LF-009

# 能動的情報資源を用いた自律的なネットワーク監視システム
# Autonomous Network Monitoring System Based on Active Information Resource

今野 将* 吉村 智志† 岩谷 幸雄‡ 阿部 亨* 木下 哲男*
Susumu Konno Satoshi Yoshimura Yukio Iwaya Toru Abe Tetsuo Kinoshita

## 1. まえがき

近年，ネットワーク管理者がネットワークシステムの監視・維持・管理を行うために必要な労力や専門的知識は増加・高度化の一途を辿っている．現在，この問題に対処するために，いくつかのネットワーク管理支援システムが提案・商品化されている [1]〜[6]．しかしそれらの多くは，監視・維持・管理に必要な機器の状態情報や一般的な対応策を管理者へ提示するに留まり，情報の総合的な判断や具体的対策の決定は依然として管理者の側に委ねられている．

これに対し筆者らは，能動的情報資源 (Active Information Resource: AIR)[7] の概念を各機器の状態情報や管理に関する諸知識へ適用することで，これらを自律的に連携・協調させ，ネットワークの管理を支援するシステム (AIR-based Network Managment Support System: AIR-NMS) の提案をしている．本稿では，この AIR-NMS の一要素であり，ネットワーク監視に重点をおいた能動化された状態情報エージェント (Status Information AIR: I-AIR) の設計・試作・実験を行い，I-AIR を用いたネットワーク監視システムによる，ネットワーク管理者の負荷軽減について議論する．

## 2. 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 (AIR-NMS)

### 2.1 能動的情報資源 (AIR)

能動的情報資源 (AIR) は，情報資源の構造を強化することで，利用者の要求へ各情報資源を能動的・自律的に対応させ，情報資源のより高度な活用を図る機構である．具体的には，各情報資源（コンテンツ）に利用支援知識および利用支援機能を付加したエージェントとして AIR を構成し，利用支援知識・機能を用い AIR 相互間で連携・協調処理を行わせることにより，利用者からの処理要求（例えば，コンテンツ検索・統合・分析など）を AIR 側（すなわちコンテンツ側）で自律的に実行させるものである．このとき，AIR が実際に活動する作業空間を AIR ワークプレースと呼び，利用者からの処理要求は AIR インタフェースを介してワークプレース内の各 AIR へ伝達される．

### 2.2 AIR によるネットワーク管理支援

通常，ネットワークシステムを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の状態や履歴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．例

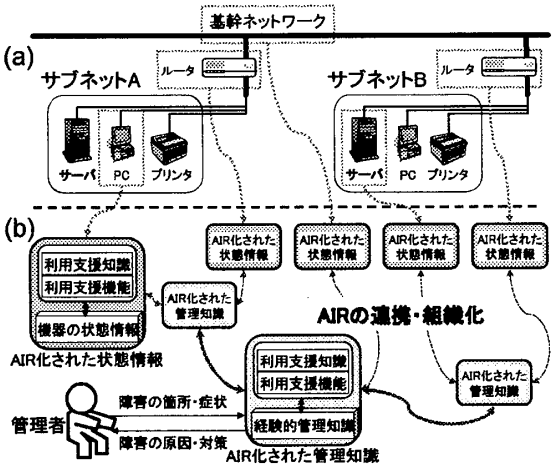


図 1: ネットワークシステムと AIR-NMS

えば，図 1 (a) に示すネットワークシステムにおいて，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場合，現状では，管理者が自らの経験的知識を用いて以下の作業を行う必要があり，その労力はネットワークの規模に応じて膨大なものとなる．

**作業 1** PC・ルータ・基幹ネットワークの状態情報の収集
**作業 2** 収集された状態情報の統合
**作業 3** 障害の原因の特定
**作業 4** 障害への適切な対策の決定
**作業 5** 決定された対策を実際に適用

このようなネットワークシステム管理の場面で，各機器の状態情報や知識ベースに蓄積された管理者の経験的知識を情報資源とみなし AIR 化すれば（図 1 (b)），管理作業の大部分（前述の例ならば作業 1〜4）を AIR の連携・協調処理により自律的に実行させることができ，管理者の労力を大幅に削減することが可能となる．また，AIR の導入により，経験的知識の継承や修正・追加，あるいは機器構成の変更への対応が容易になるため，より高度かつ柔軟なネットワーク管理が実現できる．

筆者らは，このような考えに基づくネットワーク管理支援システムとして AIR-NMS を提案しており，AIR-NMS の構成要素として，管理者の経験的知識を持つ AIR を K-AIR (Management Knowledge AIR)，ネットワーク構成機器の状態情報を持つ AIR を I-AIR とし設計を行ってきた．本稿では，このうち状態情報を能動化した I-AIR に焦点を絞りを，I-AIR を用いたネットワーク監視システムについての設計・試作を行う．

*東北大学情報シナジーセンター，Information Synergy Center, TOHOKU Univ.
†東北大学大学院情報科学研究科（現在，日立製作所ソフトウェア事業部 所属），Graduate School of Information Sciences, TOHOKU Univ. (Software Division of Hitachi, Ltd.)
‡東北大学電気通信研究所，Research Institute of Electrical Communication, TOHOKU Univ.

### 2.3 I-AIR による自律的なネットワーク監視システム

AIR-NMS の考えに基づき実現された I-AIR は各々の協調・連携により，以下のことが可能となり，自律的なネットワーク監視システムを実現する．

- 管理作業の一部を AIR 側で代行
- ネットワークの状態情報の分散管理や効果的な利用

ネットワーク上の状態情報を AIR 化した I-AIR は，自身の保持している情報資源に働きかけることによって定期的にネットワークの状態の調査を行なう．一般的に，ネットワークを管理する方法といえば，管理コンピュータに管理者が管理コマンドを使用することで情報の収集・解析を行い，必要に応じ再収集を繰り返すことで実現される．一方，I-AIR による管理方法は管理者が行なうべき作業を定期調査によって代行することで管理者の手をほとんど煩わすことのないネットワーク管理を実現している．また I-AIR 同士の協調・連携によってネットワーク全体の情報を容易に収集し，障害に関する重要情報を I-AIR が自律的に抽出し提供することが可能である．このように I-AIR による自律的なネットワーク監視システムを実現することで管理における一連の作業を代行し管理者の負担を軽減することが可能となる．

## 3. I-AIR の設計と実現

### 3.1 I-AIR の詳細設計

本節では，まず I-AIR の自律的な動作を実現するための内部状態を設計し，I-AIR の内部構造（情報資源，利用支援知識，利用支援機能）の詳細設計を述べる．

**[内部状態設計]** I-AIR には，観測状態と調査状態の 2 つの内部状態が存在する．観測状態とは，自律的に障害を検知するために定期的に遷移する状態であり，この状態によって I-AIR の自律的なネットワークの監視を行うことが可能となる．一方，調査状態とは，障害発生時に検知した I-AIR からの情報を受信してから遷移する状態である．

**[情報資源の設計]** I-AIR の持つ情報資源の形式として RDF/XML 形式 [8] とプレーンテキスト形式の 2 種類を用いる．これらは I-AIR の持つ定期調査によって逐一更新される．

**[利用支援知識の設計]** I-AIR の利用支援知識には情報資源に関する情報や障害を検知する経験的知識，障害判定条件に関する知識，他の I-AIR と協調・連携するための知識が記述され，

- AIR 識別知識 (AIR-Identifier : ID)
- 情報資源に関する知識 (Information Resource : IR)
- 障害検知に関する知識 (Failure Information : FI)
- ネットワーク調査に関する知識 (Control Method : CM)
- 協調プロトコルの知識 (Communication Protocol : CP)

の 5 つの知識から構成される．各知識の表現規則は図 2 に示すとおりである．各知識は協調・連携に必要な最低限の要素を記述するのみで I-AIR として動作することを考慮して設計を行った．これにより，I-AIR の知識や処理の汎用性や柔軟性が向上した ([9])．

**[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・操作する為の機能である．I-AIR の状態情報の監視，他の AIR との協調・連携を実現するためには，以下の機能が必要とされ，利用支援知識によって柔軟かつ効果的に実現される．

- 利用支援知識とのインターフェースとしての機能
- 情報資源を取得・加工する機能
- 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

```
<I-AIR> ::= <ID> <IR> <FI> <CM> <CP>
<ID> ::= <Air id> <Workplace id> <Task id>
        <Air id>          ::= <Value>
        <Workplace id>    ::= <Value> <Air id> | <Name> <Air id>
                            | <Value> | <Name>
        <Task id>         ::= <Value>
        <Name>            ::= <Character>+
        <Value>           ::= <Number>+
        <Character>       ::= '[a – zA - Z]' | '.' | ':'
        <Number>          ::= '[0 - 9]'
<IR> ::= <Info type> <Path> <Format type> <Time>
        <Info type>               ::= <Name>
        <Path>            ::= '/' <Name> | '/' <Name> <Path>
        <Format type>     ::= 'xml' | 'text'
        <Time>            ::= <Year> '/' <Month> '/' <Day> '/'
                              <Hour> ':' <Minute> ':' <Second>
        <Year>            ::= <Value>
        <Day>             ::= <Value>
        <Hour>            ::= <Value>
        <Minute>          ::= <Value>
        <Second>          ::= <Value>
<FI> ::= <Failure name> <Check name> <Check string> <Check info>
        <Failure name>    ::= <Name>
        <Check name>      ::= <Name>
        <Check string>    ::= <Name>+
        <Check info>      ::= <Exist> | <Threshold> <Relation>
                            | <Threshold> <Threshold> <Relation>
        <Exist>           ::= 'yes' | 'no'
        <Threshold>       ::= <Value>+
        <Relation>        ::= 'over' | 'less' | 'between'
<CM> ::= <Method name> <Arguments> <Trigger info>
        <Method name>     ::= <Name>
        <Arguments>       ::= <Argument>+
        <Trigger info>    ::= <Interval> | <Last check>
        <Argument>        ::= <Value> | <Name>
        <Interval>        ::= <Value>
        <Last check>      ::= <Name>+
<CP> ::= <protocol>+
        <protocol>        = 協調プロトコル
```

図 2: I-AIR の知識表現規則

### 3.2 I-AIR の実現

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が，そ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 [10, 11]．これは AIR の持つ，

- 知識に基づいて活動を行う
- 複数の AIR が協調・連携を行い問題を解決する
- 外部からの要求・イベントに応じて活性化される

等の特徴を実現する上で，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，本稿では分散環境上で AIR を実現するために ADIPS/DASH フレームワーク [12, 13] を用いて I-AIR を実現する．ADIPS/DASH フレームワークを用いることで，AIR は，ルール型知識として与えられた利用支援知識に基づき，Java プログラムとして実装された利用支援機能を起動し，情報資源の加工処理や他の AIR との連携・協調処理を実行する．

連携・協調処理を行った I-AIR らは，3.1 節で述べた知識や機能を用いて現在ネットワークに起こっている障

表 1: 実装した I-AIR の一覧

| | 機能目的 | 実行コマンド名 | 対象 |
|---|---|---|---|
| 観測用 I-AIR | ネットワークの不通検知 | ping | 他ホスト |
| | NICの設定ミス検知 | | NIC |
| | 大量メール送信（スパム）検知 | cat | パケットログ（25番ポート） |
| | MSBlaster攻撃の検知 | | パケットログ（135番ポート） |
| | メール送受信エラー検知 | telnet | メールサーバ |
| 調査用 I-AIR | TCP/IPスタックの異常検知 | ping | localhost |
| | NICの設定ミスの検知 | | NIC |
| | ハブの障害検知 | | 同一セグメントホスト |
| | ルータの障害検知 | | 異なるセグメントホスト |
| | 上位ホストとの通信障害検知 | | 上位ホスト |
| | 下位ホストとの通信障害検知 | | 下位ホスト |
| | 各サーバのプロセス稼動検知 | ps | DNSサーバ |
| | | | SMTPサーバ |
| | | | POPサーバ |
| | DNSへの接続検査 | nslookup | DNSサーバ |
| | 所定ホストへの経路調査 | traceroute | サーバ |
| | カーネル情報の調査 | dmsg | localhost |
| | リースIPに関する調査 | cat | dhcpリースログ |
| | メールサーバでのエラー調査 | | メールログ |
| | メール送信数によるスパム調査 | | |

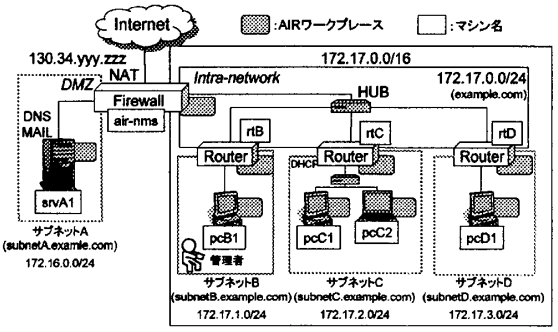


図 3: 実験ネットワーク概要図

表 2: 障害状況と原因

| 障害状況 | 考えられる障害原因 | |
|---|---|---|
| メールの送受信が出来ない | ケーブルの問題 | ①ケーブルの断線 |
| | 接続ポートの問題 | ②25番ポートが閉じている |
| | | ③110番ポートが閉じている |
| | DNSサーバの問題 | ④DNSサーバプロセスが落ちている |
| | | ⑤設定ファイルが有効になっていない |
| | メールサーバの問題 | ⑥メールサーバプロセスが落ちている |

害の特定を行う．その後，特定した障害とその原因についての詳細を管理者にインタフェースを介して提示することで，管理者のネットワーク監視作業の支援を行う．

## 4. 実験と評価

### 4.1 実験方法

本稿では，図3に示すようなNAT機能を実現したファイアウォール，ルータ，各種PCから成り，4つのサブネットから構成されるネットワーク環境上に表1に示すI-AIRを実装し実験を行った．なお，今回の実験において，I-AIRは3.1節で述べた観測状態時に効果を発揮する観測用I-AIRと調査状態時に効果を発揮する調査用I-AIRの2種類に分類され，機能目的や対象に応じて合計20個のI-AIRを実装した．

実験は，ネットワーク管理スキルを持つ5人の被験者に対して，表2の様な，1つの障害に対して考えられる原因が複数ある状況を発生させ，各障害を解消するまでに要した時間と手順を，“OS標準の機能のみを用いた場合”と“I-AIRを用いた場合”とをそれぞれ2回ずつ行い比較することによりI-AIRによる管理の負荷軽減率を示す．なお，どちらの場合も被験者には“メールの送受信が出来ない”という障害が発生したとだけ伝え，どの原因が発生したかは伝えず，原因の発生そのものもランダムとした．また，I-AIRを用いた場合の時間とは，I-AIRらが調査を始めた段階からの時間であり，手順とはI-AIRが提示した作業を管理者が行った際に要した手順である．

### 4.2 実験結果と考察

実験結果を管理者ごとに見た結果（表3）と障害原因ごとに見た結果（表4）に示す．表3に示される実験結果から手順・時間ともにI-AIR未使用時の作業量の約20%近くに軽減されるという結果が得られた．

また，表4の障害原因ごとに見た実験結果から“ケーブルの断線”や“設定ファイルが有効になっていない”などの障害原因に対する管理負荷軽減率は高く，“メールサーバプロセスが落ちている”といった障害原因に対しては他の障害原因ほど負荷軽減がされていないことがわかった．これは管理者の障害原因の究明作業が，例えばプロセスの調査（psコマンド）や名前解決の調査（nslookupコマンド）など一つのコマンドによって究明出来るような作業から行なわれているために，プロセスの問題が原因である場合には早期に発見でき，複数の情報から推測しなくてはならない“見つけにくい”障害原因に対しては多くの時間・手順数を浪費してしまっていることを示している．

本稿のI-AIRによる自律的なネットワーク監視システムは，3.1節にて述べた設計に基づき実現されるため，物理レイヤーからアプリケーションレイヤーまでの幅広い障害原因に対して満遍なく調査を行なうことができ，様々な障害原因に対して柔軟に対応することが可能である．実際に本実験結果から，I-AIRの自律的なネットワーク監視システムは物理的な障害や設定ファイルといったように通常の管理をしていただけでは発見しにくいような障害には特に効果があることが確認された．

## 5. まとめ

本稿では，AIRの概念をネットワーク管理支援システムAIR-NMSの構成要素の1つであるI-AIRを用いた自律的なネットワーク監視システムを提案し，その設計と実現方法について述べた．そして，実験ネットワークにおいて試作したI-AIRを用いて実験を行い，I-AIRがネットワーク管理に必要な一連の作業の多くを代行することにより，ネットワーク管理者の負担を大幅に軽減し，管理者に依存しない柔軟で幅広い管理支援を行なえることを確認した．特に，物理的な障害や設定ファイルといったように，通常のネットワーク管理をしているだけでは発見することが比較的困難な障害原因に関しての

表 3: 管理者ごとに見た実験結果

| | A(管理経験2年) | | | B(管理経験2年) | | | C(管理経験3年) | | | D(管理経験3年) | | | E(管理経験7年) | | | 平均 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 原因 | 時間(秒) | 手順数 | 原因 | 時間(秒) | 手順数 | 原因 | 時間(秒) | 手順数 | 原因 | 時間(秒) | 手順数 | 原因 | 時間(秒) | 手順数 | 時間(秒) | 手順数 |
| I-AIR未使用 | 4 | 158 | 9 | 2 | 566 | 8 | 5 | 929 | 23 | 6 | 235 | 5 | 1 | 655 | 19 | 529.3 | 12.3 |
| | 5 | 743 | 24 | 4 | 871 | 12 | 2 | 339 | 9 | 3 | 615 | 9 | 6 | 182 | 5 | | |
| I-AIR使用 | 1 | 51 | 1 | 6 | 104 | 2 | 3 | 82 | 3 | 1 | 40 | 1 | 2 | 86 | 2 | 80.8 | 2.6 |
| | 6 | 85 | 4 | 3 | 106 | 2 | 4 | 52 | 3 | 5 | 74 | 2 | 5 | 128 | 6 | | |
| I-AIR使用/未使用(%) | | 15.1 | 15.2 | | 14.6 | 20.0 | | 10.6 | 18.8 | | 13.4 | 21.4 | | 25.6 | 33.3 | 15.3 | 21.1 |

表 4: 障害原因ごとに見た実験結果

| | 原因1 | | 原因2 | | 原因3 | | 原因4 | | 原因5 | | 原因6 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 時間(秒) | 手順数 | 時間(秒) | 手順数 | 時間(秒) | 手順数 | 時間(秒) | 手順数 | 時間(秒) | 手順数 | 時間(秒) | 手順数 |
| I-AIR未使用 | 655 | 19 | 566 | 8 | 615 | 9 | 158 | 9 | 743 | 24 | 235 | 5 |
| | — | — | 339 | 9 | — | — | 871 | 12 | 929 | 23 | 182 | 5 |
| I-AIR使用 | 51 | 1 | 86 | 2 | 106 | 2 | 52 | 3 | 74 | 2 | 85 | 4 |
| | 40 | 1 | — | — | 82 | 3 | — | — | 128 | 6 | 104 | 2 |
| I-AIR使用/未使用(%) | 6.9 | 5.3 | 19.0 | 23.5 | 15.3 | 27.8 | 10.1 | 28.6 | 12.1 | 17.0 | 45.3 | 60.0 |

効果が高いことも確認できた.

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，実環境での実験を含めた検討を継続して行ってゆく予定である.

## 参考文献


[1] R.Stephan et al. "Network management platform based on mobile agents," Int. Journal of Network Management, Vol.14, No.1, pp.59–73, 2004.

[2] S.M.Baker et al. "Scalable Web Server Design for Distributed Data Management," Proc. 15th Int. Conf. on Data Engineering, p.98, 1999.

[3] M.P.Consens et al. "Supporting network management through declaratively specified data visualizations," Proc. IEEE/IFIP 3rd Int. Symposium on Integrated Network Management, pp.725–738, 1993.

[4] M.Hasan et al. "A conceptual framework for network management event correlation and filtering systems," Proc. 6th IFIP/IEEE Int. Symposium on Integrated Network Management, pp.233–246, 1999.

[5] A.Virmani et al. "Netmon: Network management for the SARAS softswitch," Proc. IEEE/IFIP Network Operations and Management Symposium, pp.803–816, 2000.

[6] N.Damianou et al. "Tools for domain-based policy management of distributed systems," Proc. IEEE/IFIP Network Operations and Management Symposium, pp.203–218, 2002.

[7] 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.

[8] RDF/XML *Syntax Specification (Revised) W3C Working Draft 23 January 2003.*

[9] 吉村智志 他, "AIR-NMS による状態情報エージェントの設計, " 信学技報, NS2004-90, pp.27–30, 2004.

[10] B.Li et al. "Active information resource: Design concept and example," Proc. 17th Int. Conf. Advanced Information Networking and Applications, pp.274–277, 2003.

[11] 今野将 他, "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, " 情処学論, Vol.46, No.2, pp.493–505, 2005.

[12] 藤田茂 他, "分散処理システムのエージェント指向アーキテクチャ, " 情処学論, Vol.37, No.5, pp.840–852, 1996.

[13] "DASH - Distributed Agent System based on Hybrid architecture," [Online].Available: http://www.agent-town.com/dash