器が保持する状態情報を能動化した I-AIR（Status Information - AIR），及びネットワーク管理知識を能動化した K-AIR（Management Knowledge - AIR）の二種類の AIR で構成される．I-AIR は，障害症状の検知・通知や，状態情報の取得・加工，他の AIR や管理者からの情報要求に対する応答などを行う．K-AIR は，自身が保持する管理知識に基づいて，他の K-AIR や I-AIR と協調し，障害の診断を行う．また，AIR-NMS における診断は，管理者による障害症状の発見・通知により開始する「要求ベース駆動」，または I-AIR による障害症状の検知・通知により開始する「アラームベース駆動」の，二種類の駆動方式によって開始される．AIR-NMS では，様々な機器の状態情報や管理知識を能動化・蓄積していくことで，機器の状態情報の系統的な管理や分散管理，複雑・煩雑な作業の代行・支援などの効果が期待される．これにより，前述の**作業１**から**作業４**において管理者に要求される知的負担や作業負担を大幅に軽減するネットワーク管理支援を実現する．

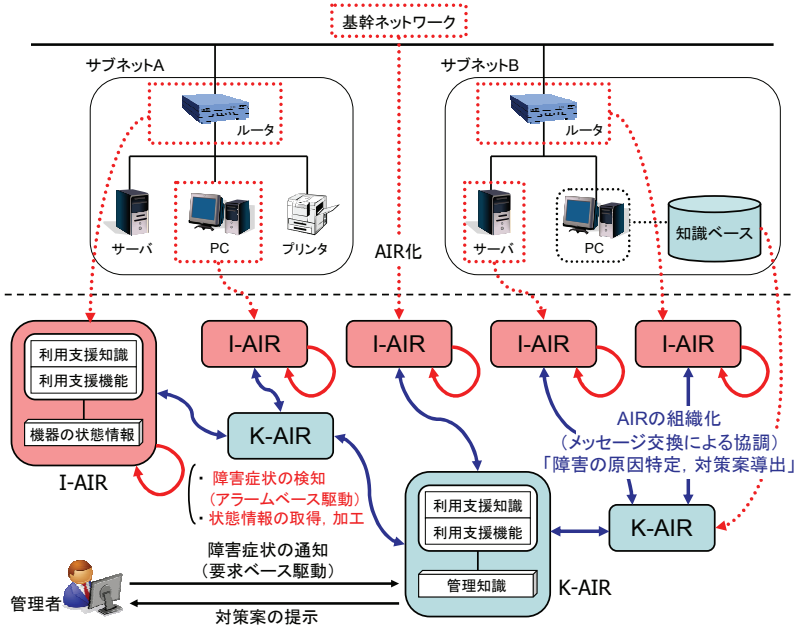


図１．AIR-NMS の概念

実際のネットワーク管理においては，多様で幅広い管理知識（物理層からアプリケーション層までの各層における様々な管理知識）の活用が求められる．従って，実用的な AIR-NMS を実現するためには，AIR-NMS における多数の管理知識の活用（獲得・利用）を図る方法が必要となる．

3. 能動化された知識の組織化によるネットワーク障害管理支援方式

3.1 提案方式の概要

本研究では，AIR-NMS におけるネットワーク管理知識の効果的な活用の実現を目的とし，実際のネットワーク管理において必要とされる多数のネットワーク管理知識を効果的に獲得・利用する方式を提案する．本提案方式は以下の二つからなる．

Vol.2010-DPS-142 No.5
Vol.2010-CSEC-48 No.5
2010/3/4

1. 管理知識の類別・表現方式
2. 管理知識の協調方式

提案する類別・表現方式により，様々な管理知識の効果的な獲得，具体的には管理知識の追加・変更の容易性，管理知識の再利用性の向上を図る．また，協調方式により，様々な管理知識の効果的な利用，具体的には類別・表現された管理知識の能動的・網羅的活用と，診断処理の頑健性の向上を図る．

3.2 管理知識の類別・表現方式

3.2.1 管理知識の類別方式

2.1 で述べたネットワーク障害管理の作業中，主に管理者の管理知識により診断（推論・処理）が進められる部分は，以下の作業と考えられる．

作業２　障害原因の想定 ― 発見・検知した障害症状から複数の障害原因を想定
作業３　想定された障害原因の診断 ― 各障害原因について診断（確認）を実施
作業４　対策案の導出 ― 障害原因・診断結果から具体的な対策案を導出

提案方式では，**作業２**，**３**，**４**の単位で管理知識を類別（分割）することを考える．具体的には，AIR-NMS での利用を踏まえ，各々の推論・処理において抽出される知識要素の組を，以下の三種類の管理知識 **K** として類別する．

- $\mathbf{K_{SC}}$： **障害症状**（Symptom）と，それから想定される**障害原因**（Cause）の組で構成される管理知識
- $\mathbf{K_{CD}}$： 想定される**障害原因**（Cause）と，その**診断手法**（Diagnosis Method），及び診断報告（Diagnosis Report）のテンプレートの組で構成される管理知識
- $\mathbf{K_{CM}}$： 特定された**障害原因**（Cause）と，**対策案**（Measure）のテンプレートの組で構成される管理知識

このように，推論・処理に用いられる管理知識を三種類に類別し，類別された各々を管理者や AIR-NMS による知識獲得の単位として用いる．

類別された管理知識の例を図２に示す．図２中の K_{SC}-1 は，障害症状として「メール送信不可」，それに関する障害原因として「名前解決不可」，「ネットワーク接続障害」，「送信可能メールサイズ超過」を管理知識として構成する．K_{CD}-2 は，障害原因として「送信可能メールサイズ超過」，その診断手法として「送信メールサイズと当該 SMTP サーバの設定情報を取得し，各々を比較することで当該障害原因の真偽を判定する」命令，診断報告として当該診断手法の実施結果を通知するテンプレートを管理知識として構成する．K_{CM}-1 は，障害原因として「送信可能メールサイズ超過」，対策案として「SMTP サーバの OS が CentOS，MTA が Postfix である場合の，サーバの設定変更の具体的な操作手順」を通知するテンプレートを管理知識として構成する．

類別された管理知識は，利用する段階で関連するもの同士の選択・統合が必要となるが，提案方式では，設計段階では相互の関連を示す静的なリンクを与えないことで，設計の独立性を保持する．

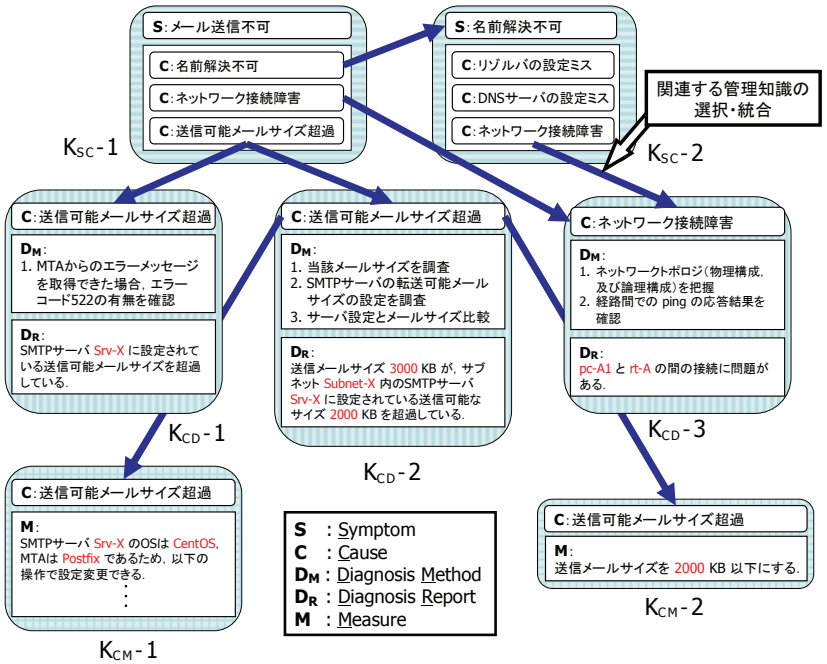


図２．類別された管理知識の例

ここで，管理知識を K_{SC}，K_{CD}，K_{CM} の形に類別する利点について考える．まず，一般に，「障害症状－障害原因」，「障害原因－診断手法」，「障害原因－対策案」各々で，知識要素間の関係は，一対多や多対一となる複雑な意味ネットワーク（知識要素の依存関係）を形成し得る．このことから，管理知識を類別して設計しない場合，管理知識の追加・変更の度に，この意味ネットワークの構造を考慮する必要があり，結果的に，管理知識の容易な追加・変更が困難となる．従って，管理知識を類別し，設計段階では相互の関連を示す静的なリンクを与えないことで，新たな管理知識の追加・変更による他への影響を抑制でき，管理知識の追加・変更の容易性の向上が期待できる．また，異なる診断の過程でも，推論・処理が部分的に重複／類似する場合が考えられる．具体的には，一つの障害原因が二つ以上の異なる障害症状を引き起こす場合や，ある障害症状の発生が別の障害症状の原因となる場合，別々の障害原因の診断手法が類似する（同一視できる）場合などが挙げられる．そのため，提案方式の通り管理知識を類別することで，共通する管理知識の再利用性が向上し，新たな管理知識の追加・変更に要する手間の削減が可能となる．

Vol.2010-DPS-142 No.5
Vol.2010-CSEC-48 No.5
2010/3/4

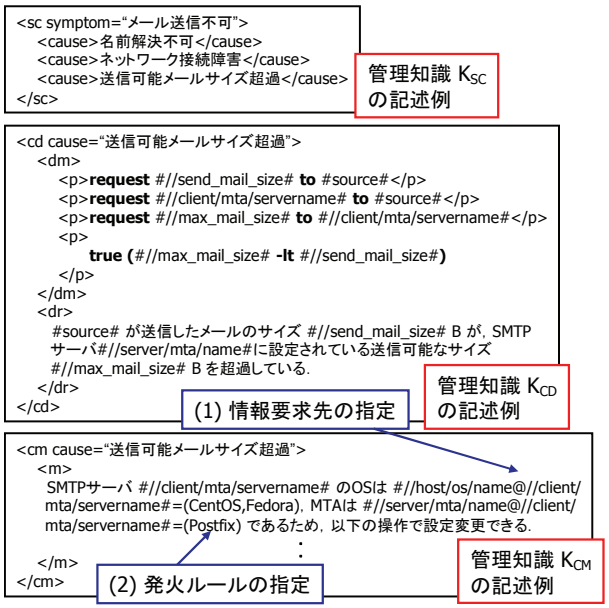


図３．類別・表現された管理知識の例

3.2.2 管理知識の表現方式

次に，類別する管理知識を構成する知識要素の表現方式について述べる．K_{SC} が持つ「障害症状」や，K_{SC}，K_{CD}，K_{CM} が持つ「障害原因」は，システムで一意的な属性・属性値を定義・記述することで表現する．また，K_{CD} が持つ「診断手法」は，診断における推論・処理の命令を記述する．その内容は例えば，I-AIR や管理者への「情報要求」や，想定される障害原因の真偽を判定するための「情報比較」などである．また，K_{CD} が持つ「診断報告」のテンプレートや，K_{CM} が持つ「対策案」のテンプレートは，ネットワークドメインに依存する項目を空欄化（変数化）したプレーンテキストで表現する．空欄化する部分に対しては，システムで一意的な属性を定義・付加する．このとき，「対策案」のテンプレートについては，必要に応じて，属性に対して対策案の生成条件（発火ルール）を付加することで，必要となる対策案の選択・生成を可能とする．診断の過程で取得・利用された情報の中から，空欄部分（属性）に対応する属性値を選択・代入することで，具体的な診断報告や対策案を生成する．特定のネットワークドメインに依存しない形で管理知識を表現にすることにより，管理知識の再利用性の向上を図る．

類別された管理知識の表現例を図３に示す．管理知識全体は，XML [7] を用いて表現する．図３中の K_{SC} では，障害症状の属性<symptom>タグに対して「メール送信不可」を与え，障害原因の属性<cause>タグに対して「名前解決不可」,「ネットワーク接続障害」,「送信可能メールサイズ超過」を与えている．また，K_{CD} では，診断手法部分を示す<dm>タグに対して推論・処理の命令を表現するスクリプト一行一行を<p>タグを付加して与えている．一行目の「**request** #//send_mail_size# **to** #source#」は「当該メールクライアント（の I-AIR）に対して当該送信メールのサイズを要求する」命令，二行目の「**request** #//client/mta/servername# **to** #source#」は「当該メールクライアント（の I-AIR）に対して利用している SMTP サーバ名を要求する」命令，三行目の「**request** #//max_mail_size# **to** #//client/mta/servername#」は「当該 SMTP サーバ（の I-AIR）に対して設定されている転送可能メールサイズを要求する」命令，四行目の「**true** (#//max_mail_size# **-lt** #//send_mail_size#)」は「当該 SMTP サーバに設定されている転送可能メールサイズが当該送信メールのサイズより小さい場合，当該障害原因を「真」と判定する」命令を意味する．また，診断報告部分を示す<dr>タグに対してドメイン依存項目が空欄化された診断結果を示すテンプレートを与えている．同様に，K_{CM} では，対策案部分を示す<m>タグに対してドメイン依存項目が空欄化された対策案のテンプレートを与えている．

また，タグ付けが行われるドメイン依存項目に関しては，XPath [8] を用いて表現する．XPath は XML 文書中の任意の情報を取り出すための問合せ言語である．XPath を利用する理由は，I-AIR によって取得される機器の状態情報を XML 表現で保持するためである．さらに，表現の拡張として，図３中の（１）に示すように情報要求先の情報（属性）を‘@’で結合し，また（２）に示すように発火ルールとする属性値を‘＝’で結合することで，管理知識の適切な選択・利用を可能とする．

3.3 管理知識の協調方式

前述のように，類別・表現された管理知識（K_{SC}，K_{CD}，K_{CM}）の利用には，関連する管理知識の統合や，統合された管理知識に基づく障害診断のための処理が必要となる．これらを管理者に委ねた場合，管理者に大きな利用負担を課すことになる．

そこで AIR-NMS では，能動化された管理知識（K-AIR）の組織化（能動的な統合・利用）による管理知識の利用負担の軽減を考える．ここでの管理知識の能動化とは，類別・表現された管理知識各々に対して，管理知識間の協調のための利用支援知識・機能の付加を行うことであり，具体的には，協調の際にやり取りされるメッセージの交換手順と表現形式の設計が必要となる．

K-AIR の組織化のイメージを図４に示す．別々のワークプレース（AIR の動作環境）で動作する類別・表現・能動化された管理知識はそれぞれ，関連する管理知識の統合・利用を目的とした組織化や，機器情報の取得・利用を目的とした組織化を行う．組織された AIR 間で協調することで，障害の診断を進める．

Vol.2010-DPS-142 No.5
Vol.2010-CSEC-48 No.5
2010/3/4

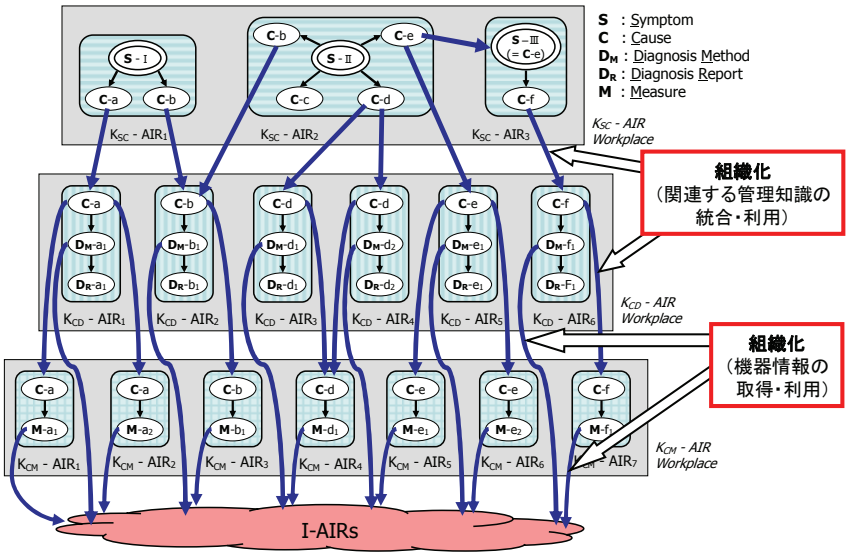


図４．K-AIR の組織化のイメージ

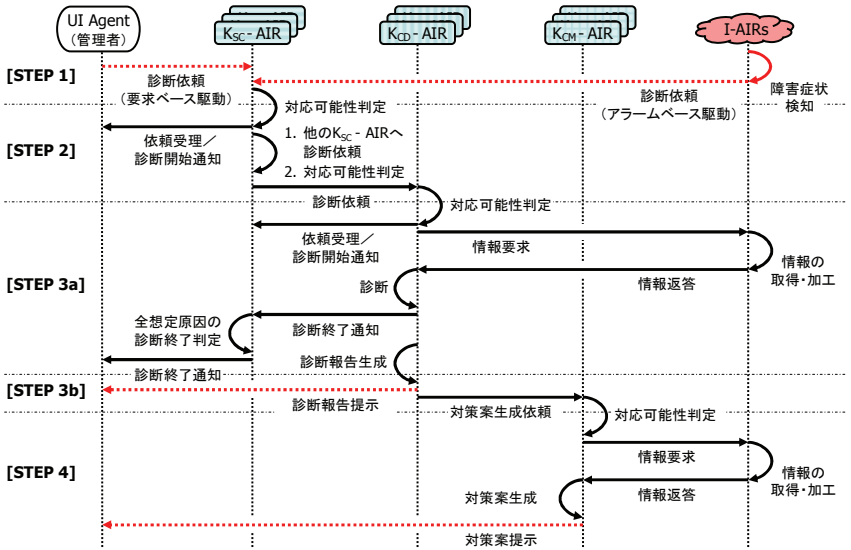


図５．K-AIR 間のメッセージの交換手順

K-AIR 間のメッセージの交換手順を図５に示す．その詳細を以下に述べる．

[STEP 1] 管理者からの要求，または **I-AIR** からのアラームとして，発生した障害症状に関する診断依頼（組織化要求）のメッセージを，**K_{SC}-AIR** 群が動作するワークプレースにブロードキャスト送信する．

[STEP 2] 診断依頼を受け取った **K_{SC}-AIR** は，当該障害症状に対応可能であるかを，自身が保持する管理知識における障害症状の属性値の一致／不一致によって判定する．対応可能である場合は，**UI_Agent**（管理者からの入力や，管理者への出力を処理するユーザインタフェースエージェント）に診断開始を通知する．また同時に，障害症状に対して想定される障害原因の診断依頼（組織化要求）のメッセージを，**K_{CD}-AIR** 群が動作するワークプレース，及び他の **K_{SC}-AIR** が動作するワークプレースにブロードキャスト送信する．ここで，診断依頼を受け取った **K_{SC}-AIR** は，当該障害原因を障害症状と捉え，**[STEP 2]**を開始する．

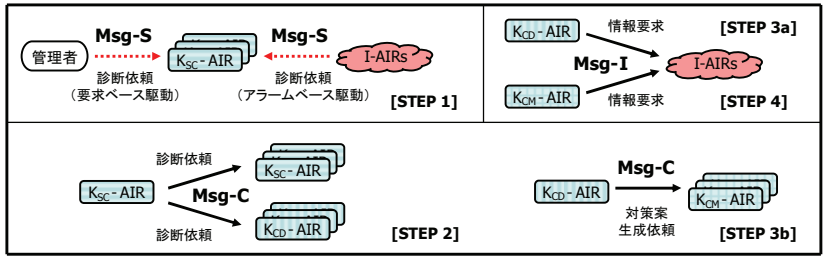
[STEP 3a] 診断依頼を受け取った **K_{CD}-AIR** は，当該障害原因に対応可能であるかどうかを，自身が保持する管理知識における障害原因の属性値の一致／不一致によって判定する．対応可能である場合は，依頼元の **K_{SC}-AIR** に診断の開始を通知する．また同時に，当該障害原因に関する診断（想定される障害原因の真偽判定）を行う．診断のために必要な具体的な情報（属性値）は，ネットワーク上の **I-AIR** 群に情報要求のメッセージをブロードキャスト送信して取得する．診断終了後，依頼元の **K_{SC}-AIR** に診断終了を通知する．このとき，**K_{SC}-AIR** は自身が診断を依頼した **K_{CD}-AIR**, あるいは他の **K_{SC}-AIR** 全てからの診断終了通知を受け取った場合，**UI_Agent** へ診断の終了を通知する．

[STEP 3b] 診断を行った **K_{CD}-AIR** において，診断の結果，当該障害原因が「真」である場合は，診断の過程で取得・利用した情報（診断情報）を元に，診断報告のテンプレートの補完を行い，具体的な診断結果を生成し，**UI_Agent** へ提示する．また同時に，当該障害原因に関する対策案の生成依頼（組織化要求）のメッセージを，**K_{CM}-AIR** が動作するワークプレースにブロードキャスト送信する．

[STEP 4] 対策案の生成依頼を受け取った **K_{CM}-AIR** は，当該障害原因に対応可能であるかどうかを，自身が保持する管理知識における障害原因の属性値の一致／不一致によって判定する．対応可能である場合は，対策案の生成に必要な情報（属性値）を，**K_{CD}-AIR** からのメッセージで得られた診断情報の中から，または **I-AIR** 群に対して情報要求を行い，取得し，テンプレートの補完を行う．対策案の生成後，それを **UI_Agent** へ提示する．

各 K-AIR は全て並列的に動作し，協調処理における自身のセッションの状態（図４中の各 K-AIR から縦に延びる破線中の位置）を常に把握し，全体の処理としての整合

Vol.2010-DPS-142 No.5
Vol.2010-CSEC-48 No.5
2010/3/4



Msg-S ::= <task id> <symptom> <source> <detail info>*
Msg-C ::= <k-air id> <task id> <cause> <source> <cooperator>+ <detail info>*
Msg-I ::= <k-air id> <request info>+ <destination>

図６．K-AIR 間のメッセージの表現形式

性を保持して診断処理を進める．管理知識各々を，K-AIR として並列的に処理させる効果して，診断処理の頑健性の保持が期待できる．各 K-AIR は，互いのリンクを与えず独立して設計し，動作させるため，K-AIR の動作の影響（例えば，管理知識の記述の不備や，AIR プログラムの予期せぬ動作不良，情報取得の失敗による実行時間の遅延，等）が全体の処理として広がりにくい特長が考えられる．診断処理の頑健性が保持されることで，運用（様々な管理知識の追加・変更，利用）が容易になるといえる．

次に，K-AIR 間のメッセージの表現形式を図５に示す．その詳細を以下に述べる．

- **Msg-S**： 図４の**[STEP 1]**における，**UI_Agent** または **I-AIR** から **K_{SC}-AIR** にブロードキャスト送信される診断依頼のメッセージ
  - **<task id>** 診断処理（タスク）が開始される際に生成されるタスクの識別子である．各 K-AIR が組織化を行う際に，この識別子を保持・共有することで，K-AIR からの同一の要求を二回以上処理することを抑制する．
  - **<symptom>** 診断対象となる障害症状の属性値（例えば「メール送信不可」,「名前解決不可」, 等）である．診断開始のために最低限入力すべき情報とする．
  - **<source>** 障害症状が観測されたホストやネットワークの名前や IP アドレスである．診断開始のために最低限入力すべき情報とする．
  - **<detail info>** 任意に（０項目以上）与えられる，診断処理に利用可能な情報（属性と属性値の組）である．
- **Msg-C**： 図４の**[STEP 2]**における，**K_{SC}-AIR** から **K_{CD}-AIR** 群と他の **K_{SC}-AIR** 群にブロードキャスト送信される診断依頼のメッセージと，**[STEP 3b]**における，**K_{CD}-AIR** から **K_{CM}-AIR** 群にブロードキャスト送信される診断依頼メッセージ
  - **<k-air id>** メッセージ送信元（組織化要求元）の K-AIR の識別子である．この識別子は K-AIR 組織化後，K-AIR や I-AIR 間で協調を行う（メッセージ送信元に返答する）ために利用する．
  - **<cause>** 診断要求された障害症状に関して想定される障害原因の属性値（例えば「送信可能メールサイズ超過」,「接続ポート未開放」, 等）である．
  - **<source>** **Msg-S** から引き継がれる障害症状の発生元の情報である．
  - **<cooperator>** このメッセージが受け取られるまでに組織化された K-AIR の情報（管理知識の情報と K-AIR の識別子）である．K-AIR が形成したネットワーク（管理知識の意味ネットワーク）の構造を把握するために利用する．これにより，管理者に対する管理知識の獲得支援の効果が期待できる．
- **Msg-I**： 図４の**[STEP 2]**における，**K_{CD}-AIR** から **I-AIR** にブロードキャスト送信される情報要求のメッセージ
  - **<request info>** **K_{CD}-AIR** における診断に必要となる情報や，**K_{CM}-AIR** における対策案のテンプレートの補完に必要となる情報となる，一項目以上の属性である．各 **I-AIR** は，自身の保持する機器の状態情報の中から対応する属性値を見つけた場合，**<k-air id>**宛にその属性値を返答する．
  - **<destination>** **<request info>**に付加される，要求する属性の要求先のホスト名や IP アドレスである．特定のホストからの情報要求を指定する場合に利用する．ここで指定する情報要求先で働く **I-AIR** のみが情報返答を行う．

4. 評価実験

4.1 類別・表現方式に関する評価

提案する類別・表現方式の有効性を検証するため，管理知識の獲得の容易性を評価する実験を行った．定量的な評価を行うために，ここでは，管理知識の設計負担（管理知識の設計に要する手間）の指標として「知識の記述量 N_K」を定義し，様々な類別方法に基づき管理知識を設計する際に必要な N_K を比較した．N_K は，管理知識中の知識要素（障害症状（**S**），障害原因（**C**），診断手法・報告（**D**），対策案（**M**））の対（**S-C**，**C-D**，**C-M**）の総数であり，例えば，**S-C** 1 対のみからなる管理知識を設計する場合は $N_K = 1$ の負担となる．

表１に示す情報資源（K_{SC} 7 種類，K_{CD} 40 種類，K_{CM} 80 種類）に対して，提案方式を含む下記の五種類の類別方法を適用した場合の各々の N_K を表２に示す．

1. 「①K_{SC}，②K_{CD}，③K_{CM}」単位の設計（提案方式）
2. 「同一 **C**（障害原因）を扱う K_{SC}，K_{CD}，K_{CM} の一組」単位の設計
3. 「①K_{SC}，②同一 **C** を扱う K_{CD} と K_{CM} の一組」単位の設計
4. 「①同一 **C** を扱う K_{SC} と K_{CD} の一組，②K_{CM}」単位の設計
5. 「同一 **C** を扱う K_{SC}，K_{CD}（複数可），K_{CM}（複数可）の一組」単位の設計

Vol.2010-DPS-142 No.5
Vol.2010-CSEC-48 No.5
2010/3/4

表１．類別・表現方式の評価実験で用いた情報資源の一覧

障害原因(C)	K_{CD} の種類数	K_{CM} の種類数	K_{SC} として設計された障害症状(S)						
			送	受	閲	ロ	共	名	接
サーバプロセスダウン	2	11	○	○	○	○		○	
接続ポート番号設定ミス	1	2	○	○					
サーバ／アカウント情報設定ミス	1	2	○	○					
POP before SMTP	1	1	○						
サーバタイムアウト時間設定ミス	0	2	○	○					
同時接続可能数超過	1	4	○	○					
送信可能メールサイズ超過	2	3	○						
MXレコード設定ミス	2	1	○						
鍵認証設定ミス	2	3	○	○		○			
宛先指定ミス	2	0	○			○			
サーバアクセス過多	2	6	○	○	○				
アドレス別アクセス制限	2	8	○	○	○	○		○	
接続ポート未解放	1	2	○	○	○	○	○	○	
名前解決不可（＝Symptom）	1	0	○	○	○	○	○		
ネットワーク接続障害（＝Symptom）	1	0	○	○	○	○	○	○	
保存可能メールサイズ超過	2	2		○					
受信可能メールサイズ超過	2	2		○					
HTTPプロキシ情報設定ミス	1	2			○				
プロキシフィルタリング	2	2			○				
古いページキャッシュ	1	2			○				
ウェブページリンク切れ	1	1			○				
ワークグループ設定ミス	1	2					○		
ファイル共有設定ミス	1	2					○		
ネームサーバ情報設定ミス	1	2						○	
ホスト名設定ミス	1	2						○	
古いリゾルバキャッシュ	1	3						○	
IPアドレス重複割当	1	2							○
ルーティングテーブル設定ミス	1	1							○
NICの故障／ダウン	1	2							○
NICドライバ異常	1	2							○
NICのIPアドレス設定ミス	1	2							○
ケーブル断線／劣化／配線ミス	0	3							○
ネットワーク機器の故障／停止	0	1							○

障害症状(Symptom)
送：「メール**送**信不可」
受：「メール**受**信不可」
閲：「ウェブページ**閲**覧不可」
ロ：「遠隔**ロ**グイン不可」
共：「ファイル**共**有不可」
名：「**名**前解決不可」
接：「ネットワーク**接**続障害」

表２．様々な類別方法における設計負担（知識の記述量 N_K）

管理知識の類別方法	N_K
1.「①K_{SC}，②K_{CD}，③K_{CM}」単位の設計（提案方式）	184
2.「同一Cを扱うK_{SC}，K_{CD}，K_{CM}の一組」単位の設計	996
3.「①K_{SC}，②同一Cを扱うK_{CD}とK_{CM}の一組」単位の設計	298
4.「①同一Cを扱うK_{SC}とK_{CD}の一組，②K_{CM}」単位の設計	250
5.「①同一Cを扱うK_{SC}，K_{CD}（複数可），K_{CM}（複数可）の一組」単位の設計	341

本実験では，類別方法「２」，「５」，「３」，「４」，「１（提案方式）」の順で N_K が大きい結果となった．同じ診断を行う管理知識を設計する際，N_K が大きい類別方法ほど，設計負担（管理知識を新たに作成するための記述量）が大きいことになる．従って，提案方式は，他の類別方法に比べ少ない負担（最大で約 5.4 倍，最小で約 1.6 倍）で管理知識を設計できていることが分かる．これは，提案方式では，K_{SC}，K_{CD}，K_{CM} 各々を一個ずつ独立して設計するため，管理知識中の重複部分の再利用性が高くなり，管理知識を新たに設計する手間を省くことができたためである．また，今回の実験で対象とした管理知識のように，一般に K_{CM} にはドメイン依存の要素（例えば，OS やア

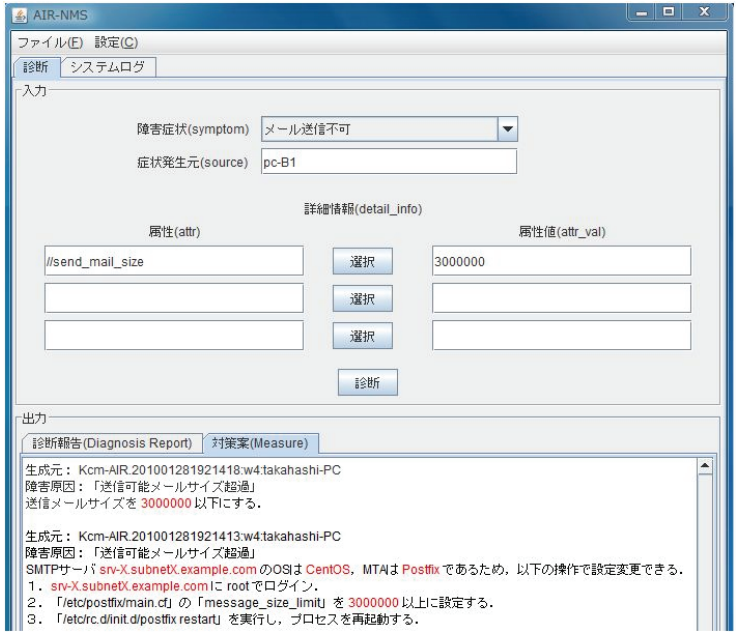


図７．実装した AIR-NMS の実行例

プリケーションに依存する操作方法・手順，等）が含まれるため，K_{SC} や K_{CD} に比べて多種類の知識の設計が必要となる（K_{CM} は，K_{SC} や K_{CD} に比べて再利用できるものが少ない）．そのため，提案方式以外の類別方法の中では，K_{CM} を独立して設計している類別方法「４」での N_K が最も小さくなったと考えられる．

以上の実験結果は，提案する類別・表現方式により，管理知識の獲得の容易性が向上されることを示すものと考えられる．

4.2 協調方式に関する評価

提案する協調方式の有効性を検証するため，診断処理を行う際の頑健性を評価する実験を行った．具体的には，診断処理を行う際に処理時間の大半を占める K_{CD} に着目し，①複数の K_{CD} を一つの情報資源として能動化して処理（逐次処理）する場合と，②提案方式である複数の K_{CD} 各々を能動化して処理（並列処理）する場合について，I-AIR からの情報取得に失敗する場合を含む障害診断の実行時間を比較・検証した．

実験には，図６に示す試作した AIR-NMS を用いた．試作システムは，分散環境上でマルチエージェントシステムを実現するためのフレームワークである ADIPS／DASH フレームワーク[9,10] を用いて実装し，以下の実験環境上に構築した．

Vol.2010-DPS-142 No.5
Vol.2010-CSEC-48 No.5
2010/3/4

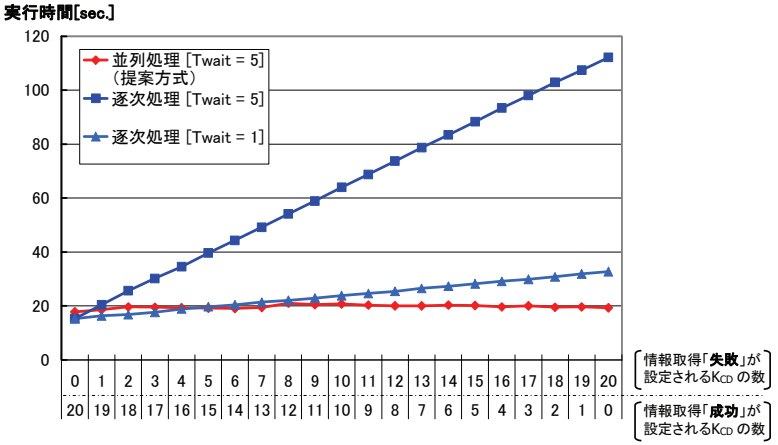


図 7．逐次処理と並列処理（提案方式）の実行時間

- OS： Windows 7
- CPU： Intel Core 2 Duo E6600 (2.4 [GHz])
- メモリ： 2.00 [GByte]

処理される K_{CD} の数を N，I-AIR への情報要求に対する最大待ち時間を T_{WAIT}，一個の K_{CD} の処理（情報応答待ち以外）に要する実行時間を T_{PS}，I-AIR からの情報取得に「失敗」する K_{CD} の発生確率を P とした場合，理論値として，逐次処理の実行時間は $N(T_{PS} + PT_{WAIT})$，並列処理の実行時間は $NT_{PS} + T_{WAIT}$ になると考えられる．T_{WAIT} は，I-AIR からの応答が得られない可能性があることを踏まえて設定する情報応答の最大待ち時間である．また今回は，単一の計算機上で実験を行うため，並列処理を行った場合でも，逐次処理と同じ N 倍の T_{PS} が要求されると考えられる．しかし，プロセスとしては並列化が行えると考えられるため，並列処理の場合，複数回の情報取得「失敗」が発生しても，最大で T_{WAIT} の増加に抑えられると想定される．

実験に用いる K_{CD} は，三回の「情報要求」と一回の「情報比較」の命令を含む診断手法で構成する．I-AIR からの情報取得に「失敗」する状況（T_{WAIT} の待ち時間の発生を設定する場合は，三回目の情報要求で情報取得に「失敗」する命令を記述する．

実験結果を図 7 に示す．実験には 20 個の K_{CD} を用いた．逐次処理では，20 個の K_{CD} をまとめて一個の K_{CD}-AIR として動作させたものである．一方，並列処理（提案方式）では，20 個の K_{CD} 各々を能動化し，20 個の K_{CD}-AIR として並列的に動作させたものである．図 7 は，情報取得「失敗」の K_{CD} を 0 から 20 個設定した，$T_{WAIT} = 5$[s]（並列処理における最小設定）とする並列処理（提案方式），$T_{WAIT} = 5$[s]，及び $T_{WAIT} = 1$[s]（逐次処理における最小設定）とする逐次処理の，三種類の処理の実行時間を示す．

実験結果から，管理知識を逐次的に処理する場合は，約 $N(T_{PS} + PT_{WAIT})$ の実行時間となり，N と P が大きくなる場合，非実用的な実行速度になるといえる．一方，提案方式である，管理知識を並列的に処理する場合は，約 $NT_{PS} + T_{WAIT}$ の実行時間となり，P に依存しないことがわかり，情報取得できない状況に対して頑健（逐次処理と比較して優位）であるといえ，診断処理の頑健性を保持できるといえる．

5. おわりに

本稿では，能動化された知識の組織化によるネットワーク障害管理支援方式において必要とされる多数の管理知識を効果的に獲得・利用するために，管理知識の類別・表現方式，及び管理知識の協調方式を提案した．また，評価実験により，管理知識の獲得の容易性，及び診断処理の頑健性の向上に対する提案方式の効果を確認した．

今後は，実際のネットワーク上で，試作システムの動作検証と協調方式の評価実験を行う予定である．

参考文献

[1] V.K.Singh and H.Schulzrinne: “DYSWIS: An architecture for automated diagnosis of networks,” Network Operations and Management Symposium, pp.851-854, Apr. 2008.
[2] J.Miao, M.A.Munawar, and T.Reidemeister, P.A.S.Ward: “Automatic Fault Detection and Diagnosis in Complex Software Systems by Information-Theoretic Monitoring,” IEEE/IFIP International Conference on Dependable Systems & Networks, pp.285-294, Jun.-Jul. 2009.
[3] M.H.Kim, S.Lim, and J.Kim: “Modeling of Real-Time Distributed Network Management based on TMN and the TMO Model,” Proceeding of the Eighth International Workshop on Object-Oriented Real-Time Dependable Systems, pp.56-63, Jan. 2003.
[4] R.Salles, E.Cecilio, S.Cardoso, A.Correia, and F.Bleasby: “A Test-oriented Architecture for Network Fault Management,” Network Operations and Management Symposium, pp.1-7, Sep. 2007.
[5] 木下 哲男: “分散情報資源活用の一手法,” 電子情報通信学会技術報告, AI99-54, 1999.
[6] S.Konno, A.Sameera, Y.Iwaya, T.Abe, and T.Kinoshita: “Knowledge-Based Support of Network Management Tasks Using Active Information Resource,” The IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp.195-199, Dec. 2006.
[7] T.Bray, J.Paoli, C.M.Sperberg-McQueen, E.Maler, and F.Yergeau: “Extensible Markup Language (XML) 1.0 (Fifth Edition),” W3C Recommendation, Nov. 2008.
[8] A.Berglund, S.Boag, D.Chamberlin, M.F.Fernandez, M.Kay, J.Robie, and J.Simeon: “XML Path Language (XPath) 2.0,” W3C Recommendation, Jan. 2007.
[9] 藤田 茂, 菅原 研次, 木下 哲男, 白鳥 則郎: “分散処理システムのエージェント指向アーキテクチャ,” 情報処理学会論文誌, Vol.37, No.5, pp.840-852, May 1996.
[10] DASH GROUP: “DASH – Distributed Agent System based on Hybrid architecture,” http://www.agent-town.com/dash