

能動化された状態情報に基づくネットワーク管理支援方式

今野 将[†] 吉村 智志^{††} 羽鳥 秀明^{††}
 岩谷 幸雄^{†††} 阿部 亨[†] 木下 哲男[†]

筆者らは、能動的情報資源（AIR）の概念に基づく新たなネットワーク管理支援システム（NMS）を提案している。AIR-NMS では、ネットワーク管理に必要な経験的知識やネットワークの状態情報が AIR として構造化（AIR 化）され、ソフトウェアエージェントを用いて実装される。そして、これらのエージェントの協調や組織化により、従来ネットワーク管理者に委ねられていた作業の一端を支援する。本論文では、ネットワークの状態情報を利用した管理業務に焦点をあてて、ネットワーク状態情報を AIR 化した I-AIR のアーキテクチャ、I-AIR の設計法、および、I-AIR における協調・組織化の機構を提案する。さらに、プロトタイプシステムの試作により、AIR-NMS の基本特性と支援機能を評価する。

Network Management Support System Based on the Activated Status Information

SUSUMU KONNO,[†] SATOSHI YOSHIMURA,^{††} HIDEAKI HATORI,^{††}
 YUKIO IWAYA,^{†††} TORU ABE[†] and TETSUO KINOSHITA[†]

The authors have proposed a novel network management support system (NMS) based on the active information resource (AIR). In the proposed system, called AIR-NMS, the empirical knowledge and status information required for network management are structured as various AIRs and implemented by using software agents. Through the coordination/organization of these agents, they support a part of the network management tasks, which are usually burdened to the network administrators. In this paper, focusing on a management task based on the network status information, we propose an architecture, a design method and the coordination/organization mechanism of I-AIR which holds network status information represented as the AIR-based descriptions in the AIR-NMS. Moreover, a prototype system of the AIR-NMS is implemented to demonstrate and evaluate the essential functions of the AIR-NMS.

1. ま え が き

近年のネットワークシステムは、ますます大規模・複雑になってきており、これを管理するための一連の作業にも、より高度な経験的知識・煩雑な手順が要求されてきている。これらの負担を軽減するために提案されている従来のネットワーク管理支援システム（NMS: Network Management Support System）は、そのほとんどが、管理に必要なネットワークの状態情報や一

般的な障害対策を管理者に提示するにとどまり、総合的な情報の判断や具体的な対策の決定は、依然として管理者に委ねられている^{1)~4)}。また、従来の NMS は、ネットワークの構成が変更されたり新たな管理知識を追加したりする場合など、NMS 自体の変更が必要な状況への対応がきわめて困難である。

筆者らは、これら従来の NMS が有する問題を克服するために、「能動的情報資源（AIR: Active Information Resource）」⁵⁾の概念を用いた新しいネットワーク管理支援システム（AIR-NMS: AIR-based Network Management Support System）^{6)~8)}を提案している。AIR-NMS では、ネットワーク管理のための経験的知識やネットワークの状態情報が各々エージェントとして能動化（AIR 化）され、それらを自律的に連携・組織化させることで、ネットワーク障害に関する「状況の把握/原因の特定/対策の決定」の能動的な支援を目

[†] 東北大学情報シナジーセンター
Information Synergy Center, Tohoku University

^{††} 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University

^{†††} 東北大学電気通信研究所
Research Institute of Electrical Communication, Tohoku University

指す。

本論文では、AIR-NMSの基本アーキテクチャ^{(6)~(8)}におけるネットワークの状態情報の構成、および、これに基づく能動的な状態監視機能に焦点を絞りその設計法を提案し、試作したシステムによる評価実験により、提案手法の有効性を示す。

2. AIRを用いたネットワーク管理支援システム

ここでは、「能動的情報資源 (AIR)」の概要と、AIRをネットワーク管理の作業に適用する際の基本的な考え方、AIR-NMSとしての動作イメージ、および、AIRの導入によって期待される効果について述べる。

2.1 能動的情報資源 (AIR)

現在、インターネットを代表とする分散環境上には、膨大な量の各種情報資源 (文字・記号、画像・図形、音など種々の対象を電子化したデータ) が蓄積されている。このような分散情報資源を実際に活用するためには、一般に、検索/加工/統合など情報資源に対する一連の処理が必要であり、これに要する煩雑な手間は、専門知識や専用ツールを持たない利用者が分散情報資源を活用する際の大きな障害となっている。

AIRは、この問題を解決し分散情報資源の有効活用を図るために提案された手法である⁽⁵⁾。この手法では、利用支援知識と利用支援機能とを情報資源に付加したエージェントやエージェント組織として各AIRが構成される (図1)。こうした機能的な強化・拡張に基づく分散情報資源の構造化をAIR化と呼ぶ。これにより、情報資源自体が能動性・自律性を持つことになり、今まで利用者へ全面的に委ねられていた情報資源の管理や、検索・利用に関わる諸作業の一部を情報資源自身が代行できるようにする^{(9),(10)}。たとえば、利用者からの利用要求メッセージを受けたAIRは、自身が有する利用支援機能・利用支援知識に基づいて、自身が受け持つ情報資源に関する処理を自動的に発動し、利用者に対して能動的に応答したり、他のAIRとの連携・組織化を行うことで、複雑・柔軟な処理を能動的・自律的に代行したりすることが可能となる。

2.2 AIR-NMS

通常、ネットワークを維持・管理するための一連の作業は、ネットワークを構成する各機器から取得するネットワークの状態情報などネットワーク内に分散した種々の情報と、管理者が持つ経験的知識とを用いることで順次処理されていく。たとえば、図2(a)に示すネットワークにおいて、サブネットA内のPCからサブネットB内のサーバへのアクセスに障害が生じた

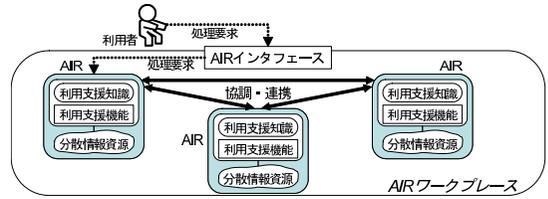


図1 AIRの概念構成図
Fig. 1 Conceptual model of AIR.

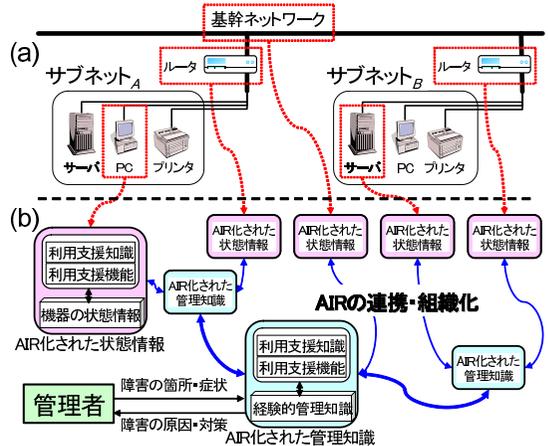


図2 AIR-NMSの概念構成図
Fig. 2 Conceptual model of AIR-NMS.

場合、管理者は、各サブネット・サーバ・PCの状態情報と自らの経験的知識を用いて以下の作業を行う必要がある。

- 構成機器および基幹ネットワークの状況の把握
- 障害の原因の特定
- 障害への適切な対策の決定
- 決定された対策の適用

ネットワークが大規模・複雑になれば、これらの作業を行う管理者の労力は多大なものとなる。また、管理者は、ネットワーク一般に関する高度な経験的知識を有するだけでなく、管理対象に固有の知識 (ネットワーク内の機器構成などに関する最新の知識) にも精通していることが要求される。

このようなネットワーク管理の場面において、ネットワークの状態情報や管理者の経験的知識を分散情報資源と見なし、図2(b)に示すように、それらをAIR化することにより、経験的知識や状態情報を能動的・自律的に動作する情報資源として集積し、ネットワーク管理作業の過程で活用できるようにする。これにより、

- 定型的な管理作業の代行
- 経験的知識の系統的な管理/更新/継承
- ネットワークの構成情報や状態情報の効果的な利用

- 経験的知識や状態情報の分散管理

などが可能となり、高度かつ柔軟なネットワーク管理支援と、管理者の労力の大幅な削減が期待できる。

筆者らは、この考えに基づき、AIR の概念を用いたネットワーク管理支援システム AIR-NMS: AIR-based Network Management Support System を提案している^{6)~8)}。AIR-NMS の主要な機能は、

- 管理者がネットワーク内の障害を発見した場合、管理者からの要求に応じて「状況の把握/原因の特定/対策の決定」を行い、それらの結果を管理者へ通知する機能
- AIR-NMS が監視しているネットワーク構成機器に障害が検知された場合、「状況の把握/原因の特定/対策の決定」を自律的に行い、それらの結果を管理者へ通知する機能

であり、その実現にむけて、筆者らは、以下の 2 種類の AIR について検討している。

- ネットワークの状態情報を AIR 化した I-AIR: Status Information AIR
- ネットワーク管理に関する経験的知識を AIR 化した K-AIR: Management Knowledge AIR

本論文では、特に、前者の I-AIR に焦点を絞り、その実現手法について述べる。

2.3 I-AIR の構成

I-AIR には、ネットワーク内における静的な情報を AIR 化した I_s -AIR: Static I-AIR と、動的な情報を AIR 化した I_d -AIR: Dynamic I-AIR の 2 種類がある。

I_s -AIR は、サブネットの構成情報やアプリケーションの設定情報など、同一のネットワーク内ならば頻繁には更新されない静的な情報を AIR 化したものである。たとえば、サブネットの構成情報を AIR 化した I_s -AIR は、情報資源として、サブネット名、管理者に関する情報、サブネットにおける各種サーバ名や IP アドレスなどの情報を持つ (図 3 (a))。

I_d -AIR は、ネットワークの状態情報など刻々と変化する動的な情報を、SNMP や MIB あるいはサーバのアクセスログなどから獲得し、AIR 化したものである。たとえば、SMTP サーバの状態情報を AIR 化した I_d -AIR は、情報資源として、メールサーバのプロセスに関するログの情報を持つ (図 3 (b))。

これらの I-AIR は利用支援知識・機能により以下を行う。

- 他の AIR からの要求に基づき、自身が持つ情報資源を加工・提供
- 自身が持つ情報資源 (状態情報) を必要に応じて更新

(a) I_s -AIR

```
<? Xml version= "1.0" encoding= "Shift_JIS" ?>
<subnet>
  <subnetName>a_lab</subnetName>      #サブネット名
  <domain>example.jp</domain>        #ドメイン名
  <address>172.20.2.0/24</address>    #アドレス空間
  <gateway>172.20.2.1</gateway>      #ゲートウェイアドレス
  <firewall>active</firewall>        #ファイアウォールの有無
  <server>                             #サーバ情報
    <service>SMTP</service>
    <ipaddress>172.20.0.2</ipaddress>
    <name>smtp_a_lab.example.jp</name>
    <process>Postfix2.1</process>#サーバプログラム
    ....
  </server>
  <server>
    ....
  </server>
</subnet>
```

(b) I_d -AIR

```
Apr 1 01:38:48 mail sendmail[10892]: ruleset-check_rcpt,
  arg1=[noname@noname.example.jp], relay=nohost.example.jp
  [172.20.2.1], reject=550 5.7.1 noname2@noname.example.jp ...
  Relaying denied
Apr 19 20:27:55 mail sendmail[13461]: ruleset-check_rcpt,
  arg1=[noname@noname.example.jp], relay=[172.20.2.2],
  reject=550 5.7.1 noname@noname.example.jp ... Relaying denied.
  IP name lookup failed [172.20.2.2]
Apr 23 16:17:23 mail sendmail[16925]: ruleset-check_rcpt,
  arg1=[noname@noname.example.jp], relay=[172.20.2.2],
  reject=550 5.7.1 noname@noname.example.jp ... Relaying denied.
  IP name lookup failed [172.20.2.2]
```

図 3 I-AIR が持つ情報資源の例

Fig. 3 Example of I-AIR's information resources.

- ネットワーク構成機器の状態の監視

これにより、AIR は他の AIR と能動的・自律的に連携・組織化を行い、管理者の要求に応じて状態情報と経験的知識を関連付けした情報を提供することが可能となる。

2.4 I-AIR の動作

I -AIR は、以下の 2 種類の状態が発生した場合に、その活動を開始する。

- 管理者からの支援要求を受信 (管理者駆動)
- I_d -AIR がネットワークの状態情報から障害を検知 (I_d -AIR 駆動)

管理者による駆動

[Step1] たとえば、“サブネット A 内で noname@example.jp へメールが送信できない”という状況が発生した場合、管理者は AIR インタフェースに対して支援要求を“障害対象”，“障害状況”，“詳細情報”の 3 つの情報に分割して入力する。この例の場合、障害対象は“サブネット A ”，障害状況は“メール，送信できない”，詳細情報は“noname@example.jp”，となる。

[Step2] AIR インタフェースは、入力された支援要求を後述する協調プロトコルの 1 つとして定義された request プロトコルを用いて各 I -AIR に支援要求を送信する。その際、管理者から入力された支援要求の各情報は、request メッセージ内の content データにリ

スト形式で保存される。

[Step3] 支援要求 (障害対象/障害状況/詳細情報) を受信した $I\text{-AIR}_i$ は、自身が持つ障害に関する知識 (3.3 節) と支援要求内の障害対象/障害状況を比較し、自身が担当している情報資源に起因する障害と関連するかを判断する。

[Step4] 障害と関連していると判断された場合、AIR インタフェースに対して要求受理のメッセージを送信した後、情報資源の加工を行い、関連があると判断された情報を“詳細情報”として付加することにより支援要求を詳細化し、request プロトコルを用いて他の $I\text{-AIR}$ に転送する。ここでいう詳細情報とは、本論文で取り上げた例の場合、“サブネット_A”、“メール”、“noname@example.jp” という情報を基に、“MailServer_A IP 172.20.0.2” というように障害を特定するために必要と判断され追加された情報 (本例の場合はサブネット_A のメールサーバの IP アドレス) を指す。

[Step5] 転送された支援要求を受信した $I\text{-AIR}_j$ は、 $I\text{-AIR}_i$ と同様に支援要求の判断を行う。ここで、詳細化が可能と判断した場合、 $I\text{-AIR}_i$ に対して要求受理のメッセージを送信した後、 $I\text{-AIR}_i$ と同様の処理を行う。詳細化が不可能と判断した場合、 $I\text{-AIR}_j$ は処理を終了する。

[Step6] 詳細化処理を行った $I\text{-AIR}_k$ にどの $I\text{-AIR}$ からも要求受理のメッセージが送信されなかった場合、 $I\text{-AIR}_k$ は情報の詳細化が終了したと判断し、後述する inform プロトコルを用いて、現在までに詳細化された情報を要求元の $I\text{-AIR}$ に送信する。このメッセージは詳細化された処理の流れを逆にたどり、AIR インタフェースまで送信される。

[Step7] 詳細化された情報を受信した AIR インタフェースは、その内容をインタフェースを介して管理者に提示する。

このように、 $I\text{-AIR}$ 間で協調・連携し、情報を詳細化することにより、 $I\text{-AIR}$ は障害の原因を特定していく。 $I_d\text{-AIR}$ による駆動

$I_d\text{-AIR}$ はネットワークの状態情報を常時モニタリングしており、状態情報の中に 3.3 節で定義した利用支援知識 OI に記述される「障害と判断するための条件 (obstacle char)」の検出を自律的に実行し、検出回数が「指定された回数 (threshold)」を上回った場合、機器やサーバに障害が発生したと判定する。 $I_d\text{-AIR}$ が機器などの障害を検知した場合、“障害対象”、“障害状況”、“詳細情報”の 3 つの情報を他の AIR へと送信する。その後は、先に述べた管理者による駆動の Step3 以降と同様に動作する。

このように、 $AIR\text{-NMS}$ を導入することで、ネットワーク管理者が行うべき作業は

- (1) $AIR\text{-NMS}$ に対する支援要求の送信
 - (2) $AIR\text{-NMS}$ が提示した対策法の実行
- の 2 つに集約され、管理作業にかかる負担の大幅な削減が期待できる。また、

- $I_d\text{-AIR}$ が機器などの障害を自律的に検知することにより、障害への早期対応が可能
- AIR の協調・連携の履歴を保持することにより、同一の障害への早期対応が可能
- 従来のネットワーク管理業務では困難であった障害原因の特定作業を AIR が代行

など、 $AIR\text{-NMS}$ の持つ特長により、さらなる効果が期待される。

2.5 関連研究

現在、ネットワーク管理支援システムに関して、様々な研究が行われている。しかしながら、たとえば、文献 1)~4) に見られるように、その多くは AS (Autonomous System) 間ネットワークのように構成の変更が頻繁には生じないネットワークを対象としており、その目的も AS 間の通信を維持・運用するためのネットワーク機器の管理など特定の管理業務の支援に限定されている。これに対して、本研究は、構成が頻繁に変更されるネットワーク (たとえば、イントラネットやサブネットなど) での多様な管理業務 (たとえば、WWW や Mail などの利用者に近いアプリケーションサービスに関する障害への対処など) の支援を対象としている。

一方、近年では文献 11)~15) に述べられているように、ネットワーク管理にエージェント技術を応用した研究も見られるようになった。これらの研究では、エージェント技術を用いた自動的な情報収集などが実現されている。しかし、本研究のように収集した情報に基づく処理を自律的に実行して管理者の作業を能動的に支援する手法は提案されていない。

上述したように、本研究では、構成の変更が頻繁に生じるネットワークにおいて多様な管理業務を包括的に支援するために、能動的な情報資源の枠組みに基づいて、ネットワーク状態情報とネットワーク管理知識の双方を効果的に活用した新たなネットワーク管理支援手法の開発を目指している。

3. ネットワークの状態情報の AIR 化

ここでは、 $AIR\text{-NMS}$ の構成要素の 1 つであり、本論文の目的でもある、 $I\text{-AIR}$ を実現するための設計に

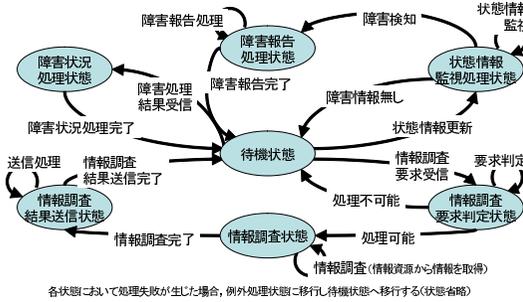


図 4 I-AIR の状態遷移図
Fig. 4 State transition diagram of I-AIR.

ついて述べる。

3.1 I-AIR の内部状態設計

I-AIR は、図 4 に示すようにいくつかの状態を持ち、その状態を遷移させることにより、各処理を進めてゆき、AIR-NMS に対して障害状況の報告や、状態情報の提示を行う。以下にそれぞれの状態について述べる。

- 待機状態 他の AIR からの情報調査要求、状態情報更新、要求した詳細情報を待つ初期状態
- 情報調査要求判定状態 情報調査要求が処理可能かの判定を行う状態
- 情報調査状態 要求された各種情報を加工・抽出する状態
- 情報調査結果送信状態 加工・抽出した情報を、要求元へと送信する状態
- 状態情報監視処理状態 状態情報が更新された際に、その状態情報に障害に関する情報が含まれているかを監視する状態
- 障害報告処理状態 状態情報に障害に関する情報が含まれていた場合、その情報を統合・詳細化し管理者に対して報告を行う状態
- 障害状況処理状態 報告した障害に関する処理結果を受信した場合、その障害に関する後処理を行う状態
- 例外処理状態 発生した例外に関する後処理(他の AIR への処理伝播など)をする状態

3.2 情報資源形式の設計

I-AIR が保持する情報資源には様々な形式のものが考えられる。本論文では、その形式として次の 2 種類の形式に着目して情報資源の設計を行った。

- RDF/XML 形式¹⁶⁾ 静的情報・MIB・一部のログ情報の情報資源を記述する形式として利用

```

I-AIR ::= <ID> <IR> <OI> <CM> <CP>
ID ::= <air id> <workplace id> <task id> <obstacle info>+ <detail info>*
air id = AIR識別子
workplace id = ワークスペース識別子
task id = 処理中のタスクの識別子
obstacle info ::= <obstacle list>* <precheck list>* <info list>*
obstacle list = 処理可能な障害のリスト
precheck list = 処理開始に必要な事前情報のリスト
info list = 詳細化可能な情報のリスト
detail info = 詳細化された情報

IR ::= <type> <path> <format> <time>
type = 情報資源の種類 例:Postfix, MIB
path = パス情報 例:/var/log/maillog
format = ファイルフォーマット 例:XML, Text
time = ファイル最終更新日 例:2004/04/01/11:11

OI ::= <obstacle>* <check time>*
obstacle ::= <obstacle name> <obstacle char>+ <threshold>
obstacle name = 障害名
obstacle char = 障害検知用文字列
threshold = 障害判断用閾値
check time = 障害検知処理の間隔

CM ::= <method> <argument>* <returned value>*
method = 利用支援機能のメソッド名
argument = 利用支援機能の引数値
returned value = 利用支援機能の戻り値

CP ::= <protocolI>+
protocol = 協調プロトコル
    
```

図 5 BNF による I-AIR の知識表現形式の定義
Fig. 5 The knowledge representation scheme of I-AIR in BNF.

(図 3(a)).

- プレーンテキスト形式 Apache や syslog などのログ情報の形式として利用(図 3(b)). I-AIR の持つ利用支援機能により RDF/XML 形式に逐次変換。

3.3 利用支援知識の設計

I-AIR の利用支援知識には

- AIR 識別知識 (AIR-Identifier: ID)
- 情報資源に関する知識 (Information Resource: IR)
- 検知する障害に関する知識 (Obstacle Information: OI)
- 利用支援機能操作の知識 (Control Method: CM)
- 協調プロトコル処理知識 (Communication Protocol: CP)

の 5 つの知識があり、BNF 記法により図 5 のように表現される。

ID の <obstacle info> には、I-AIR が対処可能な障害対象・状況のデータ <obstacle list>、情報調査を開始する際に必要な事前情報のデータ <precheck list>、および、詳細化可能な情報のメタ情報名のデータ <info list> が保持され、図 4 における情報調査要求判定状態において、他の AIR からの要求を判断するための指標の 1 つとして使用される。一方、<detail info> は、これらの情報を要求された場合に自らの持つ情報資源を

加工・操作して詳細化した情報を保持するためのデータである。

IR は、後述する利用支援機能に対して情報資源の加工・操作を依頼する際に使用される $\langle type \rangle$, $\langle path \rangle$, $\langle format \rangle$, $\langle time \rangle$ が記述される。

OI は、I-AIR に付与された経験的知識の一部であり、 I_d -AIR が保持する情報資源から障害を能動的に検知するために用いられるメタ情報として、 $\langle obstacle \rangle$ と $\langle check time \rangle$ が記述される。 $\langle obstacle \rangle$ はさらに $\langle obstacle name \rangle$, $\langle obstacle char \rangle$, $\langle threshold \rangle$ の3つの要素を持ち、図4における状態情報監視処理状態において使用される。また、 $\langle check time \rangle$ は I_d -AIR が情報資源の更新状況を監視するための間隔が記述される。

CM は、利用支援機能に対して情報資源の加工・操作処理を依頼する際のメソッドが記述され、図4の情報調査状態および状態情報監視処理状態において、利用支援機能呼び出す際に用いられる。

CP では、AIR 間の協調プロトコルに関する知識が記述される。協調プロトコルの設計や記述方法に関しては3.5節で述べる。

3.4 利用支援機能の設計

利用支援機能とは、他のAIRの要求に従って自身の持つ情報資源を加工・操作するための機能である。利用支援機能には

- 利用支援知識とのインタフェースとしての機能
- 情報資源を加工・操作するための機能
- 他のAIRに加工した情報資源を送信する機能
- 閾値または、キーワードによる障害の検知機能が必要とされる。

このような機能を持つ利用支援機能は、AIRが扱う情報資源により様々な処理を行うことになる。しかしながら、利用支援機能への要求送信方法が複雑になると、AIRの概念が持つ汎用性が損なわれる可能性がある。そのため、本論文では、利用支援知識とのインタフェース機能として、図5に示したように、

CM ::= $\langle method \rangle \langle argument \rangle^* \langle returned value \rangle^*$ に準ずるものとして $\langle method \rangle = RequestInformation$, $\langle argument \rangle = String[]$, $\langle returned value \rangle = String[]$ を定義した。

3.5 AIR間協調プロトコルの設計

本論文ではAIR間の協調・連携に用いるプロトコルとして3つのプロトコルを定義した。以下に、それぞれのプロトコルについて説明する(図6)。

- request プロトコル 管理者からの支援要求をAIRインタフェース~I-AIR間、およびI-AIR間

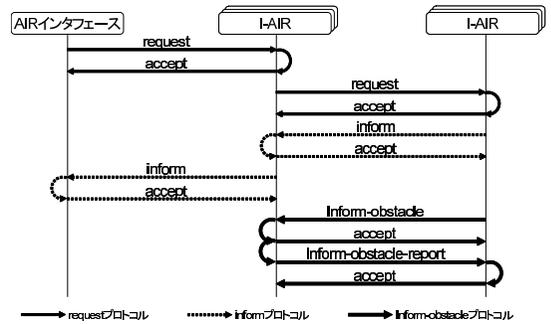


図6 AIR-NMS プロトコルシーケンス
Fig.6 Protocol sequence of AIR-NMS.

```

(Msg :performative request :from <air id> :to <air id> :taskid <taskid>
 :content ((Obstacle :target <target list>+
 :obstacle <obstacle list>+
 :detail <detail list>+))
 (DetailInfo :detailInfo <detail info>*))
  
```

図7 request メッセージの構造
Fig.7 Construction of request message.

に送信する際に用いるプロトコル。

- inform プロトコル I-AIR が特定した障害原因をAIRインタフェースを介してに送信する際に用いるプロトコル。
- inform-obstacle プロトコル I_d -AIR が障害を検知したときにその障害情報を他のAIRに送信する際に用いるプロトコル。

また、例として request プロトコルの request メッセージの構造を図7に示す。

request メッセージを受信したI-AIRは、まずメッセージ中の $\langle taskid \rangle$ を確認し、すでに自身が詳細化を行ったタスクであるかどうか調査する。自身が詳細化を行ったタスクである場合、メッセージを破棄し待機状態に遷移し、再帰的なメッセージの伝播を回避する。一方、詳細化を行ってないタスクの場合、メッセージ中の $\langle target list \rangle$, $\langle obstacle list \rangle$, および $\langle detail list \rangle$ とI-AIRが知識として持つ $\langle obstacle list \rangle$ (図5)を照合し、それらが一致する場合に対処可能であると判断する。そして、対処可能な場合には accept メッセージを送信元に返信し、対処不可能の場合にはメッセージを破棄し、その後待機状態に遷移する。一方、詳細化可能な場合には、3.3節で述べたように $\langle detail info \rangle$ を生成し、次に、他のI-AIRに対して、元のメッセージに $\langle detail info \rangle$ を付加した request メッセージを送信し、さらに、詳細化済みのタスクとしてその $\langle taskid \rangle$ を記録する。もしも、一定時間内に request メッセージを送信したI-AIRらから、accept メッセージ

ジが返送されてこなければ、システムにおいて当該情報の詳細化がこれ以上不可能である（すなわち、すべての I-AIR が詳細化済みか詳細化不可能）と判断し inform プロトコルを用いて、自身が accept メッセージを送信した I-AIR に対して詳細化した情報を送信する。

4. 試作システムと実験例

ここでは、本論文で提案・設計した I-AIR の妥当性を検証するための試作システムと実験例について述べる。

4.1 I-AIR の実現方法

I-AIR は、ルール型の知識に基づき自律的・能動的に活動するプログラムとして実装される。このような AIR の実現方法として、マルチエージェントシステムを用いる方法が提案されている⁶⁾⁻¹⁰⁾。これは、AIR の持つ

- 知識に基づく動作制御
- 外部からの要求・イベントへの知的な反応
- 複数の AIR の協調・連携に基づく問題解決

などの特徴を実現するうえで、マルチエージェントシステムが提供する機能や動作特性が効果的に活用できることによる。

そこで、本論文では、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^{17),18)}、および、IDEA 開発環境¹⁹⁾ を用い、AIR-NMS を実装した。すなわち、AIR-NMS を構成する AIR は、ルール型知識として与えられる利用支援知識に基づいて動作し、その過程で、利用支援機能として組み込まれた Java プログラムなども起動しながら、情報資源（経験的管理知識/ネットワークの状態情報）の加工処理や他の AIR との協調・連携処理を実行する。

4.2 実験例

実験では、I-AIR の動作を検証するために、図 8 に示すように、基幹ネットワーク、および 2 つのサブネットから構成されるネットワークを構築した。

図 8 に示したように、実験ネットワーク環境では、基幹ネットワークには、グローバルアドレス空間 (XXX.YYY.ZZZ.0/26) が割り当てられ、サブネット A は、サーバ類が設置されており、グローバルアドレス空間 (XXX.YYY.ZZZ.64/26) が割り当てられている。このサブネット A には、基幹ネットワークを総轄的に管理するネットワーク管理者が存在し、この管理者が AIR-NMS を用いてネットワーク管理を行う。また、Mail サーバ A および PC_{A1} 上には AIR

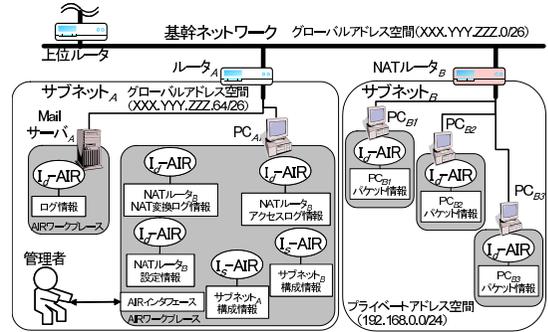


図 8 ネットワークと AIR-NMS の構成
Fig. 8 Construction of network systems and AIR-NMS.

が動作する場である AIR ワークスペースが実現されており、各 AIR ワークスペースには AIR-NMS を構成する I-AIR が存在している。サブネット B には、利用者用の端末が設置されており、プライベートアドレス空間 (192.168.0.0/24) が割り当てられている。ルータ B には、NAT (Network Address Translation) 機能が実現されており、サブネット B 外へのアクセスに対して、セッションごとに XXX.YYY.ZZZ.32 ~ XXX.YYY.ZZZ.62 の範囲内で動的にグローバルアドレスを割り振り接続処理を行い、NAT 変換や接続情報のログなどの状態情報を蓄積する。

また、本論文では以下の 3 つの実験例について検討した。

実験例 1 ネットワークから noname@example.jp 宛にウイルスを含むメールが送信された際のウイルス送信端末特定

実験例 2 NAT 環境下 (サブネット B 内) の端末がトロイの木馬型ワームに感染し攻撃をしている際の端末特定

実験例 3 新たにサブネット B に追加された端末が外部ネットワークに接続することができない際の原因特定

そして、それぞれのシナリオに必要な AIR を以下のように実装した。

- サブネット A の機器構成情報を持つ I_s-AIR₁
- サブネット B の機器構成情報を持つ I_s-AIR₂
- Mail サーバ A のログ情報を持つ I_d-AIR₁
- NAT ルータ B の NAT 変換ログ情報を持つ I_d-AIR₂
- NAT ルータ B のアクセスログ情報を持つ I_d-AIR₃
- NAT ルータ B の設定情報を持つ I_d-AIR₄
- サブネット B の PC_{B1}, PC_{B2}, PC_{B3} が送受信したパケット情報を持つ I_d-AIR₅, I_d-AIR₆, I_d-AIR₇

表 1 各 I-AIR の利用支援知識
Table 1 Knowledge of utilization supporting of I-AIR.

I-AIR名	要素名	値	I-AIR名	要素名	値
I _s -AIR ₁	Obstacle list	Mail, SubnetA	I _d -AIR ₂	Obstacle list	ALL
	Precheck list	Null		Precheck list	Ipaddress, Time, TargetIP
	Info list	MailServerName, MailServerIP		Info list	ClientIP
I _s -AIR ₂	Obstacle list	SubnetB	I _d -AIR ₃	Obstacle list	ALL
	Precheck list	Null		Precheck list	TargetIP, Time, Service
	Info list	ClientIP		Info list	ClientIP
I _d -AIR ₁	Obstacle list	Mail, Virus	I _d -AIR ₄	Obstacle list	NATRouterB
	Precheck list	MailServerName, MailServerIP, MailAddress		Precheck list	<Config Keyword>
	Info list	SourceIP, Time		Info list	<Config Parameter>
			I _d -AIR _{5~7}	Obstacle list	Worm
				Precheck list	Time
				Info list	TargetIP, Time, Port

これらの AIR のうち、サブネット_A およびサブネット_B の構成情報を持つ I_s-AIR₁ および I_s-AIR₂ は、図 3(a) に示すように RDF/XML に準拠した形式で記述された情報資源を持ち PC_{A1} 上に実現される。一方、サブネット内の Mail サーバ_A のログ情報を持つ I_d-AIR₁ は Mail サーバ_A 上に存在し、NAT ルータ_B の NAT 変換ログ情報、アクセスログ情報、および設定情報を持つ I_d-AIR₂, I_d-AIR₃, および I_d-AIR₄ はサブネット_A の PC_{A1} 上に、PC_{B1}, PC_{B2}, PC_{B3} が送受信したパケット情報を持つ I_d-AIR₅, I_d-AIR₆, I_d-AIR₇ はそれぞれの PC 上に実現される。また、本例題において設定した各 I-AIR の知識のうち、図 5 で示した 〈obstacle info〉 についての設定内容を、表 1 に示す。なお、各 I-AIR の持つ情報資源は今回の実験用に用意したものであり、規模も最小限のものとした。

4.3 実験例 1 の詳細と動作結果

実験例 1 では、“ネットワークから noname@example.jp 宛にウイルスを含むメールが送信された” という障害を考え、管理者が “サブネット_A もしくはサブネット_B 内の、どの端末から noname@example.jp 宛へウイルスを含むメールが送信されたか特定したい” と思い、本システムを使用し、結果 “サブネット_B 内の PC (PC_{B1}, PC_{B2}, PC_{B3}) から noname@example.jp 宛へウイルスを含むメールが送信された” という応答が得られる状況を設定した。

通常のネットワーク管理手法において上記の要求が発生した場合、管理者は

- (1) 管理ネットワーク内にある Mail サーバの特定
- (2) ウィルスメールが Mail サーバ_A を経由して送信されたメールであることをログを用いて確認
- (3) Mail サーバ_A を経由したものである場合、ログから当該メールが送信された時刻や送信元 IP



図 9 AIR インタフェースの要求入力画面

Fig.9 Screen shot of request-inputs into AIR interface.

アドレスなどを特定

- (4) ウィルスメールのヘッダ情報が取得可能であれば、ヘッダ情報より当該ウィルスメールが送信された時刻を特定
- (5) NAT ルータ_B の NAT 変換ログから当該時刻に当該 IP アドレスに変換された PC を特定
- (6) 当該 PC が複数台あった場合、NAT ルータ_B のアクセスログから当該セッション (Mail 接続) を行った PC を特定

という作業が必要になり、その負荷も非常に高いといえる。

これらの作業を AIR-NMS を用いて支援を行った際の処理の流れを以下に示す。

[Step1] 管理者は “サブネット_A もしくはサブネット_B 内の、どの端末から noname@example.jp 宛へウイルスを含むメールが送信されたか特定したい” という要求を “障害対象”, “障害状況”, “詳細情報” の 3 つの情報に分割し、PC_{A1} の AIR インタフェースに対して障害対象として “SubnetA SubnetB”, 障害状況として “Virus Mail”, 詳細情報として “noname@example.jp” を入力する (図 9)。

[Step2] 要求を受け取った AIR インタフェースは “SubnetA SubnetB”, “Virus Mail”, “non

```
Msg :performative request :from AIR-Interface :to broadcast
      :taskid 20040404162433
      :content (Obstacle :target "SubnetA SubnetB" :obstacle "Virus Mail"
              :detail "noname@example.jp")
```

図 10 Request メッセージ
Fig.10 Request message.

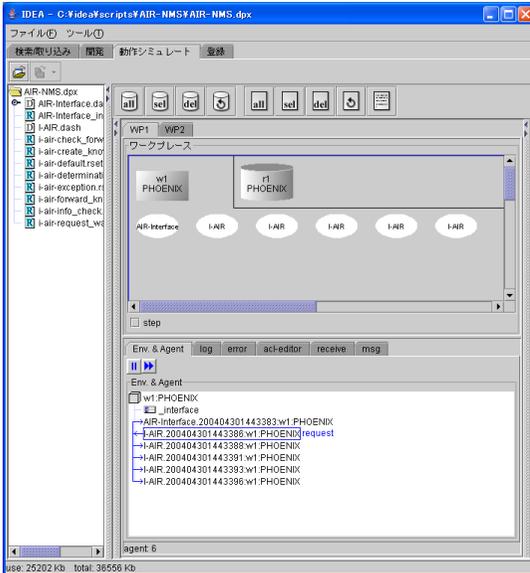


図 11 AIR-Workplace の動作画面
Fig.11 Screen shot of AIR workplace.

ame@example.jp” という要求を request プロトコルを用いて AIR ワークスペース内の I-AIR に送信する (図 11)。上記の request メッセージの詳細な構造を図 10 に示す。

[Step3] AIR インタフェースからの要求を受け取った I-AIR は、それが保持している (obstacle list) と (precheck list) とを request メッセージ内の “SubnetA SubnetB”, “Virus Mail”, および “noname@example.jp” の各情報と比較することにより、要求が対処可能かどうかを判定する。この場合、 I_d -AIR は AIR インタフェースから受け取った要求に含まれる情報が表 1 に示した (precheck list) を満たしていないため反応しない。一方、 I_s -AIR_A は (obstacle list) が “Mail”, (precheck list) が NULL であるため反応し、(info list) を用いて管理しているネットワークの Mail サーバの名前と IP アドレスを (detail info) として付加する。この際、request プロトコルの処理に基づき、accept メッセージを要求元 (AIR インタフェース) に対して送信する。

[Step4] I_s -AIR_A は (detail info) を付加した request メッセージを他の AIR に送信する。

[Step5] I_d -AIR は新たに加わった (detail info) と



図 12 AIR インタフェースの結果表示画面
Fig.12 Screen shot of result of AIR interface.

(precheck list) を比較し、要求が対処可能かどうかを判定する。

[Step6] 対処可能である場合、 I_d -AIR は accept メッセージを要求元に送信し、(info list) を用いて (detail info) を新たに付加し、request メッセージを他の AIR に送信する。

[Step7] Step3 から Step6 を繰り返すと、request メッセージを送信しても accept メッセージが返信されてこなくなる。I-AIR は、一定時間 (今回の実験では 5 秒に設定した) を過ぎても accept メッセージが返信されない場合、その時点で処理を打ち切り、その I-AIR が现阶段で提供可能な情報を統合し、AIR インタフェースに対して inform メッセージを送信する。

[Step8] I-AIR から inform メッセージを受け取った AIR インタフェースは、提供された情報をまとめあげ、管理者に対して結果を表示して処理を終える (図 12)。

以上のように、AIR-NMS を用いることで、管理者は AIR インタフェースを介して要求を伝えるだけで原因の特定と対策方法が得られる。

4.4 実験例 2 の詳細と動作結果

実験例 2 では、“NAT 環境下の端末がトロイの木馬型ワームに感染し攻撃をしている際の端末特定” という状況を取り上げた。本実験では、トロイの木馬型ワームとして MSBlaster を想定し、サブネット_B 内に新しく接続された端末 PC_{B4} が感染して他の端末に攻撃をかけているとした。そして、攻撃を受けた端末からアラートが発生し、管理者に対して “トロイの木馬型ワーム (MSBlaster) に感染した端末が発見された” と報告する状況を設定した。

通常のネットワーク管理手法において上記の要求が発生した場合、管理者はある程度被害が大きくなるまでトロイの木馬型ワームに感染している端末がサブネット内に存在することに気がつかない場合が多い。さらに、発見した後も攻撃元端末が NAT 環境下にある場合、その端末の特定を行うために、NAT ルータ_B の NAT 変換ログや攻撃を受けている端末のログなどを確認する作業が必要になる。

これらの作業を AIR-NMS を用いて支援を行った際の処理の流れを以下に示す。

[Step1] PC_{B4} のトロイの木馬型ワーム (MS-Blaster) による攻撃を受けた PC_{B1} の I_d -AIR₅ は、自身の知識である 〈OI〉 内の 〈check time〉 に従い、その間隔で PC_{B1} に対する特定のポート (今回は TCP/135 番ポート) へのアクセス数を監視する。このアクセス数が閾値 〈threshold〉 として設定した値を超えることで MSBlaster の攻撃の可能性を検知する。攻撃の可能性を検知した場合、まず、他の MSBlaster に感染した PC の有無を調べるために各 I-AIR に “Worm”, “〈time〉 (検知した時刻)”, “UnderAttack (攻撃の有無)”, “Port TCP/135” という inform-obstacle メッセージを送信する。

[Step2] I_d -AIR₅ からの inform-obstacle メッセージを受信した各 I-AIR は、自らの持つ利用支援知識 〈obstacle list〉, 〈precheck list〉 の値と、送られてきたメッセージ内の値と一致するか否かで詳細な情報が提供可能かを判定する。本実験では I_d -AIR₂, I_d -AIR₆, I_d -AIR₇ が詳細化可能と判断する。

[Step3] I_d -AIR₂ では、〈time〉 に NAT ルータ_B を介して PC_{B2} に送信されたパケットの ClientIP が、 I_d -AIR₆ および I_d -AIR₇ では、同様のアクセスが情報資源に見られるため、そのアクセスに関する情報 (TargetIP, Time, Port) を Inform-obstacle-report メッセージを用いて I_d -AIR₅ に送信する。

[Step4] Inform-obstacle-report メッセージを受信した I_d -AIR₅ は、一定時間 (今回の実験では 5 秒に設定した) を過ぎた段階で処理を打ち切り、得られた情報を統合し、AIR インタフェースに対して inform メッセージを送信する。

[Step5] I-AIR から inform メッセージを受け取った AIR インタフェースは、提供された情報をまとめあげ、管理者に対して結果を表示して処理を終える。管理者に提示される情報は “ワーム警告 発信元: PC_{B4} , 発信元 IP アドレス: 192.168.0.XXX, 被攻撃 PC: PC_{B1} , PC_{B2} , PC_{B3} ” となる。

以上のように、AIR-NMS を用いてトロイの木馬型ワームの存在を監視させることで、ワームの発見から特定までを自動で行うことが可能となり、管理者の負担も軽減される。

4.5 実験例 3 の詳細と動作結果

実験例 3 では、“新たにサブネット_B に追加された端末が外部ネットワークに接続することができない際の原因特定” という状況を取り上げた。本実験では、追加された端末は、 PC_{B4} : IP アドレス 192.168.0.128

であるとし、接続ができない原因として、NAT ルータ_B の設定が既存端末 ($PC_{B1} \sim PC_{B3}$, IP アドレス 192.168.0.11 ~ 192.169.0.13) のみが外部ネットワークに接続可能となっている状況を設定した。

通常のネットワーク管理手法において上記のような要求が発生した場合、管理者は上位ルータや NAT ルータ_B が動作しているかの確認を行い、動作している場合は、各ルータの設定やログの確認を行う。また、他に影響が出ている範囲などの特定を行い、影響が出ている場合は、その事実も考慮して原因を特定してゆく必要があり、その負荷も非常に大きいといえる。

これらの作業を AIR-NMS を用いて支援を行った際の処理の流れを以下に示す。

[Step1] 管理者は “サブネット_B の端末 PC_{B4} : IP アドレス 192.168.0.128 が外部ネットワークに接続できない原因の特定をしたい” という要求を “障害対象”, “障害状況”, “詳細情報” の 3 つの情報に分割し、 PC_{A1} の AIR インタフェースに対して障害対象として “SubnetB”, 障害状況として “Outgoing Failure”, 詳細情報として “PCb4 192.168.0.128” を入力する。

[Step2] 要求を受け取った AIR インタフェースは “SubnetB”, “Outgoing Failure”, “PCb4 192.168.0.128” という要求を request プロトコルを用いて AIR ワークスペース内の I-AIR に送信する。

[Step3] AIR インタフェースからの要求を受け取った I-AIR は、それが保持している 〈obstacle list〉 と 〈precheck list〉 とを request メッセージ内の各情報と比較することにより、要求が対処可能かどうかを判定する。本例題の場合、 I_s -AIR_B は 〈obstacle list〉 が “SubnetB”, 〈precheck list〉 が NULL であるため反応し、accept メッセージを要求元に送信した後、サブネット_B を構成している機器、NAT ルータ_B, PC_{B1} , PC_{B2} , PC_{B3} の情報を 〈detail info〉 に “NATRouterB, 192.168.0.1, PCb1, 192.168.0.11, PCb2, 192.168.0.12, PCb3, 192.168.0.13” として付加し、他の AIR に request メッセージを送信する。

[Step4] I-AIR は新たに加わった 〈detail info〉 と 〈precheck list〉 を比較し、要求が対処可能かどうかを判定する。本例題の場合、新たに 〈detail info〉 として加わった “NATRouterB” により I_d -AIR₄ が反応する。

[Step5] I_d -AIR₄ は、自身が持つ NAT ルータ_B の設定情報からメッセージ内のキーワードに一致する項目に関する情報を抜き出す。本例題の場合、“Outgoing Failure” というキーワードからネットワークへの

接続に関する情報 (outgoing の情報) を (detail info) として新たに付加し, request メッセージを他の AIR に送信する. 本例題では, request メッセージを送信しても accept メッセージが返信されてこなくなるため, 処理を打ち切り, 情報を統合し, AIR インタフェースに対して inform メッセージを送信する.

[Step6] I-AIR から inform メッセージを受け取った AIR インタフェースは, 提供された情報をまとめあげ, 管理者に対して結果を表示して処理を終える. 本例題の場合, NAT ルータ_B における接続設定の情報 (Outgoing) が管理者に提示され, 管理者は提示された情報を確認し, 設定の変更を NAT ルータ_B に施すことで対処完了となる.

以上のように, AIR-NMS を用いることで, 障害の原因の特定に必要な情報の自動的な収集・整理が可能となり, 管理者の負担も軽減されている.

4.6 考 察

I-AIR を協調・連携させることで, 通常はネットワーク管理者が行っていた作業の一部が自動化され, 管理者の作業量が削減できることを確認した.

一方, 試作システムで実装した I-AIR の動作制御のルールは 63 個であり, これがすべての I-AIR に埋め込まれている. そして, 開発者は, 各 I-AIR が担当する情報資源に応じて, 表 1 に示した利用支援知識を設定し, 固有の利用支援機能を追加するだけで, 種々の I-AIR を容易に実現することができる. これは, 多数の I-AIR の設計・実装を行ううえでの利点といえる. さらに, 試作システムにおいて管理者からの要求を発信して結果が表示されるまでに要した時間は最終結果前のタイムアウト値を除けば平均 3 秒ほどであった. こうした問題解決に要する時間は, 取り扱う問題の複雑さや情報資源の規模, あるいは, 分散環境におけるネットワーク遅延などにより変動するが, 本実験の結果は, 同様の状況下で作業する管理者にとって, おおむね許容できる範囲に収まっていると考える.

また, 本試作システムでは, 管理者からの要求 “サブネット_A もしくはサブネット_B 内の, どの端末から noname@example.jp 宛へウイルスを含むメールが送信されたか特定したい” という要求を, AIR インタフェースで, “障害対象: SubnetA SubnetB, 障害状況: Virus Mail, 詳細情報: noname@example.jp” としてマッピングを行った. なお, このマッピングでは, 管理者が発した要求の細かいニュアンス (“どの端末”・“送信されたか”・“特定したい” など) を AIR-NMS まで伝えることは困難であるが, こうしたインタラクション形態の改良・高度化については今後の課

題としたい.

5. ま と め

本論文では, AIR の概念を導入したネットワーク管理支援システム AIR-NMS においてネットワークの状態情報を持つ I-AIR: Status Information AIR の設計と実装について述べた.

AIR-NMS は, ネットワークシステムの維持・管理に必要な一連の作業を部分的に代行することにより, ネットワーク管理者の労力を大幅に削減することを目的として提案されたシステムである. 本論文では, I-AIR を中心とした支援の枠組みに焦点をあてて議論したが, 状態情報を能動的に活用する I-AIR の導入により, 従来であれば管理者が行わなければならなかった各種ネットワークの状態情報の収集・蓄積・加工などの処理を, I-AIR が管理者に代わって行うことができるようになり, 管理者の作業負荷の軽減が可能となる.

また, 本論文で提案した I-AIR は, 取り扱う問題に応じて自由に定義・構成することができ, 様々なタイプの I-AIR を容易に構成することが可能となり, I-AIR の形式で利用する状態情報の種類や内容を拡大することにより, さらなる管理者の負荷の軽減が期待できる.

今後, 提案手法に基づく実用的な知的管理支援ツールの実現を目指して, AIR-NMS における専門家知識の供給源となる K-AIR を含めた AIR 相互の協調・連携手法を中心に, 実環境での実験を含めた検討を継続してゆく.

参 考 文 献

- 1) Consens, M. and Hasan, M.: Supporting network management through declaratively specified data visualizations, *IEEE/IFIP 3rd International Symposium on Integrated Network Management*, pp.725-738 (1993).
- 2) Hasan, M., Sugla, B. and Viswanathan, R.: A conceptual framework for network management event correlation and filtering systems, *IEEE/IFIP 6th International Symposium on Integrated Network Management*, pp.233-246 (1999).
- 3) Virmani, A., Lobo, J. and Kohli, M.: Netmon: Network management for the SARAS softswitch, *IEEE/IFIP Network Operations and Management Symposium*, pp.803-816 (2000).
- 4) Damianou, N., Dulay, N., Emil, L., Sloman, M. and Tonouchi, T.: Tools for domain-based policy management of distributed systems, *IEEE/IFIP Network Operations and Manage-*

- ment Symposium, pp.203–218 (2002).
- 5) 木下哲男：分散情報資源活用の一手法 — 能動的情報資源の設計, 信学技報 AI99-54, pp.13–19 (1999).
 - 6) 今野 将, 加藤真也, 岩谷幸雄, 阿部 亨, 木下哲男：能動的情報資源の概念に基づくネットワーク管理支援システムの設計, 情報処理学会 DPS 研究報告 2003-DPS-114, pp.189–194 (2003).
 - 7) 今野 将, 加藤真也, 岩谷幸雄, 阿部 亨, 木下哲男：能動的情報資源を用いたネットワーク管理支援システム, FIT2003 情報技術レターズ LG-008, pp.135–137 (2003).
 - 8) Konno, S., Iwaya, Y., Abe, T. and Kinoshita, T.: Design of Network Management Support System based on Active Information Resource, *18th International Conference on Advanced Information Networking and Applications*, pp.102–106 (2004).
 - 9) Li, B., Abe, T., Sugawara, K. and Kinoshita, T.: Active information resource: Design Concept and Example, *17th International Conference on Advanced Information Networking and Applications*, pp.274–277 (2003).
 - 10) Li, B., Abe, T. and Kinoshita, T.: Design of agent-based active information resource, *1st International Conference on Agent-Based Technologies and Systems*, pp.233–244 (2003).
 - 11) Martin-Flatin, J.-P., Znaty, S. and Hubaux, J.-P.: A Survey of Distributed Enterprise Network and Systems Management Paradigms, *Network and Systems Management*, Vol.7, No.1, pp.9–26 (1999).
 - 12) Cabri, G., Leonardi, L. and Zambonelli, F.: Network Management based on Mobile Agents using Programmable Tuple Spaces, *4th International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents* (1999).
 - 13) Satoh, I.: Building Reusable Mobile Agents for Network Management, *IEEE Trans. Systems, Man and Cybernetics*, Vol.33, No.3, pp.350–357 (2003).
 - 14) Constantinescu, Z.: Towards an Autonomic Distributed Computing System, *14th International Workshop on Database and Expert Systems Applications*, pp.1–5 (2003).
 - 15) Stephan, R., Ray, P. and Paramesh, N.: Network Management Platform based on Mobile Agent, *International Journal of Network Management*, Vol.14, pp.59–73 (2003).
 - 16) RDF/XML Syntax Specification (Revised) W3C Working Draft 23 January 2003. <http://www.w3.org/TR/rdf-syntax-grammar/>
 - 17) 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 (1996).

- 18) DASH — Distributed Agent System based on Hybrid architecture. <http://www.agent-town.com/dash>
- 19) 高垣 暁, 打矢隆弘, 原 英樹, 阿部 亨, 木下哲男：エージェントシステムのためのインタラクティブな開発支援環境, FIT2003 情報技術レターズ LG-010, pp.143–145 (2003).

(平成 16 年 5 月 13 日受付)

(平成 16 年 11 月 1 日採録)



今野 将 (正会員)

1973 年生。2001 年千葉工業大学大学院博士後期課程情報工学専攻期間満了退学。同年東北大学電気通信研究所助手。2004 年東北大学情報シナジーセンター助手。博士 (工学)。

エージェント指向コンピューティング, マルチエージェントシステム, ネットワーク管理支援システムの研究開発に従事。2002 年度電子情報通信学会総合大会学術奨励賞, 2003 年度 FIT2003 論文賞各受賞。電子情報通信学会, AAAI, IEEE 各会員。



吉村 智志

1981 年生。2003 年東北大学工学部卒業。同年同大学大学院情報科学研究科博士前期課程進学, 現在に至る。マルチエージェント技術を用いたネットワーク管理者支援の研究に従事。電子情報通信学会会員。



羽鳥 秀明

1981 年生。2004 年東北大学工学部卒業。同年同大学大学院情報科学研究科博士前期課程進学, 現在に至る。マルチエージェント技術に基づく知識管理の研究に従事。電子情報

通信学会会員。



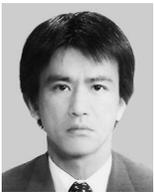
岩谷 幸雄（正会員）

1967年生。1993年東北大学大学院博士課程前期2年の課程修了。同年秋田大学鉱山学部助手。2000年同大学工学資源学部講師。2002年東北大学電気通信研究所助教授。博士（情報科学）。ネットワーク管理，マルチメディア信号処理，バーチャルリアリティの研究開発に従事。1996年度電気学会論文発表賞，2003年度FIT2003論文賞，2004年度石田（實）記念財団研究奨励賞各受賞。電子情報通信学会，日本音響学会，バーチャルリアリティ学会，IEEE，アメリカ音響学会各会員。



木下 哲男（正会員）

1953年生。1979年東北大学大学院修士課程修了。同年沖電気工業（株）入社。1996年東北大学電気通信研究所助教授。2001年同大学情報シナジーセンター教授。知識工学，エージェント技術，知識情報システム等の研究開発に従事。工学博士。1989年度情報処理学会研究賞，1996年度同論文賞，2000年度電子情報通信学会業績賞，2003年度FIT2003論文賞各受賞。電子情報通信学会，人工知能学会，日本認知科学会，AAAI，IEEE，ACM各会員。



阿部 亨（正会員）

1962年生。1990年東北大学大学院博士後期課程単位取得退学。同年同大学情報処理教育センター助手。1993年北陸先端科学技術大学院大学情報科学研究科助教授。2001年東北大学情報シナジーセンター助教授。工学博士。パターン認識・知識処理の研究に従事。2003年度FIT2003論文賞受賞。電子情報通信学会，人工知能学会，IEEE各会員。
