［論文］

ネットワークサービスの可視化を主眼に置いたシステム運用者支援方法の提案

川崎敏行†‡

要 旨

近年，インターネットの普及によって，多くの企業および学校，官公庁でインターネットを利用したミッションクリティカルなシステムが増加してきている。そのため，今まで以上にシステムの高可用性が必要とされるようになってきた。しかし，一方でシステムの大規模化，複雑化が進み，限られたシステム運用者リソースで全てのイベントに対応することが困難になってきている。

本研究では，この問題に対応するため，ネットワークサービスの可視化という概念を取り入れ，システム運用者の監視業務を支援するシステム改善方法を提案する。これによって，限られたシステム運用者リソースで大規模複雑化したシステムを管理することが可能となり，システムの可用性を重視した運用方針の戦略を立てることが可能となる。

Abstract

The recent diffusion of Internet is leading to more Internet-based mission-critical systems at companies, schools, and governmental agencies with the result that high system availability is more important than ever. As systems grow in scale and become more complicated, however, it becomes difficult to cope with all events that arise using the limited system administrator resources.

In this study, it is proposed how to support the operation of system administrators incorporating the concept of visible Network services to deal with this issue. This should enable the system administrator to manage a large and complicated system with limited system administrator resources, allowing them to plan an operation strategy with an emphasis on system availability.

1 はじめに

近年，インターネットの普及によって，多くの企業および学校，官公庁でインターネットを利用したミッションクリティカルなシステムが増加してきている。そのため，今まで以上にシステムの高可用性が必要とされるようになってきた。なぜなら，短時間のシステム停止であっても，ビジネスや業務，社会サービスに大きな影響を及ぼす可能性が高いからである。システム停止による影響を避けるため，現在ではシステムの冗長性確保，負荷分散を目的とした分散処理が一般的になってきており，監視すべきサーバおよびネットワーク機器の台数が増加傾向にある。そのため，以前に比べてシステムの大規模複雑化が進み，システムを保有する組織では独自のリソースでコストに見合ったシステム運用の実施が困難な状況になってきている[1]。

この状況を改善するため，最近ではシステム運用業務の一部を外部組織に委託するシステム運用アウトソーシング（以下，アウトソーシング）を採用する動きが活発である。しかし，専門的知識を有するアウトソーシング組織であっ

The proposal how to support system administrators aiming principally at visible Network services

Toshiyuki Kawasaki†‡
† Nomura Research Institute, Ltd.
‡ Shinshu University

［論文］2008 年 6 月 2 日受付

表 1　プロアクティブ監視項目とリアクティブ監視項目の違い

種類	サービス影響	検知内容
プロアクティブ監視項目	要調査	監視対象機器への Ping 疎通が 20%ロスした
リアクティブ監視項目	あり	監視対象機器への Ping 疎通が Timeout した

ても，限られたシステム運用者（以下，運用者）リソースの中でシステムの高可用性を維持していくことは，決して容易なことではない。

現在，システムの高可用性を維持するために用いられている代表的な方法としては，次の 2 つが挙げられる。

- サーバ・ネットワーク機器等の冗長構成をとる
- サーバ・ネットワーク機器等のシステム監視をする

サーバ・ネットワーク機器（以下，機器）の冗長構成は，稼動系機器のハードウェア障害等が発生したとしても，待機系で引続きサービスを継続できるというメリットがある。一方，機器のシステム監視は，ハードウェアやアプリケーションの異常を早期に発見し，障害対応に対するトリガを運用者へ通知することができるというメリットがある[2] [3] [4] [5]。

しかしながら，システム監視は，機器の増加に伴い，システム監視項目（以下，監視項目）が増大傾向にあり，運用者が全ての監視項目を厳密に管理できないという状況になる可能性が高い。また，監視項目の増大によって，監視システムで検知される障害イベント（以下，イベント）の増加も予想され，限られた運用者リソースでの対応が困難となる可能性が高い。

本研究報告では，アウトソーシングの依頼を受ける組織または大規模なシステムを複数所有する組織が，システムの高可用性を意識したシステム運用を行うために，監視システムをどのように改善していけば良いかに関して提案をしていく。

2 従来のシステム監視手法と問題点

2.1 従来のシステム監視手法

2.1.1　システム監視ツール

システム監視を行うための代表的なツールとしては，次の 3 つが挙げられる。

- 監視ソフトウェアベンダが販売している商用監視ソフトウェアを用いる
- フリーの監視ソフトウェアを用いる
- Simple Network Management Protocol 等の管理プロトコルを用いる

商用監視ソフトウェアの代表的なものとしては，日本ヒューレット・パッカード社の「OpenView」[6]，野村総合研究所社の「千手」[7], BMC ソフトウェア社の「BMC Performance Manager」[8]，コンピュータ・アソシエイツ社の「Unicenter」[9]，日立製作所社の「JP1」[10]等がある。フリーの監視ソフトウェアの代表的なものとしては「nagios」[11]，NTT データ社の「Hinemos」[12]等がある。

システム監視ツールは，対象となるシステム構成や規模，監視要件によって最適なものを導入するのが一般的である。

2.1.2　監視項目の決定

監視項目は，対象となるシステム要件や提供するネットワークサービス（以下，サービス）要件に合わせて必要分定義する必要がある。

定義を行う監視項目は，サービス停止の予防を目的としたプロアクティブ[13] [14]なもの（以下，プロアクティブ監視項目），サービス停止の復旧を目的としたリアクティブ[13] [14]なもの（以下，リアクティブ監視項目）がある。表 1 は，監視対象機器への Ping 疎通を例にプロアクティブ監視項目とリアクティブ監視項目の違いを示したものである。

プロアクティブ監視項目のイベントを検知した際は，発生原因の特定，事象悪化の回避を目的に対応を進めることになる。一方，リアクティブ監視項目のイベントを検知した際は，障害の復旧を最優先に対応を進めることになる。

さらに，プロアクティブ監視項目とリアクティブ監視項目は，正常または異常の判断が明確にできるもの（以下，ステータス監視項目），各種ログファイルの文字列監視を行うもの（以

表 2　ステータス監視項目とログ監視項目の違い

種類	正常・異常の判断	検知内容
ステータス監視項目	明確	アプリケーションのプロセスが存在しなくなった
ログ監視項目	運用者による判断が必要	アプリケーションのログに「エラー」の文字列が出力された

表 3　監視項目の分類

大分類	小分類
A.　プロアクティブ監視項目	A-1.　ステータス監視項目
	A-2.　ログ監視項目
B.　リアクティブ監視項目	B-1.　ステータス監視項目
	B-2.　ログ監視項目

表 4　監視項目の例

	監視項目
1	1 台の物理サーバに対する Ping 死活監視
2	WEB サーバアプリケーションプロセスの個数に対する閾値監視
3	WEB サーバアプリケーション使用ポートに対する PORT ALIVE 監視
4	WEB コンテンツに対する GET リクエスト監視

下，ログ監視項目）に分類することができる。表 2 は，アプリケーションの監視を例にステータス監視項目とログ監視項目の違いを示したものである。

ステータス監視項目にて，アプリケーションプロセスの非存在を検知した際，アプリケーションが正常に動作していない状態であると判断することができる。一方，ログ監視項目にてアプリケーションのエラー文字列を検知した際は,運用者がメッセージの意味,アプリケーションの稼動状態等を調査し，最終判断をする必要がある。

表 3 は，各監視項目の分類をまとめたものである。大分類では，イベント発生時の対応方法の違いを明確にしており，小分類ではイベント発生時の判断方法の違いを明確にしている。実際に各種提供サービスを監視するには，さまざまな監視項目をバランス良く組み合わせることが必要となる。

1 つのサービスに対して，同じ分類に属する複数の監視項目を定義する場合もある。例えば，1 台の物理サーバ上で提供される WEB サービスを監視する場合，表 4 の監視項目が考えられる。表 4 の監視項目は全て表 3 の B-1 に属する監視項目である。

1 つのサービスに対して，同じ分類に属する複数の監視項目が考えられる背景には，イベント発生時の原因切り分けが行いやすく，さまざまな角度から動作の正常性を確認し厳密性を向上させることができるという利点がある。

2.2　従来のシステム監視手法の問題点

運用者は，提供サービスと監視項目の関係を明確に把握している必要がある。特にリアクティブ監視項目の場合は，イベントの発生と同時に影響範囲の特定を行うことが必須である。例えば，表 4 の監視項目の 1 つがイベントとして検知された場合，同時にサービス利用者が正常に WEB ページを閲覧できない状況であるということを判断しなくてはならない。

しかしながら，現在は 1 台の機器上で各種サービスを提供している場合は少なく，複数の機器に機能を分散させて提供する場合が多い。そのため，1 台の機器を用いてサービスを提供する場合に比べて，提供サービスと監視項目と

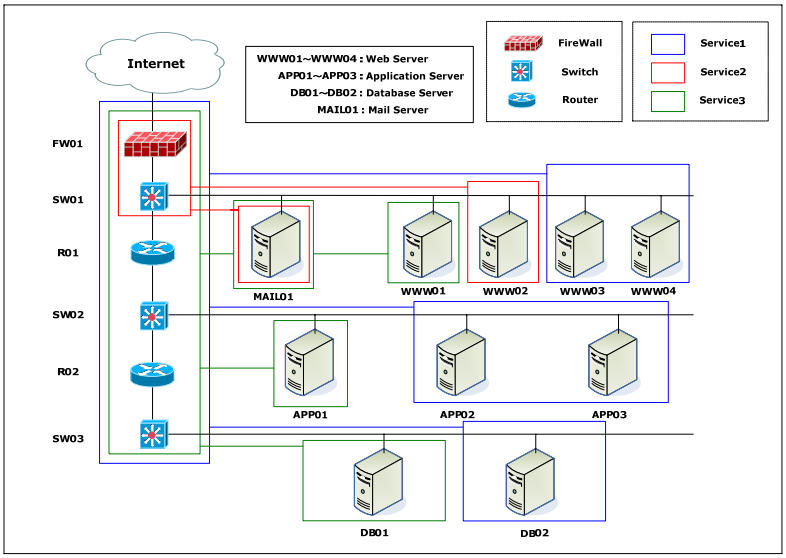

図 1　複数のサービスが混在した複雑なシステムの構成例

の関係把握が難しい。

例えば，図 1 のデータベースサーバ（DB01）において，データベースアプリケーションプロセスのダウンを検知した際，即座に Service3 に影響が出ていると判断できるであろうか。また，図 1 の WEB サーバ（WWW03）において，WEB サーバアプリケーションのプロセスダウンを検知した際，WWW04 にて冗長化されているため Service1 は完全に停止していないと判断できるであろうか。

現在の監視システムの多くは，検知したプロアクティブ監視項目に属するステータス監視項目またはログ監視項目，リアクティブ監視項目に属するステータス監視項目またはログ監視項目に関するイベント内容をそのまま運用者へ通知するのみである。そのため，検知したイベントが各種提供サービスと，どのような依存関係を持っているかという判断に関しては，運用者のスキルレベルおよび熟練度に大きく左右される。

また，どんなにスキルレベルおよび熟練度の高い運用者であっても，監視システムにて 1 度に大量のイベントを検知した場合，提供サービスの影響度を意識しながら対応順位付けをし，実際に対応を行っていくには限界があると考えられる。

これらの問題は，アウトソーシングの依頼を受ける組織または大規模なシステムを複数所有する組織にとってさらに深刻度が増すことになる。例えば，図 1 のように複数の機器を用いて複数のサービスを提供しているシステムが多く存在する図 2 のような環境下においては，提供サービスと監視項目との関係把握がさらに難しくなる。

つまり，現在運用されているシステムの多くは提供サービスと監視項目との関係を明確に把握することが大変困難であり，多くの場合，運用者のスキルレベル，熟練度によってシステムの可用性が左右される状況にある[15]。このような状況下では，運用者の人員交代等によって障害対応時間に大きな差が生じる可能性が高く，同時にシステムの可用性低下を招いてしまう可能性が高い。また，提供サービスと監視項目との関係把握および大量イベント検知時の対応順位づけを系統的に行う方法が無ければ，システムの大規模複雑化に合わせて単に運用者リソースを増加させても，システムの可用性改善を目的とした根本的な解決にはならない。

プロアクティブ監視項目の場合，多くの時間をかけて原因究明やシステム復旧へのアプローチを策定することができるが，リアクティブ監視項目の場合は既にサービス影響が発生してしまっているため，多くの時間をかけることができない。

そのため，リアクティブ監視項目のイベントを検知した際は，大規模複雑なシステムのどこ

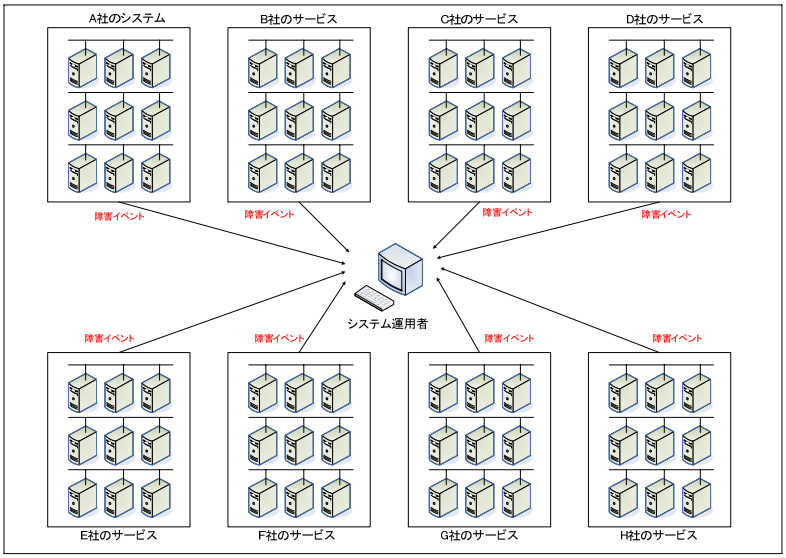

図 2　複数のサービスが混在した複雑なシステムが複数あるシステムの構成例

でサービス影響が発生しているのか，大量に検知したイベントは互いにどのような関係にあるかということを即座に把握する方法が必要である。

3 システム監視手法の改善策

3.1 改善範囲

前章で述べた通り，監視項目を大きく分類すると次の 2 つになる。

- プロアクティブ監視項目による監視
- リアクティブ監視項目による監視

本研究では，サービス影響を伴うイベント発生時の対応について取り扱う。そのため，プロアクティブ監視項目に関しては改善範囲に含めない。

ステータス監視項目による監視は，正常または異常の判断を明確に行うことができる。一方，ログ監視項目は運用者が検知したメッセージ内容に関して調査が必要となる場合が多い。したがって，ログ監視項目にて検知した内容を機械的に処理することは大変難しい。

しかし，システムのリアルタイムな状態や動作に関する内容であれば，ログ監視項目はステータス監視項目で表現できる場合が多い。

例えば，ログ監視項目にて「NFS server ホスト名 not responding」という文字列を含むメッセージを検知した際は，NFS サーバへのアクセスが正常に行えないことを意味する。このログ監視項目は，NFS サーバへの読込み・書込み結果を判断するプログラムで代替可能であり，ステータス監視項目で表現することができる。

また，ログ監視項目にて「Disk full」という文字列を含むメッセージを検知した際は，ハードディスクの領域に問題があることを意味する。このログ監視項目に関しても df コマンドの結果を判断するプログラムで代替可能であり，ステータス監視項目で表現することができる。

即ち，システムの正しい状態を定義できれば，あらゆる監視項目はステータス監視項目として扱うことができ，そのステータスを監視することにより，リアクティブ監視項目のイベントを即座に検知・判別することが可能となる。

そのため，本研究では，正常・異常の判断が明確にできるステータス監視項目のみを取り扱う事とし，ログ監視に関しては改善範囲に含めない。以降，本研究で取り扱うリアクティブ監視項目に属するステータス監視項目を主監視項目と呼ぶこととする。

3.2 改善方法

3.2.1 主監視項目のグループ化

表 3 で示した通り，各種監視項目は目的によってさまざまなものが存在する。そのため，ここでは主監視項目のグループ化を行う。

図 3 は，主監視項目の分類例を示したものである。例えば，監視対象の機器名を Node，Node で定義されている全ての監視項目を

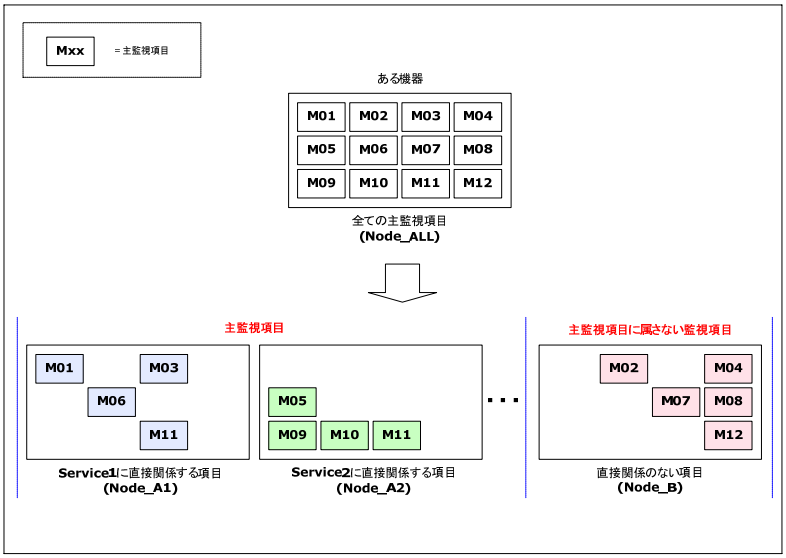

図 3　主監視項目の分類例

表 5　主監視項目の整理例

グループ	主監視項目
Node_A1	M01, M03, M06, M11
Node_A2	M05, M09, M10, M11
Node_B	M02, M04, M07, M08, M12

Node_ALL（M01～M12），Service1 に直接関係する全ての主監視項目を Node_A1, Service2 に直接関係する全ての主監視項目を Node_A2, 主監視項目に属さない全ての監視項目を Node_B とした場合，表 5 のように整理することができる。

既存のシステム環境において，主監視項目のグループ化を行うことは非常に難しい作業である。なぜなら，複数のサービスの監視に同一の主監視項目が用いられる場合，各サービスのモジュールの依存関係と監視項目の依存関係が一致しないため，ある一つのサービスを構成するモジュールの状態だけを特定する監視項目だけを抜き出す事ができないからである。しかし，各監視項目の目的を明確にするためには省略のできない作業である。同時に，このような作業はイベント発生時行うべき作業ではない。そのため，多少時間がかかるとしても，あらゆる資料を調査して根気強く厳密に実施すべきである。

主監視項目のグループ化が完了次第，グループの関連性を分かりやすく表現するためブール演算式を用いて表す。例えば，Node_A1 が正常になるには，全ての主監視項目が正常である必要がある場合，次の式で表すことができる。

$$\text{Node\_A1} = \text{M01} \ \textit{AND} \ \text{M03} \ \textit{AND} \ \text{M06} \ \textit{AND} \ \text{M11} \quad (1)$$

具体的に，主監視項目 M06 が正常でなくなった場合，ブール値を True（主監視項目正常時）および False（主監視項目異常時）で表し演算すると次のようになる。

$$\begin{aligned} \text{Node\_A1} &= \text{True} \ \textit{AND} \ \text{True} \ \textit{AND} \ \text{False} \ \textit{AND} \ \text{True} \\ &= \text{False} \end{aligned} \quad (2)$$

また，Node_A2 に関して，主監視項目 M10 または M11 のどちらか一方が正常であれば良く，それ以外は全て正常である必要がある場合，次の式で表すことができる。

表 6　各種サービスに関する機器のグループ例

機器	Service1	Service2	Service3
MAIL01		○	○
WWW01			○
WWW02		○	
WWW03	○		
WWW04	○		
APP01			○
APP02	○		
APP03	○		
DB01			○
DB02	○		
FW01	○	○	○
SW01	○	○	○
SW02	○		○
SW03	○		○
R01	○		○
R02	○		○

$$\begin{aligned} \text{Node\_A2} \;=\; & \text{M05} \;\; \textit{AND} \;\; \text{M09} \\ & \textit{AND} \;\; (\text{M10} \;\; \textit{OR} \;\; \text{M11}) \end{aligned} \tag{3}$$

具体的に，主監視項目 M10 が正常でなくなった場合，ブール値を True（主監視項目正常時）および False（主監視項目異常時）で表し演算すると次のようになる。

$$\begin{aligned} \text{Node\_A2} \;=\; & \text{True} \;\; \textit{AND} \;\; \text{True} \\ & \textit{AND} \;\; (\text{False} \;\; \textit{OR} \;\; \text{True}) \\ =\; & \text{True} \end{aligned} \tag{4}$$

このようにブール演算式で表現することで，発生イベントに対応する主監視項目と影響サービスを運用者へ伝えることが可能となる。

3.2.2　機器のグループ化

1 台の機器を用いて各種サービスが提供される場合は，主監視項目のグループ化によってイベント検知時の各種提供サービスの影響範囲を特定することができる。しかし，現在は多くの機器を用いて各種サービスを提供している場合が多いため，機器のグループ化が必要である。例えば，図 1 の各種サービスに関して機器をグループ化すると表 6 のようになる。

主監視項目のグループ化と同様，既存のシステム環境において，機器のグループ化を行うことは非常に難しい作業である。なぜなら，同一の機器を用いて複数のサービスを提供する場合，サービスの見た目の構成とサービスを構成するためのハードウェア構成が一致しないことが多いからである。

しかし，各種提供サービスに利用されている各機器の目的を明確にするためには省略のできない作業である。同時に，このような作業はイベント発生時行うべき作業ではない。そのため，多少時間がかかるとしても，あらゆる資料を調査して根気強く厳密に実施すべきである。

機器のグループ化が完了次第，主監視項目のグループ化の時と同様，グループの関連性を分かりやすく表現するためブール演算式を用いて表す。例えば，図 1 および表 6 より，Service1～Service3 は次のブール演算式で表すことができる。

$$\begin{aligned} \text{Service1} \;=\; & (\text{WWW03} \;\; \textit{OR} \;\; \text{WWW04}) \\ & \textit{AND} \;\; (\text{APP02} \;\; \textit{OR} \;\; \text{APP03}) \\ & \textit{AND} \;\; \text{DB02} \;\; \textit{AND} \;\; \text{FW01} \\ & \textit{AND} \;\; \text{SW01} \;\; \textit{AND} \;\; \text{R01} \\ & \textit{AND} \;\; \text{SW02} \;\; \textit{AND} \;\; \text{R02} \\ & \textit{AND} \;\; \text{SW03} \end{aligned} \tag{5}$$

表 7　各種サービスに関係する主監視項目と機器の関係例

機器	Service1 に関係する主監視項目	Service2 に関係する主監視項目	Service3 に関係する主監視項目
MAIL01		*MAIL01_A2*	*MAIL01_A3*
WWW01			*WWW01_A3*
WWW02		*WWW02_A2*	
WWW03	*WWW03_A1*		
WWW04	*WWW04_A1*		
APP01			*APP01_A3*
APP02	*APP02_A1*		
APP03	*APP03_A1*		
DB01			*APP01_A3*
DB02	*DB02_A1*		
FW01	*FW01_A1*	*FW01_A2*	*FW01_A3*
SW01	*SW01_A1*	*SW01_A2*	*SW01_A3*
SW02	*SW02_A1*		*SW02_A3*
SW03	*SW03_A1*		*SW03_A3*
R01	*R01_A1*		*R01_A3*
R02	*R02_A1*		*R02_A3*

Service2 = WWW02 *AND* MAIL01 *AND* FW01 *AND* SW01 (6)

Service3 = WWW01 *AND* MAIL01 *AND* APP01 *AND* DB01 *AND* FW01 *AND* SW01 *AND* R01 *AND* SW02 *AND* R02 *AND* SW03 (7)

具体的に, 機器 MAIL01 でハードウェア障害が発生した場合, ブール値を True（機器正常動作時）および False（機器異常動作時）で表し演算すると次のようになる。

Service1 = (True *OR* True) *AND* (True *OR* True) *AND* True *AND* True *AND* True *AND* True *AND* True *AND* True
= True (8)

Service2 = True *AND* False *AND* True *AND* True
= False (9)

Service3 = True *AND* False *AND* True *AND* True *AND* True *AND* True *AND* True *AND* True
= False (10)

このようにブール演算式で表現することで, 運用者は各機器と各種提供サービスの関係を把握することが可能となる。

3.2.3　主監視項目および機器のグループ化情報の結合

主監視項目のグループ化は，イベントとして検知した主監視項目より各種提供サービスの影響度を特定することができるが，サービスを提供する上で必要な機器の依存関係に関しては特定できない。

機器のグループ化は，ハードウェア障害等のイベントが発生した際，対象機器が各種提供サービスとどのような関係にあるかを特定できるが，主監視項目との依存関係に関しては特定できない。

そのため，主監視項目および機器のグループ

化情報を結合すれば，システムが複数の機器によって構成されている場合であっても，イベント検知時に各種提供サービスの影響範囲を特定することができる。表 7 は，図 1 の各種サービスに関係する主監視項目と機器の関係例を示したものである。

表 7 の内容を基に，各種提供サービスの関係をブール演算式で表すと次のようになる。

$$\begin{aligned}\text{Service1} = {} & (\text{WWW03\_A1 } \textit{OR} \text{ WWW04\_A1}) \\ & \textit{AND } (\text{APP02\_A1 } \textit{OR} \text{ APP03\_A1}) \\ & \textit{AND } \text{DB02\_A1 } \textit{AND } \text{FW01\_A1} \\ & \textit{AND } \text{SW01\_A1 } \textit{AND } \text{R01\_A1} \\ & \textit{AND } \text{SW02\_A1 } \textit{AND } \text{R02\_A1} \\ & \textit{AND } \text{SW03\_A1}\end{aligned} \tag{11}$$

$$\begin{aligned}\text{Service2} = {} & \text{WWW02\_A2 } \textit{AND } \text{MAIL01\_A2} \\ & \textit{AND } \text{FW01\_A2 } \textit{AND } \text{SW01\_A2}\end{aligned} \tag{12}$$

$$\begin{aligned}\text{Service3} = {} & \text{WWW01\_A3 } \textit{AND } \text{MAIL01\_A3} \\ & \textit{AND } \text{APP01\_A3 } \textit{AND } \text{DB01\_A3} \\ & \textit{AND } \text{FW01\_A3 } \textit{AND } \text{SW01\_A3} \\ & \textit{AND } \text{R01\_A3 } \textit{AND } \text{SW02\_A3} \\ & \textit{AND } \text{R02\_A3 } \textit{AND } \text{SW03\_A3}\end{aligned} \tag{13}$$

例えば，WWW03_A1 に属するある主監視項目の 1 つが障害となり，その他の主監視項目が正常であった場合，ブール演算式の演算結果は次の通りとなり，Service1 は完全に停止していないという事が確認できる。

$$\begin{aligned}\text{Service1} = {} & (\text{False } \textit{OR } \text{True}) \\ & \textit{AND } (\text{True } \textit{OR } \text{True}) \\ & \textit{AND } \text{True } \textit{AND } \text{True} \\ & \textit{AND } \text{True } \textit{AND } \text{True} \\ & \textit{AND } \text{True } \textit{AND } \text{True} \\ & \textit{AND } \text{True} \\ = {} & \text{True}\end{aligned} \tag{14}$$

なお, (14)式の演算結果は True となっているが，実際は縮退稼動している状態である。縮退稼動をしているかどうかに関しては，Service1 の主監視項目を全て AND 演算した結果と(14)式の演算結果を用いて確認することができる。Service1 の主監視項目を全て AND 演算した結果を Service1_and，(14)式の演算結果を Service1_real とし，縮退稼動判定式を Service_av で表すと次の通りとなる。

$$\begin{aligned}\text{Service1\_av} = {} & \text{Service1\_real} \\ & \textit{AND } \text{Service1\_and} \\ = {} & \text{True } \textit{AND } \text{False} \\ = {} & \text{False}\end{aligned} \tag{15}$$

演算結果が True であれば縮退稼動をしておらず，False であれば縮退稼動をしている状態となる。(15)式の演算結果より，Service1_av は False となるため, 縮退稼動をしていると判断することができる。

また，機器 MAIL01 で動作しているメールサーバアプリケーションのダウンを検知したとしても，MAIL01_A2 のみが False となり，MAIL01_A3 が True という状態であれば，結果として関連する Service2 および Service3 の正常性は次のようになる。

$$\begin{aligned}\text{Service2} = {} & \text{False } \textit{AND } \text{True} \\ & \textit{AND } \text{True } \textit{AND } \text{True} \\ = {} & \text{False}\end{aligned} \tag{16}$$

$$\begin{aligned}\text{Service3} = {} & \text{True } \textit{AND } \text{True} \\ & \textit{AND } \text{True } \textit{AND } \text{True} \\ & \textit{AND } \text{True } \textit{AND } \text{True} \\ & \textit{AND } \text{True } \textit{AND } \text{True} \\ & \textit{AND } \text{True } \textit{AND } \text{True} \\ = {} & \text{True}\end{aligned} \tag{17}$$

このような場合，Service2，Service3 ともに機器 MAIL01 を用いているが, 使用しているメールサーバアプリケーションが異なるため，結果として Service2 は完全に停止しているが，Service3 は完全に停止していないという事が確認できる。

各種提供サービスを 1 つにまとめて定義することが難しい場合は，管理のしやすい単位に分割して定義をしても良い。例えば，各種サービス ServiceA～ServiceH が連携することによって，エンドユーザ向けサービス（ServiceZ）を提供しており，ServiceC～ServiceE，ServiceG～ServiceH に関しては, いずれかが正常であれば良い場合，次のブール演算式で表すことができる。

$$\begin{aligned}\text{ServiceZ} = {} & \text{ServiceA } \textit{AND } \text{ServiceB} \\ & \textit{AND } (\text{ServiceC } \textit{OR } \text{ServiceD} \\ & \textit{OR } \text{ServiceE}) \textit{ AND } \text{ServiceF} \\ & \textit{AND } (\text{ServiceG } \textit{OR } \text{ServiceH})\end{aligned} \tag{18}$$

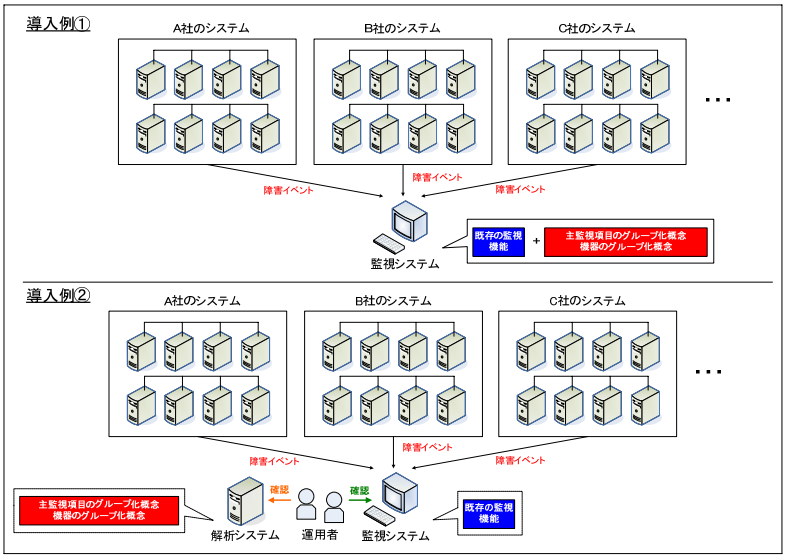

図 4　本研究概念の導入例

具体的に，ServiceD および ServiceH が異常であった場合，ブール値を True（サービス正常時）および False（サービス異常時）で表し演算すると ServiceZ は次のようになる。

$$\begin{aligned} \text{ServiceZ} &= \text{True} \ \textit{AND} \ \text{True} \\ &\quad \textit{AND} \ (\text{True} \ \textit{OR} \ \text{False} \\ &\quad \textit{OR} \ \text{True}) \ \textit{AND} \ \text{True} \\ &\quad \textit{AND} \ (\text{True} \ \textit{OR} \ \text{False}) \\ &= \text{True} \end{aligned} \tag{19}$$

このように，主監視項目および機器のグループ化情報の結合は，主監視項目と提供サービス，機器と提供サービス，主監視項目と機器に関する関係を明確にすることができ，イベント発生時の運用者の判断レベル統一化を図ることが可能となる。

4　導入方法

4.1　導入パターン

導入方法としては，次の 2 通りが考えられる。

- 既存の監視システムに本研究の概念を取り入れる
- 既存の監視システムとは別に，本研究の概念を反映させたシステムを用意する

図 4 は，本研究概念の導入例を示したものである。導入例①は，既存の監視システムに本研究概念を取り入れ，今までと同様にシステム監視を行う場合の例である。一見，導入例①は現実的にみえるが，商用の監視ソフトウェアを用いている場合は，監視プログラムの直接改修をすることができない。また，商用の監視ソフトウェア以外を用いている場合であっても，監視システムの直接改修は大変リスクの高い行為であるため現実的ではない。

導入例②は，既存の監視システムとは別に本研究概念を取り入れたシステムを用意し，運用者を介して互いに連携させる場合の例である。例えば，既存の監視システムにて主監視項目に関するイベントを検知した際，運用者は本研究概念を取り入れた解析システムで影響範囲を特定し，各種提供サービスの正常性を確認することができる。

この方法は，監視システムの情報のみを取り扱い，監視システム自体のコード変更等を必要としないため現実的な方法と考えられる。なお，導入例②を用いた具体的な実装実験例に関しては次章で紹介する。

4.2　具体的な実装実験

4.2.1　主監視項目および機器のグループ化情報の整理と結合

ここでは，現実的な導入方法である図 4 の導入例②をモデルに具体的な動作実験結果を紹介する。図 1 の Service1（以下，Service1）に関して，主監視項目のグループ化が図 5 の通り行われ，次のブール演算式で表現できたと仮定する。

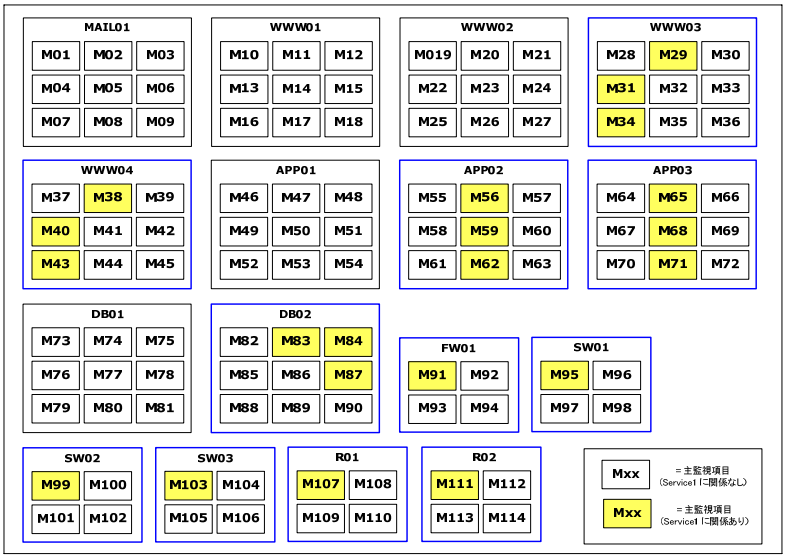

図 5　Service1 の主監視項目グループ化

WWW03_A1 = M29 *AND* M31 *AND* M34 (20)

WWW04_A1 = M38 *AND* M40 *AND* M43 (21)

APP02_A1 = M56 *AND* M59 *AND* M62 (22)

APP03_A1 = M65 *AND* M68 *AND* M71 (23)

DB02_A1 = M83 *AND* M8 4 *AND* M87 (24)

FW01_A1 = M91 (25)

SW01_A1 = M95 (26)

SW02_A1 = M99 (27)

SW03_A1 = M103 (28)

R01_A1 = M107 (29)

R02_A1 = M111 (30)

また，機器のグループ化に関しては，次のブール演算式で表現できたと仮定する。

$$\begin{aligned} \text{Service1} = {} & (\text{WWW03 } \mathit{OR} \text{ WWW04}) \\ & \mathit{AND} \ (\text{APP02 } \mathit{OR} \text{ APP03}) \\ & \mathit{AND} \ \text{DB02 } \mathit{AND} \text{ FW01} \\ & \mathit{AND} \ \text{SW01 } \mathit{AND} \text{ SW02} \\ & \mathit{AND} \ \text{SW03 } \mathit{AND} \text{ R01} \\ & \mathit{AND} \ \text{R02} \end{aligned} \quad (31)$$

(20)～(31)式より，Service1 の正常性判定式は次の通りとなる。

$$\begin{aligned} \text{Service1} = {} & (\text{WWW03\_A1 } \mathit{OR} \text{ WWW04\_A1}) \\ & \mathit{AND} \ (\text{APP02\_A1 } \mathit{OR} \text{ APP03\_A1}) \\ & \mathit{AND} \ \text{DB02\_A1 } \mathit{AND} \text{ FW01\_A1} \\ & \mathit{AND} \ \text{SW01\_A1 } \mathit{AND} \text{ SW02\_A1} \\ & \mathit{AND} \ \text{SW03\_A1 } \mathit{AND} \text{ R01\_A1} \\ & \mathit{AND} \ \text{R02\_A1} \end{aligned} \quad (32)$$

(20)～(30)式に主監視項目の状態（正常時を True，異常時を False とする）を代入し，(32)式の演算を行うことによって Service1 の正常性を明確にすることが可能となる。

4.2.2　解析システムの実装

(32)式の演算を行う仕組みはさまざまな方法が考えられるが，今回は図 6 のような仕組みで実現させた。

運用者と解析システム間のインタフェースに apache（WEB サーバ）[16]，データベース問合せやブール演算の実行に PHP プログラム[17]，機器や主監視項目，提供サービスと主監視項目との関係定義に PostgreSQL[18]を使用した。

A) 監視システムにて検知したイベントは運用者に通知され，運用者によって該当の主監視項目の入力が行われる（図 6 ①，②）。

B) 入力された主監視項目は，マスタープログラムへ渡されデータベース問合せが行われる。データベースには機器，主監視項目，提供サービスと主監視項目との関係が定義されており，ブール演算を行うために必要な情報が管理されている（図 6 ③，④，⑤）。

C) データベースからの問合せ結果を基に，マスタープログラムはブール演算を行い，その演算結果をユーザインタフェースへ渡す（図 6 ⑥，⑦，⑧）。

D) 運用者は解析システムからの結果より，

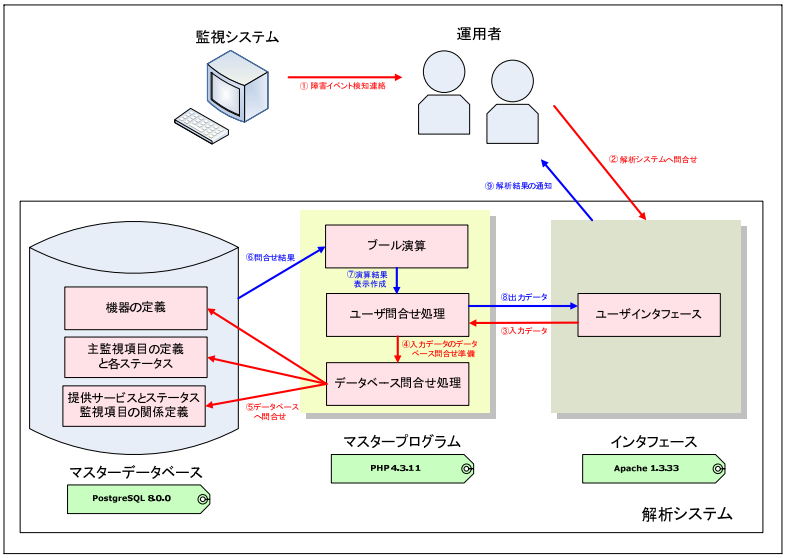

図 6　解析システムの実装例

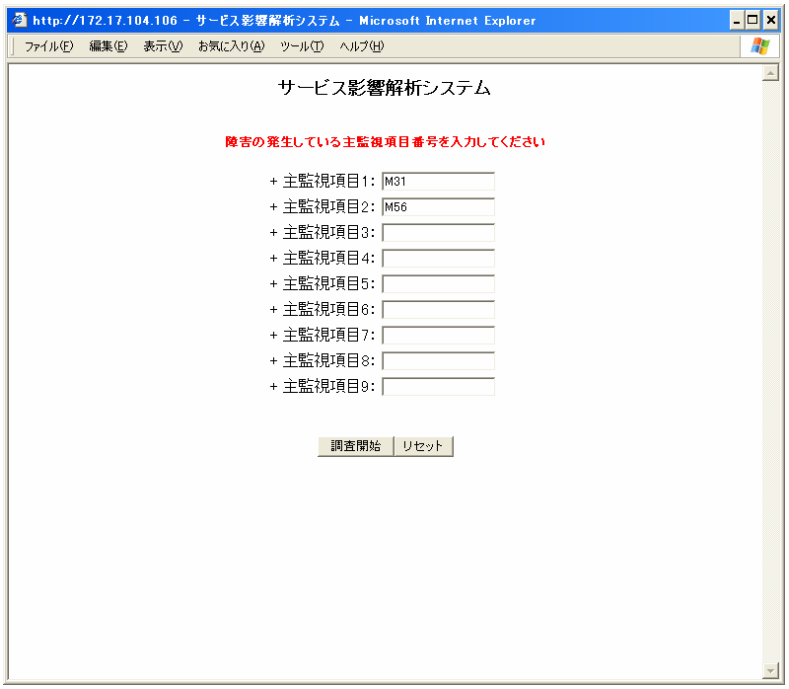

図 7　解析システムの入力インタフェース例

提供サービスの正常性を認識する（図 6 ⑨）。

4.2.3　具体的な動作例

図 7 は運用者と解析システム間の入力インタフェース例である。監視システムにて検知したイベントは，運用者によって図 7 のインタフェースを用いて情報の入力が行われる。

例えば，監視システムにて検知したイベントの主監視項目番号が M31 および M56 ならば，図 7 のテキストボックスにその内容を入力する。なお，主監視項目に関しては，一意に判別できる番号または記号が付与されている必要がある。

図 7 より入力された主監視項目は，図 6 の仕組みを用いて解析が行われる。主監視項目番号 M31 および M56 が False，それ以外の主監視項目は全て True として(20)～(30)を演算すると次の通りとなる。

$$\begin{aligned} \mathrm{WWW03\_A1} &= \mathrm{True}\ \mathit{AND}\ \mathrm{False}\ \mathit{AND}\ \mathrm{True} \\ &= \mathrm{False} \end{aligned} \tag{33}$$

図 8　解析システムの出力インタフェース例（M31 および M56 が False である時）

$$\mathrm{WWW04\_A1} = \mathrm{True}\ \mathit{AND}\ \mathrm{True}\ \mathit{AND}\ \mathrm{True} = \mathrm{True} \tag{34}$$

$$\mathrm{APP02\_A1} = \mathrm{False}\ \mathit{AND}\ \mathrm{True}\ \mathit{AND}\ \mathrm{True} = \mathrm{False} \tag{35}$$

$$\mathrm{APP03\_A1} = \mathrm{True}\ \mathit{AND}\ \mathrm{True}\ \mathit{AND}\ \mathrm{True} = \mathrm{True} \tag{36}$$

$$\mathrm{DB02\_A1} = \mathrm{True}\ \mathit{AND}\ \mathrm{True}\ \mathit{AND}\ \mathrm{True} = \mathrm{True} \tag{37}$$

$$\mathrm{FW01\_A1} = \mathrm{True} \tag{38}$$

$$\mathrm{SW01\_A1} = \mathrm{True} \tag{39}$$

$$\mathrm{SW02\_A1} = \mathrm{True} \tag{40}$$

$$\mathrm{SW03\_A1} = \mathrm{True} \tag{41}$$

$$\mathrm{R01\_A1} = \mathrm{True} \tag{42}$$

$$\mathrm{R02\_A1} = \mathrm{True} \tag{43}$$

機器のグループ化は(31)式の通りであり，(33)～(43)式を(32)式に代入すると次の通りとなり，Service1 が完全に停止していないことが確認できる。

$$\begin{aligned}\mathrm{Service1} = {} & (\mathrm{False}\ \mathit{OR}\ \mathrm{True}) \\ & \mathit{AND}\ (\mathrm{False}\ \mathit{OR}\ \mathrm{True}) \\ & \mathit{AND}\ \mathrm{True}\ \mathit{AND}\ \mathrm{True} \\ & \mathit{AND}\ \mathrm{True}\ \mathit{AND}\ \mathrm{True} \\ & \mathit{AND}\ \mathrm{True}\ \mathit{AND}\ \mathrm{True} \\ & \mathit{AND}\ \mathrm{True} \\ = {} & \mathrm{True}\end{aligned} \tag{44}$$

図 8 は運用者と解析システム間の出力インタフェース例（M31 および M56 が False であるとき）である。(44)式の演算結果は，図 8 の通り運用者へ伝えられる。なお，(15)式の縮退稼動判定式を演算すると次の通りとなり，Service1 はサービス提供できているが，縮退稼動中であることが分かる。

$$\begin{aligned}\mathrm{Service1\_av} = {} & \mathrm{Service1\_real} \\ & \mathit{AND}\ \mathrm{Service1\_and} \\ = {} & \mathrm{True}\ \mathit{AND}\ \mathrm{False} \\ = {} & \mathrm{False}\end{aligned} \tag{45}$$

このことは，図 8 より視覚的に確認することが可能である。具体的には，主監視項目の一部で False（赤色）の項目があるが，結果としてサービス影響なしという判断が行われているということは，縮退稼動によってサービス提供を維持している状態である判断することができる。

同様に主監視項目番号 M84 が False，それ以外のステータス監視項目は全て True として(20)～(30)式を演算すると次の通りとなり，Service1 に影響が発生していることが確認できる。

$$\mathrm{WWW03\_A1} = \mathrm{True}\ \mathit{AND}\ \mathrm{True}\ \mathit{AND}\ \mathrm{True} = \mathrm{True} \tag{46}$$

$$\mathrm{WWW04\_A1} = \mathrm{True}\ \mathit{AND}\ \mathrm{True}\ \mathit{AND}\ \mathrm{True} = \mathrm{True} \tag{47}$$

$$\mathrm{APP02\_A1} = \mathrm{True}\ \mathit{AND}\ \mathrm{True}\ \mathit{AND}\ \mathrm{True} = \mathrm{True} \tag{48}$$

主監視項目番号	対象機器	対象機器の説明	ステータス
M29	WWW03	Web Server #3	true
M31	WWW03	Web Server #3	true
M34	WWW03	Web Server #3	true
M38	WWW04	Web Server #4	true
M40	WWW04	Web Server #4	true
M43	WWW04	Web Server #4	true
M56	APP02	Application Server #2	true
M59	APP02	Application Server #2	true
M62	APP02	Application Server #2	true
M65	APP03	Application Server #3	true
M68	APP03	Application Server #3	true
M71	APP03	Application Server #3	true
M83	DB02	Database Server #2	true
M84	DB02	Database Server #2	false
M87	DB02	Database Server #2	true
M91	FW01	FireWall #1	true
M95	SW01	Switch #1	true
M99	SW02	Switch #2	true
M103	SW03	Switch #3	true
M107	R01	Router #1	true
M111	R02	Router #2	true

図 9　解析システムの出力インタフェース例（M84 が False である時）

$$\text{APP03\_A1} = \text{True } \textit{AND } \text{True } \textit{AND } \text{True} = \text{True} \tag{49}$$

$$\text{DB02\_A1} = \text{True } \textit{AND } \text{False } \textit{AND } \text{True} = \text{False} \tag{50}$$

$$\text{FW01\_A1} = \text{True} \tag{51}$$

$$\text{SW01\_A1} = \text{True} \tag{52}$$

$$\text{SW02\_A1} = \text{True} \tag{53}$$

$$\text{SW03\_A1} = \text{True} \tag{54}$$

$$\text{R01\_A1} = \text{True} \tag{55}$$

$$\text{R02\_A1} = \text{True} \tag{56}$$

$$\begin{aligned}\text{Service1} = {} & (\text{True } \textit{OR } \text{True}) \\ & \textit{AND } (\text{True } \textit{OR } \text{True}) \\ & \textit{AND } \text{False } \textit{AND } \text{True} \\ & \textit{AND } \text{True } \textit{AND } \text{True} \\ & \textit{AND } \text{True } \textit{AND } \text{True} \\ & \textit{AND } \text{True} \\ = {} & \text{False}\end{aligned} \tag{57}$$

図 9 は運用者と解析システム間の出力インタフェース例(M84 が False であるとき)である。具体的な実装実験では，Service1 に関する例を示したが，Service2 および Service3 に関しても同様に定義を行えば同様に実装可能である。

5 研究の考察

従来の監視手法を用いた場合，イベントを検知した際に監視項目の大分類（プロアクティブ監視項目またはリアクティブ監視項目）を判別しなくてはならなかった。もしも，検知したイベントがリアクティブ監視項目に属するものであれば，提供サービスとの依存関係を調査した上で早急に影響範囲を特定し，サービス影響度の評価を行う必要があった。　しかし，各監視項目と提供サービスの依存関係を把握するには多くの時間を要する可能性が高い。例えば，N 個の監視項目の中で，1 個，2 個，3 個… の監視項目を使用する組合せが，それぞれ 1 個ずつ出現する時の組合せ総数 Monitor_total は次の通り表現できる。

$$\text{Monitor\_total} = {}_nC_1 + {}_nC_2 + \ldots + {}_nC_{n-1} + {}_nC_n \tag{58}$$

一方，二項定理より

$$(x+y)^n = \sum_{k=0}^{n} {}_nC_k x^k y^{n-k} \tag{59}$$

ここで，$x = y = 1$ とすると

$$2^n - 1 = {}_nC_1 + {}_nC_2 + \ldots + {}_nC_{n-1} + {}_nC_n \tag{60}$$

を得る。(58)式の右辺＝(60)式の右辺となることから

$$\text{Monitor\_total} = 2^n - 1 \tag{61}$$

となる。実際には，${}_nC_m$ が複数個存在することが予想され，(61)式よりも大きな組合せ総数を扱わなくてはならないことになる。当然，イベ

ントに対応する運用者のスキルレベルや熟練度等を加味すると対応の状況は変わるが，監視項目の絶対数から比べればその有効性は不確実と考えられる。また, いくらスキルレベルが高く，熟練度の高い運用者であっても，(61)式で表されるような膨大な監視項目の組合せを基に，常に正しい依存関係を特定できるとは限らない。たとえ提供サービスの正しい組合せを特定できたとしても，冗長構成環境やディザスタリカバリ環境では，検知された監視項目の組合せによっては，与えられた時間内にサービス影響度を正しく評価しきれない可能性も大きくなる。

本研究では，主監視項目と提供サービス，機器と提供サービス，主監視項目と機器に関する依存関係を明確にし，運用者のスキルレベルや熟練度によらず，主監視項目の総組合せから提供サービスの正しい組合せを特定する方法を提案した。加えて，提供サービスの正しい組合せをブール演算式で表すことで，冗長構成環境やディザスタリカバリ環境といった複雑なシステムにおいても，正しくサービス影響度を評価できることを示した。これらの結果より，本研究概念の導入によって，アウトソーシングの依頼を受ける組織または大規模なシステムを複数所有する組織で，各種提供サービスの正常性を効率的に把握することが可能となった。さらに，主監視項目に属するイベントの検知時に行っていたドキュメント調査，リバースエンジニアリングの実施等といった運用者のスキルに依存する作業を大きく減少させ，結果的に運用者の対応品質を高品質に均一化する事も可能となる。

監視システムの機能は日々向上しているが，検知したイベントを扱う方法はシステム運用を行っている現場に依存している場合が多い。そのため，確実にイベント検知ができる環境があっても，システムの高可用性維持を目的とした効率的な利用ができているとは限らない。特に，(61)式で表される膨大な監視項目の組合せがあるにもかかわらず，異常を検知してからサービスの影響度を評価する現在の監視方法では，巨大システムの監視に限界が訪れる事は明白である。したがって，事前にサービスの正常性及び系統的なサービス影響度の評価方法を定義しておく事が強く求められるであろう。そのため，本研究概念によって可能となる検知イベントの効率的かつ確実性の高い取り扱いは，システムの高可用性維持を目的とした運用方法の改善に大きな貢献をもたらすに違いない。また，IPv6 を活用したユビキタス社会が訪れようとしている今日，ネットワーク管理下におかれるサービス・機器は急速に増加の一途をたどると思われる。本研究では現存のネットワークシステム上での監視に主眼をおいているが，近未来に求められる監視システムにも同様に応用でき，重要な指針を与えると期待される。

6 今後の検討

本研究概念は，誤った情報で主監視項目のグループ化定義，機器のグループ化定義，主監視項目と機器のグループ化情報の結合定義を行ってしまうと，提供サービスの正常性判断を正しく行うことができず，さらに状況を悪化させてしまう可能性がある。この問題は運用開始前の厳密なチェック，運用開始後の動作傾向チェックで十分に回避可能であるが，さらに利便性の高い厳密なチェック方法に関して引続き検討していきたいと考えている。

本研究概念の導入方法は，4 章で述べた内容以外にもさまざまな方法が考えられる。具体的な導入方法に関しては，既存のシステム監視環境を考慮の上, 柔軟に検討していくべきである。しかし，導入方法によってはシステム監視情報等の 2 重管理を発生させてしまう可能性も高い。この問題に関しても運用で十分にカバーできると思われるが，柔軟かつ厳密な管理を可能とする方法を引続き検討していきたいと考えている。

謝辞

本論文の執筆にあたり，指導・協力をしていただいた株式会社東京技術計算コンサルタント堀江晋吉氏に厚くお礼申し上げます。

参考文献

[1] 菅野幹人, 大越冬彦, 村田篤, “大規模ネットワーク・サーバ運用監視向けインシデント管理システム”, 情報処理学会研究報

告 分散システム/インターネット運用技術，No.38，pp.49-54，Mar.2006.
[2] 笠原浩介，敷田幹文，“大規模ネットワークでのフレキシブルな階層構造によるサーバ監視システムの提案”，情報処理学会研究報告 マルチメディア通信と分散処理，No.107，pp.61-66，Nov.2004.
[3] 後藤宏志，敷田幹文，“サーバの依存関係を考慮したログ情報による障害監視支援の提案” ，情報処理学会研究報告 マルチメディア通信と分散処理，No.121，pp.37-42，Nov.2006.
[4] 長田智和，谷口祐治，玉城史朗，“大規模分散ネットワーク運用管理システムの提案”，情報処理学会研究報告 分散システム/インターネット運用技術，No.113，pp.31-36，Dec.2000.
[5] 伊藤武，礒川弘実，萱島信，寺田真敏，“インターネットサービスの特徴を考慮したサーバ稼動監視システムの実装”，情報処理学会研究報告 分散システム/インターネット運用技術, No.66, pp.25-30, Jul.1998.
[6] 日本ヒューレット・パッカード株式会社 日本 HP HP OpenView, http://h50146.www5.hp.com/products/software/management/openview/.
[7] 株式会社野村総合研究所 Senju Family 理想の IT サービスマネージメントの実現へ，http://senjufamily.nri.co.jp/.
[8] BMC ソフトウェア株式会社 オープンシステム管理ソリューション BMC Performance Manager, http://www.bmc.com/ja_JP/products2/L2-infra.html.
[9] 日本 CA 株式会社 CA Unicenter Network and Systems Management, http://ca.com/jp/products/product.aspx?ID=2869.
[10] 株式会社日立製作所 総合システム運用管理 JP1 Version8 Home Pages, http://www.hitachi.co.jp/Prod/comp/soft1/jp1/.
[11] Nagios: Home，http://www.nagios.org/, 2007-12-17.
[12] 株式会社 NTT データ Hinemos | NTT データ, http://www.nttdata.co.jp/services/hinemos/index.html.
[13] Office of Government Commerce，“サービスサポート”，TSO 出版，2003.
[14] Office of Government Commerce，“サービスデリバリ”，TSO 出版，2003.
[15] 森一，敷田幹文，“サーバ依存関係を考慮したシステム構成管理の支援法”，情報処理学会論文誌, Vol.41, No.12, pp.940-948, Apr.2005.
[16] Welcome! - The Apache Software Foundation，http://www.apache.org/.
[17] PHP: Hypertext Preprocessor, http://jp.php.net/.
[18] PostgreSQL: The world's most advanced open source database, http://www.postgresql.org/.

著者略歴

2006 年株式会社野村総合研究所入社。現在は，NRI セキュアテクノロジーズ株式会社へ出向中。システム監視設計，セキュリティデバイスを用いたサービス提供に従事。同時に 2006 年より信州大学大学院総合工学系研究科博士課程に入学。システム可用性の向上を目指した監視手法の研究を行っている。