

動的に変化するネットワークシステムのための 知識型障害解決支援システム

谷村 優介^{1,a)} 笹井 一人^{2,b)} 北形 元^{2,c)} 木下 哲男^{2,d)}

概要: 本稿では、運用中に構成変化が発生するような動的なネットワークシステムに焦点をあてるとともに、ネットワーク管理の中でも特に管理者の負担が大きい障害管理に着目し、構成変化に柔軟に対応可能な知識型ネットワーク障害解決支援システムを提案する。診断システムをモジュール化し、構成変化の発生する単位で診断システムを柔軟に拡張・縮小可能とする。そして、ネットワークサービスに関する知識を用いてシステムの構成を自律的に推測することで、構成管理に係る管理者負担を軽減する。動的なネットワークシステムにおける管理者の支援を行う。最後に、試作システムを用いた実験により、本提案の有効性を示す。

1. 背景

近年、インターネットが広く普及するとともに、情報機器の発達によりスマートフォンやタブレット PC などインターネットに接続可能な情報端末がより身近なものとなったことで、コンピュータネットワーク（以下、ネットワーク）を介したサービスは生活に欠かせないものとなっている。また、多くの企業がネットワークシステムを基盤とした活動を行っており、ネットワークシステムに障害が発生しサービスが停止した場合、ユーザの信頼や利益の損失につながるため、可用性の高いネットワークシステムの運用管理が必要とされる。一方で、新しい技術やソフトウェアの導入・応用により、ネットワークの大規模・複雑化やアプリケーションの多様化が加速度的に進んでおり、ネットワークシステムの管理運用には、より高度で幅広い知識とスキルが要求されるようになった。そのため、ネットワークシステムの管理運用に伴う管理者の負担が増大すると同時に、管理者の人材不足が大きな問題となっている。以上のような背景から、ネットワークシステムの管理運用を自動化し、システム管理者の負担を軽減するためのネットワーク管理システム (Network Management System: NMS) へ

の期待と必要性が高まっている。

その一方で、近年の仮想化技術の成熟によって仮想化システムが実用化するとともに、コンピュータの低価格化によってネットワークサービスを提供するための環境を容易に構築することが可能となった背景から、サービス運用中もネットワークシステム内で他のサービスの追加や変更が頻繁に発生する、動的なネットワークシステムへと変遷してきた。しかしながら、従来のネットワーク管理支援システムは、ハードウェア構成やサービス内容が変化しない静的なネットワークを前提として開発されてきたために、動的に変化する監視対象に対応する機能が十分でないという課題が存在する。そこで、本研究では、多種多様なネットワークサービスが運用され、サービスの追加・変更が頻繁に発生するような、日々変化・拡大するネットワークシステム（以下、動的なネットワークシステム）における、システム管理者の障害解決に伴う負担の軽減を目的とする。

2. 知識型のネットワーク管理支援

2.1 関連研究

これまでネットワーク管理運用の自動化に関して様々な研究が行われてきた。ここでは、特に本研究の焦点である、ネットワークシステムの障害解決に関する研究について述べる。ネットワーク障害解決支援システムは、管理者由来の知識に基づいて障害診断・対策案の導出を行う「エキスパート型」とシステム由来の知識に基づいて障害診断・対策案の導出を行う「学習型」に分けることができる。

まず、エキスパート型の障害解決支援システムについて述べる。エキスパート型のシステムの既存研究としては、

¹ 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University

² 東北大学電気通信研究所
Research Institute of Electrical Communication, Tohoku University

a) tanimura@k.riec.tohoku.ac.jp

b) kazuto@riec.tohoku.ac.jp

c) minatsu@riec.tohoku.ac.jp

d) kino@riec.tohoku.ac.jp

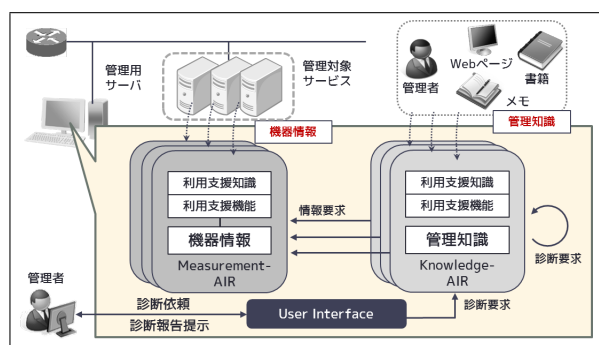


図 1 AIR-NMS による障害解決作業における管理者支援の概要

文献 [1] や [2] が挙げられる。エキスパート型のシステムは、管理者が持つ障害診断方法や対策案の導出に関するノウハウをルールやポリシーとして記述したものを知識とする。そのため、知識を用いて障害診断や対策案を導出する過程は人間の思考過程を模倣したものである。特定のネットワークシステムに依存しない知識記述が可能であるため、知識の再利用が容易であり、幅広いネットワークシステムに適用できるという利点がある。

しかし、エキスパート型システムには知識ベースの管理負担が大きいという欠点がある。知識の重複・欠損・矛盾は推論効率の低下や推論結果の矛盾だけでなく、障害解決支援システムの動作停止の原因となるため、管理者は知識を蓄積する知識ベースを常にメンテナンスし、知識の整合性を維持しつづける必要がある [3][4]。知識を追加・変更する際には、知識ベース内の知識のコンフリクトを避ける必要があるため、既存の知識を参照しながら注意深く知識ベースを更新する必要がある。そのため、知識ベース内の知識が増えるほど知識の追加・変更が困難となり、同時に知識ベースの管理負担が増大する。前述の通り、近年のネットワークシステムは日々変化・拡大する動的なものであることから、エキスパート型システムにおいては知識を頻繁に更新する必要があるため、管理者の負担がより増大すると言える。また、エキスパート型システムの知識にはシステムの構成情報といったシステム依存の情報は含まれないため、管理者がネットワークシステムの構成を調査し、障害解決支援システムに情報を追加する手間がある。そのため、動的なネットワークシステムでは、管理者がシステム構成を調査し情報を追加する手間が頻繁に発生するため、管理者の負担がより増大する。

次に、学習型の障害解決支援システムについて述べる。学習型のシステムの既存研究としては、文献 [3] や [5] が挙げられる。学習型のシステムは、Neural Network や Case-based Reasoning などの手法に基づいて障害発生時のネットワークシステムの状態を学習することで知識を生成する。ネットワークシステムに不具合が発生した場合には、知識を参照することで過去に障害が発生した際のネットワークシステムの状態と比較し、障害原因の特定や対策案の導出

を行う。また、未知の障害症状であっても、既存の知識との類似度によって推論が可能であるという利点がある。

しかし、学習型システムは有効に機能するまでに十分な学習が必要であるため、配備直後には活用できないという欠点がある。学習型システムの知識にはシステム依存の情報が多く含まれるため、システム構成が変化する度に再学習が必要となる。そのため、学習型システムは動的なネットワークシステムにおいて十分な性能を発揮することが難しいと言える。

2.2 能動的情報資源に基づくネットワーク管理システム

能動的情報資源 (Active Information Resource: AIR) は、分散環境上に存在する電子的なデータに対して、そのデータを利用するための知識 (利用支援知識) と機能 (利用支援機能) を付与することで、自律的に動作可能な情報資源とする考え方である [6]。利用支援知識には情報資源の内容や構造、用法に関する知識や他の AIR との連携のための知識などが含まれる。また、利用支援機能には情報資源から情報を取得・加工する機能や他の AIR との連携のための機能などが含まれる。AIR の概念を用いることで、情報資源の利用時に必要な、情報の検索・取得・加工といった作業を AIR が自律的に代行することが可能となり、利用者の負担軽減を図ることができる。

能動的情報資源に基づくネットワーク管理システム (AIR-based Network Management System: AIR-NMS) は、AIR の概念をネットワーク管理に適用したものである [7][8]。ネットワーク管理に関する知識を情報資源とみなし AIR 化したものを K-AIR (Knowledge-AIR)、管理対象機器の機器情報を AIR 化したものを M-AIR (Measurement-AIR) とする。これらの K-AIR と M-AIR が自律的に連携することで、ネットワーク管理作業を代行し、管理者の負担を軽減する。AIR-NMS はネットワークシステムの管理運用の自動化を目標としているが、これまでは主に障害解決支援への適用に注力した研究が行われてきた。

図 1 に AIR-NMS による障害解決作業における管理者支援の概要を示す。AIR-NMS は K-AIR および M-AIR の連携によって障害診断・対策案の導出を行う。推論は K-AIR の連携によって行われ、管理対象機器の情報が必要な場合には、M-AIR 群に問い合わせることで機器情報を取得する。そして、診断終了時には推論結果を診断報告および対策案として管理者に提示する。管理者は提示された対策案を実行することで、障害解決を図ることができる。そのため、AIR-NMS を用いることで、上級管理者の作業負担を軽減することが可能となると同時に、知識やスキルの十分でない初級管理者でも管理業務に携わることが可能となる。

AIR-NMS は障害診断方法や対策案の導出に関する管理者のノウハウを知識とし、管理者から得られた知識に基づいて障害診断を行うため、エキスパート型のシステムであ

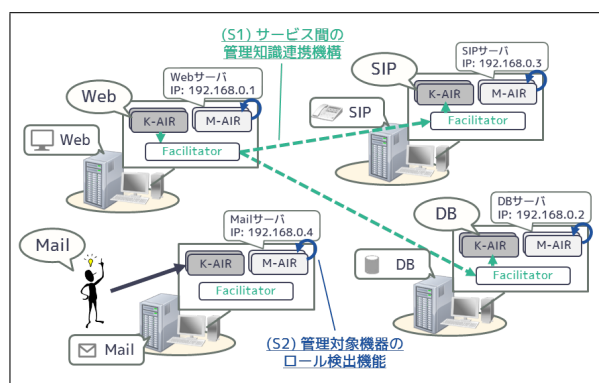


図 2 提案システムの概要

る。よって、前述のエキスパート型ネットワーク障害解決支援システムに関する課題が同じく当てはまる。そのため、動的なネットワークシステムにおける AIR-NMS の適用には、(P1) 知識ベースの管理負担が大きい、(P2) システム構成情報を管理者が与える手間が大きい、という 2 つの課題がある。

そこで本研究では、ネットワークシステムの構成変化を考慮した知識型障害解決支援システムを提案することで、前述の 2 つの課題を解決し、動的なネットワークシステムにおける管理者の障害解決に関する負担の軽減を目指す。

3. 動的なネットワークシステムのための知識型障害解決支援システムの提案

3.1 問題解決のアプローチ

初めに、図 2 に提案システムの概要を示す。本研究では前述の 2 つの課題について、以下のようなアプローチで解決を図る。

(P1) 知識ベースの管理負担が大きい

既存の障害解決支援システムでは、単一の知識ベースで全ての知識を管理していたために、システム構成が発生した際の知識の追加・更新が難しいものとなっていた。そこで、診断システムをモジュール化し、システム構成の変化する単位（サービス）で障害解決支援システムを拡張・縮小することができれば、知識の追加・更新が影響する範囲をサービス/モジュール内に限定することができる。サービス毎にモジュール化された診断システムによって構成された障害解決支援システムでは、サービス単位で知識を追加・更新することが可能となるため、知識ベースの管理負担を軽減することができる。しかし、実際のネットワークシステムにおいてはサービス間に依存関係があり、一つのサービス障害に起因して連鎖的に複数のサービスに障害が波及することがあるため、サービス毎にモジュール化した知識間で、連携して診断を実行できる必要がある。そこで本提案では、AIR-NMS に新しく (S1) サービス間の管理知識連携機構を導入する。

(P2) システム構成情報を管理者が与える手間が大きい

既存の障害解決支援システムでは、管理者が手作業で

ネットワークシステムの構成を調査して情報を入力する必要があった。管理者がネットワークシステムの構成を調査する際には、まずシステムを構成する機器がシステム内でどのような役割（ロール）を持つかを調べる必要がある (e.g. Web サーバ, DB サーバ, Storage サーバ, Firewall)。多種多様な機器に対して共通の方法でロールを検出することができれば、システム構成情報の取得を自動化することが可能であるが、そのような汎用プロトコルは存在しない。しかし、人間の管理者は経験的知識に基づき、複数のツールから適切なツールを選択・組み合わせるなどして、多種多様な機器からロールを取得することが可能である。そこで本提案では、人間の管理者が持つ経験的知識を AIR に付与し、人間の管理者と同様の方法でロールを取得可能とするための (S2) 管理対象機器のロール検出機能を導入する。

3.2 (S1) サービス間の管理知識連携機構

サービス間の管理知識連携機構は、各サービスに分散する知識を必要に応じて活用し、依存関係のあるサービス間での障害診断を可能とするための仕組みである。サービス間で管理知識を連携させるため、AIR-NMS に新たに Facilitator-Agent を導入する。また、Facilitator-Agent はサービス毎に配置する。発生した障害症状が、依存関係のある他のサービスの障害に起因する可能性がある場合、各サービスの Facilitator-Agent が連携することによって、対応可能な K-AIR をもつサービスが診断を引き継ぐ。

このような仕組みの実現のため、Facilitator-Agent に以下の 3 つの機能を持たせる。

(F1) 診断要求の転送機能

通信可能な他サービスの Facilitator-Ag を検索・認識し、自サービスの K-AIR から受け取った診断要求を他サービスの Facilitator-Ag に送信する

(F2) サービス内の管理知識抽出機能

自サービス内の管理知識に関するメタデータを抽出し、診断可能な障害をリストアップする

(F3) 診断要求の対応可能性判定機能

診断可能な障害のリストに基づき、他サービスからの診断要求に対応可能かを判定する

3.3 (S2) 管理対象機器のロール検出機能

管理対象機器のロール検出機能は、人間の管理者が持つロール取得に関する経験的知識を M-AIR に付与することで、M-AIR が管理者に代わって機器のロールを自律的に取得し、システム構成情報を推測可能とするための機能である。管理者によるシステム構成情報の入力が無い場合でも、本機能によってシステム構成情報を推測することで障害診断を可能とする。そして、システム構成に変化が発生した際には自律的にシステム構成情報を取得・更新することで、システム構成の変化に頑健な障害診断を可能とする。

```
<sc symptom="SIPクライアント発信不可">
<info>SIP-client_ID</info>
<cause>SIPサーバのプロセスダウン</cause>
<cause>SIP接続ポート未開放</cause>
<cause>発信が許可されていないID</cause>
</sc>
```

図 3 障害症状から想定される障害原因の候補に関する知識 (Ksc)

```
<cd cause="SIP接続ポート未開放">
<dm>
<p>get #SIP-port_num# cmd #grep ^port /etc/asterisk/sip.conf
|awk -F "=" '{print $2}' # login #SIP-server_IP#</p>
<p>get #SIP-port_state# cmd #grep val(SIP-port_num) /etc/sysconfig/iptables
|awk -F " " '{print $NF}' # login #SIP-server_IP#</p>
<p>true (#SIP-port_state# -ne ACCEPT)</p>
</dm>
<dr> SIPサーバ #SIP-server_IP# の #SIP-port_num# 番ポートが開放されていない </dr>
</cd>
```

図 4 想定された障害原因に対する診断手法および診断報告に関する知識 (Kcd)

```
<cm cause="SIP接続ポート未開放">
<m>SIP用のUDPポート #^port="value"$# 番を開放してください。 \rtn
1. #SIP-server_IP# に root でログイン。 \rtn
2. 「/etc/sysconfig/iptables」に 「-A INPUT -m state --state NEW -m udp -p udp
--dport #^port="value"$# -j ACCEPT」を追加して下さい。 \rtn
3. 「/etc/init.d/iptables restart」でiptablesを再起動して下さい。 </m>
</cm>
```

図 5 特定された障害原因を解決するための対策案に関する知識 (Kcm)

管理対象機器のロールを検出するため、管理対象機器の機器情報を扱う M-AIR に本機能を付与する。M-AIR にはロール取得に関する知識として、管理対象機器への接続方法に関する知識とロール探索方法に関する知識を付与する。

まず、管理対象機器への接続方法に関する知識として、管理者が機器へ接続する際に用いるツール (e.g. telnet, ssh) を使用するための知識を付与する。M-AIR はこの知識に基づいて管理対象機器にリモートログインを行い、コマンドを投入することでロールを取得する。また、機器に投入するコマンドはロール探索方法に関する知識に基づいて導出される。ロール探索方法に関する知識は管理者のロール探索行動を知識化したものであり、ロール探索に使用するツールの選択およびツールの使用方法に関する知識が含まれる。M-AIR は本知識に基づき、人間の管理者と同様の方法でロールの探索を行う。

M-AIR は本機能によって取得した機器のロール名を、ロール取得元の機器の名前とあわせて、(S1) で導入した Facilitator-Agent に送信する。Facilitator-Agent は M-AIR から機器のロール名と機器名を受け取った際に、これらに対応付けてリスト化し、システム構成情報として保持する。障害診断時に機器の名前が不明である場合には、Facilitator-Agent に機器のロール名で問い合わせることで、システム構成情報のリストから機器名を得る。

4. 試作システムの設計と実装

4.1 設計

提案に基づき、ネットワークシステムの構成変化を考慮した知識型ネットワーク障害解決支援システムを試作した。試作システムは先行研究での設計・実装に基づき、

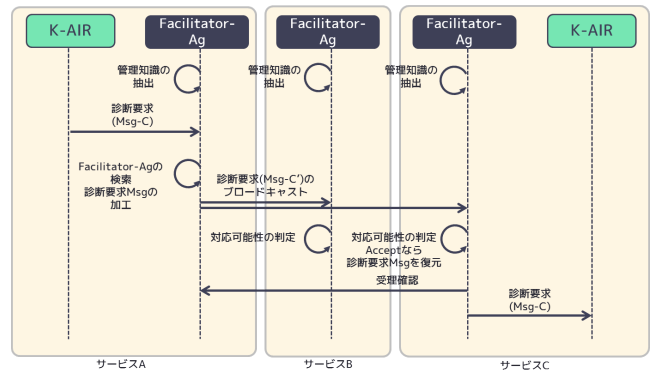


図 6 サービス間の管理知識連携機構に関するエージェントおよび AIR 間メッセージの流れ

- ・ (SIPクライアント発信不可, Ksc-AIR.201412111553015:w2:AIR-NMS-PC-1)
- ・ (SIPサーバのプロセスダウン, Kcd-AIR.201412111553020:w3:AIR-NMS-PC-1)
- ・ (SIP接続ポート未開放, Kcd-AIR.201412111553023:w3:AIR-NMS-PC-1)

図 7 対応可能な障害原因名と K-AIR のインスタンス名のマッピング例

DASH/IDEA[9] を AIR-NMS プラットフォームとして使用することを考慮して設計を行った。

4.1.1 K-AIR の設計

先行研究での設計・実装に基づき、提案システムにおいても、以下の通り 3 種類の K-AIR を設計した。

- Ksc-AIR: 障害症状から想定される障害原因の候補に関する知識 (Ksc) を扱う
- Kcd-AIR: 想定された障害原因に対する診断手法および診断報告に関する知識 (Kcd) を扱う
- Kcm-AIR: 特定された障害原因を解決するための対策案に関する知識 (Kcm) を扱う

また、図 3, 4, 5 にそれぞれ Ksc, Kcd, Kcm の例を示す。管理知識は XPath の文法に基づき、XML 形式で記述する。

4.1.2 Facilitator-Agent の設計

図 6 に「(S1) サービス間の管理知識連携機構」に関するエージェントおよび AIR 間メッセージの流れを示し、以下に Facilitator-Agent の持つ各機能の設計について述べる。

(F1) 診断要求の転送機能

他サービスの Facilitator-Agent の検索には、AIR-NMS プラットフォームとして使用する DASH/IDEA のネームサーバ機能を使用する。定期的には再検索することで、サービスの変化に対応する。

(F2) サービス内の管理知識抽出機能

Facilitator-Agent はサービス内の K-AIR に問い合わせることで、K-AIR が対応可能な障害原因の情報を得る。そして、図 7 のように、K-AIR が対応可能な障害原因名と K-AIR のインスタンス名をマッピングする。

(F3) 診断要求の対応可能性判定機能

他サービスからの診断要求受信時、(F2) で作成したリストを参照することで対応可能性を判定する。対応可能で


```

• Linux
$ ls /etc/rc3.d |grep ^S |sed -e 's/^S.*/'
• FreeBSD
$ cat /etc/rc.conf |grep _enable=\\|\"YES\\|\"$ |grep ?v \\\"^#\\|
sed ?e 's/_enable=\\\"YES\\\"$/'
• SunOS
$ svcs -a |awk '$1==\"online\" {print $3}' |awk -F: '{print $2}' |
awk -F/ '{print $NF}'
    
```

図 8 運用サービス取得コマンドの例

あれば、受理確認を返信するとともに、当該原因に対応することのできる K-AIR のインスタンス名を取得し、当該 K-AIR に診断要求メッセージを送信する。

4.1.3 M-AIR の設計

試作システムではサーバを管理対象機器と想定して M-AIR の設計を行った。

(S2) 管理対象機器のロール検出機能

M-AIR にはロール取得に関する知識として、管理対象サーバへの接続方法に関する知識とロール取得に関する知識を付与した。試作システムでは、管理対象サーバへの接続には Secure Shell (ssh) を使用する。そこで、M-AIR に ssh 接続のための機能を付与するとともに、ssh を使用してサーバへ接続するための知識を付与した。また、試作システムでは、サーバ起動時に自動で実行されるサーバプロセスを調査することで、ロールを検出する。そこで、サーバ起動時に起動されるサーバプロセスを調査するための知識を付与した。サーバプロセスの自動起動設定方法は複数存在するため、複数の自動起動設定取得コマンドを用意し、状況（サーバ OS の種類）に合わせて使い分けるような知識を付与した。図 8 にサーバプロセスの自動起動設定取得コマンドの例を示す。M-AIR はサービス管理コマンドや OS の起動スクリプトを参照するコマンドを投入することで、OS 起動時に実行されるサーバプロセスの一覧を取得する。この際取得されるプロセス名をもとにサーバのロールを判定する (e.g. httpd → Web サーバ)。

4.2 実装

設計にしたがって試作システムの実装を行った。実装および動作環境として、DASH/IDEA[9]を使用した。

5. 実験と評価

5.1 実験概要

5.1.1 目的

本実験の目的は、提案システムによって前述の課題が解決可能であることを確認することである。

5.1.2 実験内容

サービス追加を想定し、前章で設計・実装した試作システム（以下、提案システム）と既存研究 [7] に基づく試作システム（以下、既存システム）に対してそれぞれ知識を追加し障害診断を行った。そして、提案システムと既存システムの診断結果および診断中の AIR/Ag 間メッセージ数を

表 1 各サービスの管理知識の数

| | 管理知識 (種類) | | |
|--------------|-----------|-----|-----|
| | Ksc | Kcd | Kcm |
| Web サービス | 2 | 11 | 11 |
| DB サービス | 1 | 5 | 5 |
| Storage サービス | 1 | 4 | 4 |
| SIP サービス | 1 | 6 | 6 |
| Mail サービス | 3 | 14 | 14 |
| 合計 | 8 | 39 | 39 |

比較することで、新しく導入した機構と機能の有効性の検証を行った。実験は既存システムを用いた障害診断 (ケース 1) と提案システムを用いた障害診断 (ケース 2) に分けて行った。これらの実験の流れを以下に示す。

- (1) 管理者は管理対象ネットワークシステムで Web サービス、DB サービス、Storage サービス、SIP サービスを運用中である
- (2) 運用中のネットワークシステムに新たに Mail サービスを追加し、管理対象に加える
- (3) 管理者は Mail サービス障害診断用の知識を作成し、障害解決支援システム (AIR-NMS) に知識を追加する
- (4) 運用中のネットワークシステム内で障害が発生する (運用中の Web ページの閲覧ができなくなる)
- (5) Web サービス利用者からの連絡を受けて、管理者が AIR-NMS を用いて Web サービスの障害診断を行う
- (6) AIR-NMS により生成された対策案を実行し、Web サービスが復旧するかを確認する

また、Mail サービスの障害診断用の知識を作成する際には、運用中の他サービスの知識を参照せず、新しく追加する Mail サービスのみを考慮して知識を作成する。そのため、新しく追加する知識は Mail サービスのみの環境では正しく動作するが、運用中の他サービスの知識に追加した際には管理者が意図した動作をするかが不明である。本実験ではこのような状況を想定し、運用中のサービスに関する知識とコンフリクトが発生するような Mail サービスの障害診断知識を意図的に作成し追加した。

実験に使用した各サービスの管理知識の数を表 1 に示す。本実験では Kcd と Kcm を 1 対 1 対応として作成した。また、前述の通り、既存の知識とのコンフリクトが発生するように Mail サービス用の知識を作成し追加した。具体的には、Mail サービス内の DB サーバに関する Kcd と Kcm (DB プロセスダウン、DB 接続ユーザ・パスワード間違い、接続先のデータベースが存在しない) を、既存の DB サービスの Kcd と Kcm と重複させた。

5.2 ケース 1:既存システムによる障害診断

5.2.1 実験手順

ケース 1 では既存システムを使用して障害診断を行う。ケース 1 における実験を図 9 に示す。ケース 1 では管理知

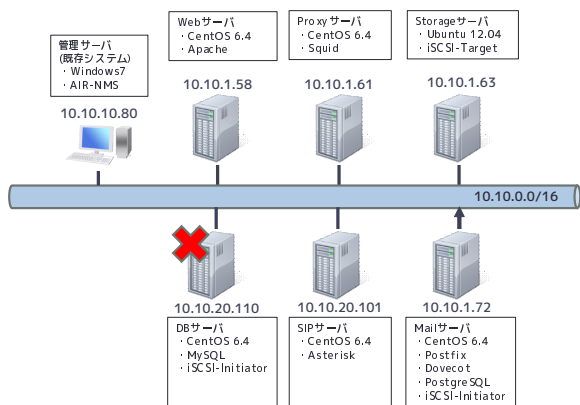


図 9 ネットワークシステムの構成 (ケース 1)



図 10 AIR-NMS インタフェースへの入力内容 (ケース 1)

識を全て 1 つの管理サーバ (10.10.10.80) に配置する。

障害を発生させるため、DB サーバ (10.10.20.110) の MySQL プロセスを停止する。これにより、Web サーバが DB から Web ページのコンテンツを取得できなくなるため、障害症状「Web ページ閲覧不可」が発生する。そこで、管理者が Web サービス利用者からの連絡を受けたことを想定し、AIR-NMS を使用して障害診断を行う。

図 10 に管理サーバの AIR-NMS インタフェースに入力した内容を示す。障害症状として「Web ページ閲覧不可」を選択し、Web サーバの IP アドレス、アクセス元 Web クライアントの IP アドレス、閲覧ができなかった Web ページ URL、DB サーバのアドレス、ストレージサーバのアドレスを入力した。そして診断ボタンを押下することで診断を実行した。診断結果を次項にて述べる。

5.2.2 実験結果

図 11, 12 に AIR-NMS インタフェースから出力された診断報告と対策案を示す。図 11 より、障害原因として「DB プロセスダウン」に関する 2 種類の診断報告が出力されたことが分かる。しかし、実験開始前に意図的に発生させた障害は MySQL のプロセスダウンに起因する障害であるため、診断報告「10.10.20.110 で MySQL のプロセスがダウンしています」は意図的に発生させた障害原因と一致するが、診断報告「10.10.20.110 で PostgreSQL のプロセスがダウンしています」は誤りである。図 12 より、障害原因「DB プロセスダウン」に関する対策案が複数提示されたことが

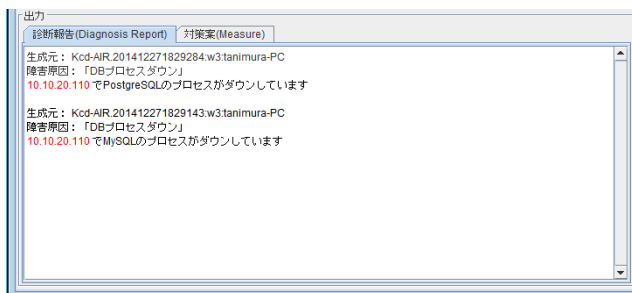


図 11 出力された診断報告 (ケース 1)

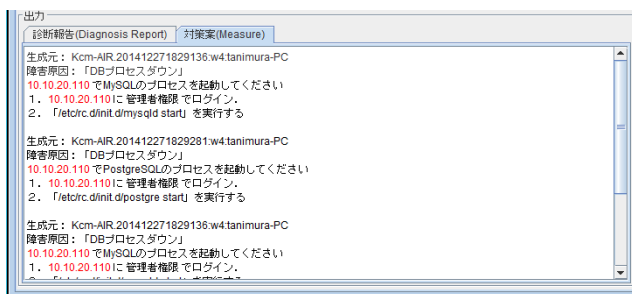


図 12 出力された対策案 (ケース 1)

分かる。具体的には、「10.10.20.110 での MySQL のプロセスの再起動の方法」と「10.10.20.110 での PostgreSQL のプロセスの再起動の方法」がそれぞれ 2 回ずつ提示された。しかし、10.10.20.110 のサーバにおいて PostgreSQL は運用していないため、対策案「10.10.20.110 での PostgreSQL のプロセスの再起動の方法」は誤りである。

Kcd-AIR 群の動作ログより、「DB プロセスダウン」など同名の障害原因を扱う複数の AIR が診断を開始したことが分かった。これは、Mail サービス用に作成・追加した Kcd が、DB サービスの診断中に発火したことを示す。MySQL を運用するサーバに対して PostgreSQL に関する知識を使用して診断が行われた結果、誤った障害原因を特定し、誤った対策案を生成した。

以上より、既存システムでは知識のコンフリクトによって、誤った障害原因を特定するとともに誤った対策案を提示した。また、同じ対策案を複数生成してしまうなど、知識のコンフリクトは推論効率にも影響を及ぼしたことが分かる。診断中の AIR/エージェント間メッセージ数はケース 2 の結果と併せて示す。

5.3 ケース 2:提案システムによる障害診断

5.3.1 実験手順

ケース 2 では提案システムを使用して障害診断を行う。ケース 2 における管理サーバを含めたシステム構成を図 13 に示す。ケース 2 では管理サーバをサービス毎に配置し、サービス毎に管理知識を管理する。

障害を発生させるため、ケース 1 と同様に DB サーバ (10.10.20.110) の MySQL プロセスを停止する。これにより、Web サーバが DB から Web ページのコンテンツを取

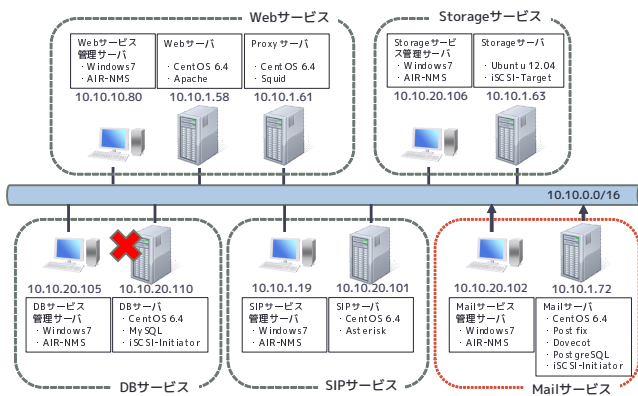


図 13 ネットワークシステムの構成 (ケース 2)



図 14 AIR-NMS インタフェースへの入力内容 (ケース 2)

得できなくなるため、障害症状「Web ページ閲覧不可」が発生する。そこで、管理者が Web サービス利用者からの連絡を受けたことを想定し、Web サービス管理サーバの AIR-NMS を使用して障害診断を行う。

図 14 に Web サービス管理サーバの AIR-NMS インタフェースに入力した内容を示す。障害症状として「Web ページ閲覧不可」を選択し、アクセス元 Web クライアントの IP アドレス、閲覧ができなかった Web ページ URL を入力した。このように、ケース 2 では Web サービス利用者から得られる情報のみを入力し、サーバの IP アドレスなどのシステム構成情報は入力しない。その後、診断ボタンを押下することで診断を実行した。

5.4 実験結果

図 15, 16 に Web サービス管理サーバの AIR-NMS インタフェースから出力された診断報告と対策案を示す。

図 15 より、障害原因として「DB プロセスダウン」が特定され、診断報告が出力されたことが分かる。これは、実験開始前に意図的に発生させた障害原因と一致する。ケース 1 と異なり、診断報告は 1 つのみ表示された。

図 16 より、特定された障害原因「DB プロセスダウン」に関する対策案が生成されたことが分かる。提示された対策案を実際に実行することで障害症状「Web ページ閲覧不可」は解消し、Web サービスが復旧した。また、ケース 1 と異なり対策案は 1 回のみ提示された。

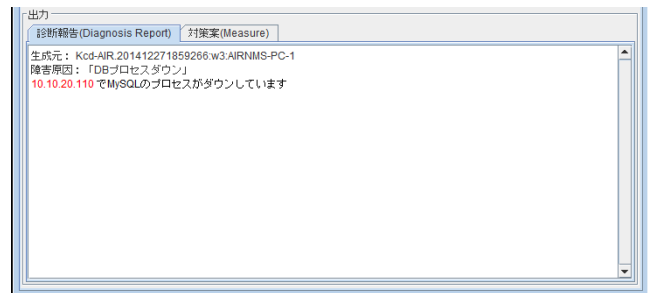


図 15 出力された診断報告 (ケース 2, Web サービス管理サーバ)

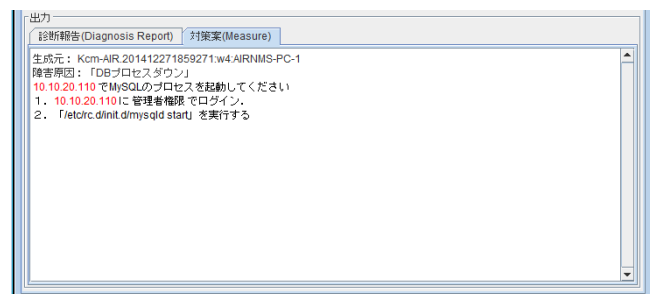


図 16 出力された対策案 (ケース 2, Web サービス管理サーバ)

Web サービス管理サーバの AIR-NMS 動作ログおよび DB サービス管理サーバの AIR-NMS 動作ログより、Web サービス管理サーバにて開始された障害診断が DB サービス管理サーバに引き継がれたことが分かった。具体的には、Web サービス管理サーバの Facilitator-Agent からの診断要求「DB サーバ接続不可」を DB サービス管理サーバの Facilitator-Agent が受理した。そして DB サービス管理サーバにて「DB サーバ接続不可」に関する詳細な障害原因の診断が行われ、障害原因「DB プロセスダウン」を特定するとともに対策案を生成した。

また、Mail サービス管理サーバの AIR-NMS 動作ログより、Mail サービス管理サーバでは各 AIR に動作が見られなかった。これは、Web サービス管理サーバからの診断要求「DB サーバ接続不可」が Facilitator-Agent によって却下されたからである。

図 17 にケース 1 およびケース 2 における、診断時の AIR/エージェント間メッセージ数を示す。この結果より、提案システムでは診断中のメッセージ数が大幅に減少していることが分かる。これは、知識をサービス毎に構造化することで、診断要求メッセージや情報要求メッセージのブロードキャスト範囲がサービス内に限定されたことや、知識のコンフリクトによる冗長な診断・対策案の生成が行われなかったからである。

以上の結果より、(S1) サービス間の管理知識連携機構が有効に機能し、知識を構造化することで知識のコンフリクトを解消するとともに、サービス毎に分散する知識を効率的に活用した障害診断を行えることを確認した。そして、提案システムでは管理者によるシステム構成情報の入力が無くても各機器の障害診断を行えたことから、(S2) 管理

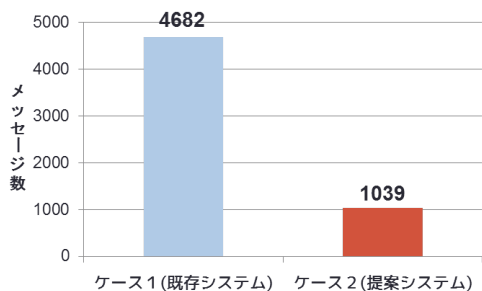


図 17 診断中の AIR/Ag 間メッセージ総数

対象機器のルール検出機能が有効に機能し、自らシステムの構成情報を推測して障害診断を行えることを確認した。

5.5 考察と評価

(S2) 管理対象機器のルール検出機能により、提案システムでは、システム構成に変化が発生しても管理者によるシステム構成情報の追加が不要となったため、構成管理の負担が軽減されている。今回は実験に用意したネットワークシステム向けに機能の実装を行ったが、検出したいルール、つまりサーバプロセス名と対応するサービス名を知識として M-AIR に追加するのみで取得可能なサービスを増やすことができるため、多様な環境に対応して拡張することが可能である。

サービス毎に分散する知識を必要に応じて活用して障害原因を特定し対策案を生成したことから、(S1) サービス間の管理知識連携機構が有効に機能したことを確認した。提案システムではサービス毎に知識をモジュール化することで、知識の追加・更新にともなう影響がサービス内に限定されるため、他サービスとの知識のコンフリクトを考慮せずに知識の追加が可能となっている。よって、提案システムでは、頻繁にシステム構成が変化するような状況に置ける知識ベースの管理負担が軽減されている。

また、今回は Web サービスと DB サービスの管理知識の連携に関する実験を行ったが、試作システムではこれに加えて DB サービスと Storage サービス、Mail サービスと Storage サービスの管理知識の連携が可能である。サービスを越えて管理知識の連携が必要な場合は、依存関係のあるサービスの障害症状を Ksc に書き加えるのみで可能であり、サービスの種類に依存した特殊な知識記述は不要であるため、他のサービスに対しても容易に適用可能である。

6. 結論

本稿では、日々変化・拡大するネットワークシステムにおけるシステム管理者の障害解決に関する負担軽減を目的とし、AIR の概念に基づくネットワーク管理システムを拡張することで、既存のネットワーク障害解決支援システムの課題の解決を図った。AIR-NMS に (S1) サービス間の管理知識連携機構を導入し、診断システムをサービス毎にモジュール化することで知識ベースの管理負担を軽減する

とともに、(S2) 管理対象機器のルール検出機能を導入することで、システム構成情報の入力に関する管理者負担の軽減を図った。提案システムの有効性を検証するため、試作システムを設計・実装し、評価実験を行った。実験の結果から、(S1) (S2) が有効に機能し、提案システムではシステム構成に変化が発生してもシステム構成情報の入力が不要であることや、容易に知識の追加・変更が可能であることを示した。そして、提案システムによって前述の課題を解決し、日々変化・拡大するネットワークシステムにおけるシステム管理者の障害解決に関する負担を軽減可能である事を示した。

謝辞

本研究の一部は、総合科学技術・イノベーション会議の SIP (戦略的イノベーション創造プログラム)「レジリエントな防災・減災機能の強化」(管理法人: JST) によって実施されました。

参考文献

- [1] Ross, M., Covo, A. and Hart Jr, C.: An AI-based network management system, *Computers and Communications, 1988. Conference Proceedings., Seventh Annual International Phoenix Conference on*, IEEE, pp. 458-461 (1988).
- [2] Cronk, R. N., Callahan, P. H. and Bernstein, L.: Rule-based expert systems for network management and operations: an introduction, *Network, IEEE*, Vol. 2, No. 5, pp. 7-21 (1988).
- [3] Keeney, J., van der Meer, S. and Hogan, G.: A recommender-system for telecommunications network management actions, *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, IEEE, pp. 760-763 (2013).
- [4] Lupu, E. C. and Sloman, M.: Conflicts in policy-based distributed systems management, *Software Engineering, IEEE Transactions on*, Vol. 25, No. 6, pp. 852-869 (1999).
- [5] Montani, S. and Anglano, C.: Case-based reasoning for autonomous service failure diagnosis and remediation in software systems, *Advances in Case-Based Reasoning*, Springer, pp. 489-503 (2006).
- [6] 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, 電子情報通信学会技術研究報告. AI, 人工知能と知識処理, Vol. 99, No. 446, pp. 13-19 (1999).
- [7] Sasai, K., Sveholm, J., Kitagata, G. and Kinoshita, T.: A Practical Design and Implementation of Active Information Resource based Network Management System, *International Journal of Energy, Information and Communications*, Vol. 2, No. 4, pp. 67-86 (2011).
- [8] Tanimura, Y., Sveholm, J., Sasai, K., Kitagata, G. and Kinoshita, T.: A knowledge-based support method for autonomous service operations after disasters, *Computer and Information Science (ICIS), 2013 IEEE/ACIS 12th International Conference on*, IEEE, pp. 229-233 (2013).
- [9] 打矢隆弘, 前村貴秀, 菅原研次, 木下哲男: エージェントシステムのインタラクティブ開発環境 (プラットフォーム/ツール, ソフトウェアエージェントとその応用論文), 電子情報通信学会論文誌. D-I, 情報・システム, I-情報処理, Vol. 88, No. 9, pp. 1344-1355 (2005).