

運用管理システムにおける監視対象モデルを用いた 障害影響範囲分析の課題と解決策

細川 武彦†, 森 信胤†、虎渡 昌史‡

†三菱電機(株)、‡三菱電機インフォメーションシステムズ(株)

今日、企業の経営活動は情報システムなくしては遂行できず、情報システムの不慮の停止やサービス品質の低下は、企業の信用を失い顧客離れにも繋がる大きな悪影響をもたらす。本稿では監視対象システムの標準情報モデル(CIM)を用いたモデル化に関して説明し、モデル化されたシステム構成情報を用いた障害の影響範囲分析における課題を整理し、対策を検討した結果について報告する。

Subjects and Solutions on Business Impact Analysis using Common Information Model

Takehiko HOSOKAWA †, Nobutane MORI † and Masashi TORATO ‡

† Mitsubishi Electric Corporation,

‡ Mitsubishi Electric Information Systems Corporation

Today, the management activity of the enterprise cannot be accomplished without the information system. An unexpected halt of the information system and the decrease in the system quality cause a big adverse effect to the business activity. In this paper, the subjects and the solutions is described to realize business impact analysis using CIM(Common Information Model).

1. はじめに

今日、企業の経営活動は情報システムなくしては遂行できない状況となっている。情報システムの不慮の停止やサービス品質の低下は、業務停止による損失だけでなく、企業の信用を失い顧客離れにも繋がる大きな悪影響をもたらす。情報システムの安定稼働維持は、企業にとって必達事項であり、経営者にとっても大きな関心事である。情報システムの安定稼働維持のためには、業務復旧のための対処箇所を迅速、正確に特定するための障害箇所分析、適切な優先度判断、緊急度判断のもとに対処するための障害影響分析の実現が必須であるが、現在のシステム障害対応には以下のような課題がある。

個人スキル依存：システム障害への対応は、システムを熟知した限られたエキスパート要員により実施されているケースが多く属人的である。
分析ツールの実用性：現在の障害分析を行うツールには次のような課題があり、特に大規模システム、構成変更の多いシステムに対しては実用レベルにないといえる。

- 分析のためのデータ(構成情報、依存関係、障害検知メッセージ等)は、各ツール独自のデータベースへの登録が必要であり、汎用性がない。
- データベース、分析ロジックともに拡張性、柔軟性が乏しく、限定された範囲の分析にしか利用できない。
- データの登録は「手入力」的であり、大規模で変化の多いシステムにとってはデータの初期登録や最新化維持管理は大きな負担になる。

本稿では、障害影響範囲分析を実現するために、現在 DMTF で標準化されている共通情報モデル(CIM)の関係情報を用いる方式について検討し、その課題と解決策をまとめる。具体的には2章において、背景となる「監視対象システムの変遷」、「管理対象のモデル化技術」、「障害影響分析技術」について簡単に説明する。3章では、これらの技術を用いて障害影響分析を行うための監視対象システムのモデル化についてとその課題についてまとめる。4章で課題を解決するために提案する方式に関して述べ、最後に5章に今後の展開等についてまとめる。

2. 背景

2.1. 監視対象システムの変遷

現在の計算機システムは、「安定性」、「信頼性」、「性能」、「利用効率」、「コストパフォーマンス」などの向上を目的とした様々な技術が適用されており、その規模、複雑度は増加の一途を辿っている。これら複雑化の要因となる技術は大別すると以下のように整理することが出来る(図 1)。

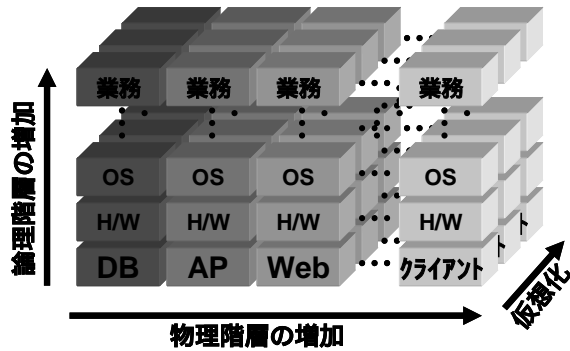


図 1 複雑化の要因

- 物理階層(ティア)の増加：ホスト/端末といった簡単だった構成から、クライアント/サーバ型、Web3 階層型といったシステム構成に変遷しており、更にセキュリティ対策や Web サービスへの対応などにより多階層化が進んでいる。
- 論理階層(レイヤ)の増加：HW/業務プログラムといった簡単だった構成から、OS、MW、アプリケーションフレームワークといった論理階層が増加している。更に Java などの開発効率向上を目的としたオブジェクト指向言語の発展により、JavaVM 層、J2EE 層といった論理階層が増加している。
- 仮想化(パーチャライゼーション)技術の導入：信頼性向上のためのクラスタリング技術、性能向上のためのロードバランシング技術、リソース利用効率向上のためのパーティショニング技術、等の導入により、各構成要素の多重化が進んでいる。

このように複雑化が進んだシステムにおいて障害が発生した場合、その根本原因/影響範囲を特定することは困難であり、障害対応の遅れの大きな一因となっている。障害対応の遅れは、業務停止による損失だけでなく、企業の信用を失い顧客離れにも繋がる大きな悪影響をもたらす。障害対応を効率的に実施するためには適用されている各種技術を理解して、各構成要素/業務処理の間の関係を正確に把握する必要がある。

2.2. 監視対象のモデル化

2.1.で記したシステムの複雑化に対する対策として監視対象をモデル化する技術が注目されており、次のような効果が期待できる。

- 抽象度の高いモデルを定義することによりシステムに適用されている各種技術に関してベンダ毎の差異を吸収する。
- モデル化することでデータに意味づけが出来るため、情報の流用を容易とし、情報(技術)の経年劣化に対応可能とする。
- 構成要素間の関係情報を表現することにより、複雑なシステム構成が表現可能となる。

運用管理技術の標準化の動向([1] ~ [7])としては、DMTF(Distributed Management Task Force)において策定されている CIM (Common Information Model)を中心とした活動が活発化している(図 2)。

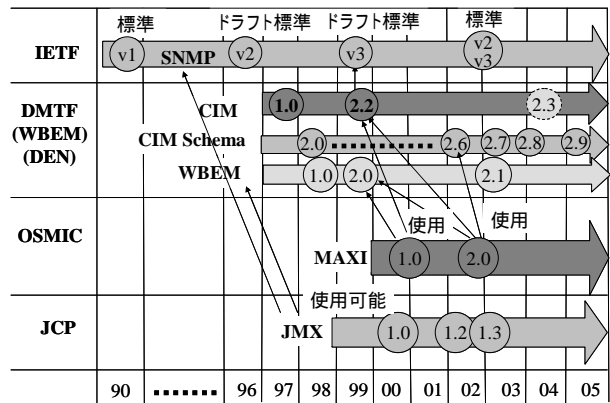


図 2 標準化の動向

CIM[1]では監視対象をモデルとして定義するための枠組みが決められている。また、その他に共通的に利用できるスキーマ(CIM Schema[2])や、CIMのモデル(情報)を汎用的な技術(HTTP/XML など)を用いて操作するための仕組み(WBEM[3])が策定されている。CIM Schema では、汎用的な管理に利用できる抽象度の高い「コアモデル」、各技術領域毎にまとめられた共通性の高い「共通モデル」、各ベンダ毎に詳細な情報が定義可能な「拡張モデル」という 3 レベルのモデル化が行われている(図 3)。レベルに分けたモデル化を行うことにより、例えば計算機の共通的な情報を得ようとする場合、ComputerSystem クラスの情報を取得すれば良く、各ベンダによって拡張されたクラスに関しては考慮する必要が無い。各ベンダにおいて拡張された情報が必要な場合にのみ、各ベンダが拡張したクラスを考慮すれば良いため、管理 S/W の流用が容易となる。

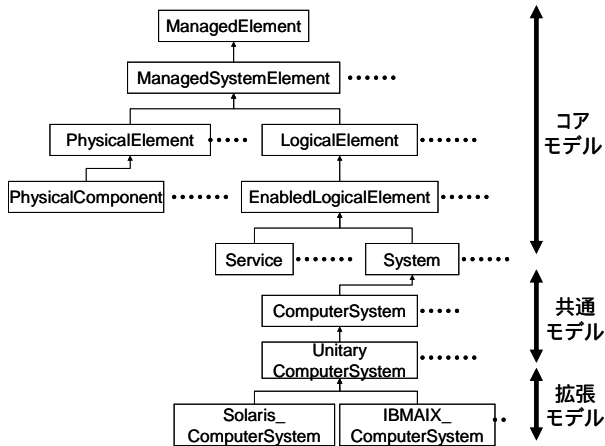


図 3 CIM モデル

2.3. 障害影響分析技術

次に、障害影響を把握するための技術、方式(8)~(11))について簡単にまとめる。障害影響を把握するための技術/方式を大別すると以下の通りである。

・計測技術

- プローブ方式：実際に業務を行っている処理と同等の処理を行うプローブを端末上で動作させることにより、業務が継続的に利用できているかを計測する方式

・分析技術

- 依存関係探索方式：事前に各構成要素間の依存関係の情報を登録しておき、障害発生時に依存関係を探索することにより、障害の影響範囲、および根本原因の分析を行う方式

- 障害 DB 検索方式：事前に過去障害の情報やフィルタリング、相関処理のルールを障害 DB として登録しておき、障害発生時に複数あるイベント(メッセージ)からルールに従って障害 DB を検索し、障害の根本原因、影響範囲を特定する方式

上記各方式のメリット/デメリットをまとめると表 1 ようになる。

表 1 各方式の比較

| 方式 | メリット | デメリット |
|----------|--|---------------------------------------|
| プローブ | 実際の顧客の体験が測定可能なため SLA 違反等の監視が容易 | ・監視トラフィックの増加 ・プローブ設定作業の増加 |
| 依存関係探索 | 簡単なロジックによる根本原因/影響範囲の把握が容易となる | ・依存関係定義作業の増加 ・構成要素/依存関係増加に伴う処理量の増加 |
| 障害 DB 検索 | 複数のイベント(メッセージ)を一度に処理できるため、不要なメッセージを排除し、障害に関する情報のみ抽出できる | ・ルール作成が困難 ・ルールにより誤検出や検出漏れが発生する可能性 |

3. 障害影響分析の課題

3.1. 要件

本節では、本稿で検討する障害影響分析の要件についてまとめる。機能要件は各構成要素内の 1 つまたは複数に障害が発生した場合の業務への影響を分析することであり、特に以下のような点を満たすことが望まれる。

- ・要件 1：クラスタやロードバランサといった技術を導入した高可用システムにおいても正確な影響範囲が分析できること
- ・要件 2：分析では影響のある/なしだけではなく、系切替(フェイルオーバー)により継続可能、冗長度の低下、性能低下の可能性など詳細な影響が分析できること

上記を検討するために、対象システムとして図 4 のような計算機センターと複数の拠点があるような大規模な分散システムを想定する。また、図 5 のように信頼性向上のために、業務サーバは多重系が組みられ、障害発生時には処理を引き継ぐことにより、業務の継続が可能な構成を想定する。

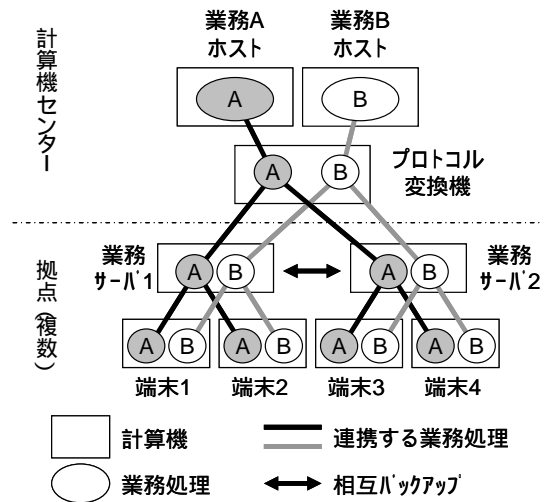


図 4 管理対象の構成例

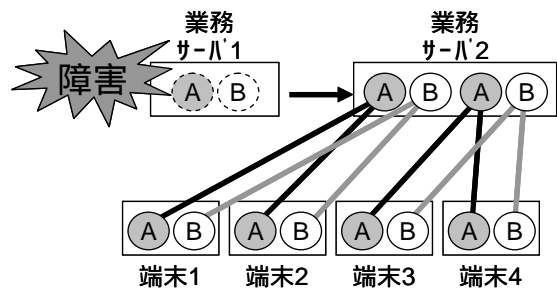


図 5 障害発生時の切替処理

3.2. 共通情報モデル(CIM)の適用

CIMを用いて管理対象システムをモデル化し、関係情報を探索することにより、障害影響の分析が可能を検討する。初めにCIMクラスを用いて、3.1.に記したシステムのモデル化を実施した。各構成要素とCIMクラスの対応を表2にまとめる。

表2 管理対象とCIMクラスの対応

| 管理対象 | CIMクラス |
|--------------|---|
| 計算機 | ComputerSystem |
| 業務処理 | Service |
| 多重システム | Cluster、または RedundancyGroup |
| 計算機と業務処理の関係 | HostedService |
| 業務処理間関係 | ServiceServiceDependency |
| 多重システムと計算機関係 | ParticipatingCS、または RedundancyComponent |

作成したモデルには複数の拠点があり、各拠点での多重システムの構成が異なるが、本稿で全体像を説明するには紙面が足りないため、1部分の構成を例として説明する。なお、以降の図中において関係は表3のように表記する。

表3 関係の表記

| | 本稿 | CIM | UML(参考) |
|-----------------------|----|---------|---------|
| 継承関係(Inheritance) | ← | ← (青) | ← |
| 関連(Association) | — | — (赤) | — |
| 従属(集約)関係(Aggregation) | ◊ | ◊ (緑) | ◊ |
| 依存関係(Dependency) | ← | 関連と同一表記 | ← |

● 端末1と業務Aホスト間の依存関係の構成例

「各計算機(ComputerSystem)」と「計算機上で動作している業務処理(Service)」を「依存関係(HostedService)」で対応付け、「業務処理(Service)」と「その業務処理が停止すると機能しなくなる業務処理(Service)」の間を「依存関係(ServiceServiceDependency)」で対応付ける(図6)。

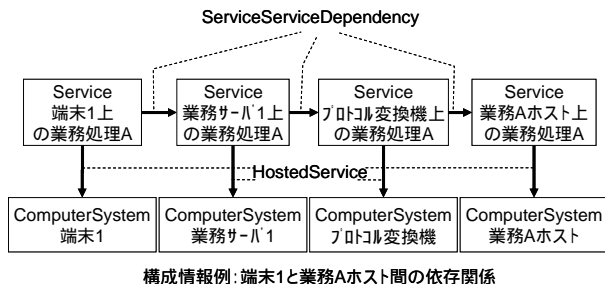


図6 計算機と業務処理の構成例

● 業務サーバの2重システムの構成例

「クラスタ(Cluster)」と「クラスタ上で動作する業務処理(Service)」を「依存関係(HostedService)」で対応付け、「クラスタ(Cluster)」と「クラスタを構成する計算機(ComputerSystem)」を「依存関係(ParticipatingCS)」で関連付ける(図7(a))。または、冗長構成(RedundancyGroup)を用いて、「冗長構成」と「構成する計算機(ComputerSystem)」を「従属関係(RedundancyComponent)」で関連付ける(図7(b))。

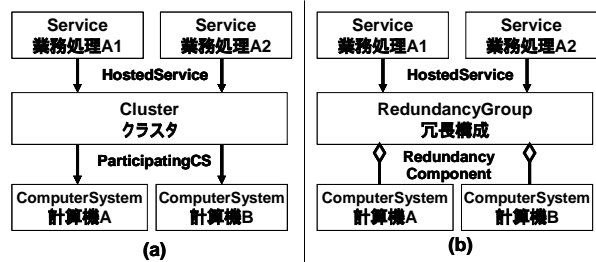


図7 2重システムの構成例

3.3. 課題

3.2.にてCIMを用いてモデル化した監視対象システムの構成情報を利用した障害影響範囲分析について検討する。すなわち、要件1を満たすために、ある構成要素xに障害が発生した場合、構成要素xから関係情報を探索した結果得られた集合をA(x)とし、実際にオペレータが業務yを行っている構成要素(業務処理)の集合をB(y)とした場合に、「A(x) B(y) = 」ならば「業務(y)に影響無し」、「A(x) B(y) = 」ならば「業務(y)に影響有り」として分析できるかを検討する。また、要件2を満たすために、影響の有る/無いのみでなく詳細な影響を分析するための方式について検討する。

ここで、多重系(本例では2重系)システム内の1台の計算機に障害が発生した場合を仮定する。クラスタ(Cluster)クラスを用いてCIMの依存関係(Dependencyクラス、およびそのサブクラス)を探索した場合や、冗長構成(RedundancyGroup)クラスを用いて依存関係と従属関係を探索した場合に得られた集合を影響範囲とすると、実際には系切り替えにより、業務が継続可能なものに影響がある分析結果となる。また、冗長構成クラスを用いた場合で、依存関係のみを探索した場合は計算機の障害による影響が無いという解析結果となる。例えば、3.2.でモデル化したシステムにおいて業務サーバ1に障害が発生した場合、図8のようになる。ここで、構成要素xの障害により、実際に影響のある構成要素の集合を $A_R(x)$ 、依存関係のみを探索して得られた集合を $A_D(x)$ 、依存関係と従属関係を探索して得られた集合

を $A_{D,C}(x)$ とすると、 $A_{D,C}(x) \supseteq A_R(x) \supseteq A_D(x)$ であり、 $A_R(x) \supseteq A_D(x)$ かつ $A_R(x) \supseteq A_{D,C}(x)$ のため、正確な影響範囲を分析できないことがわかる。

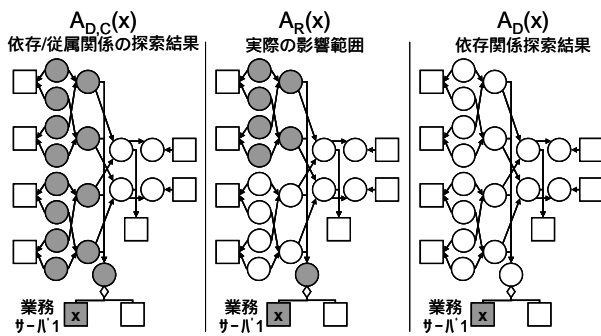
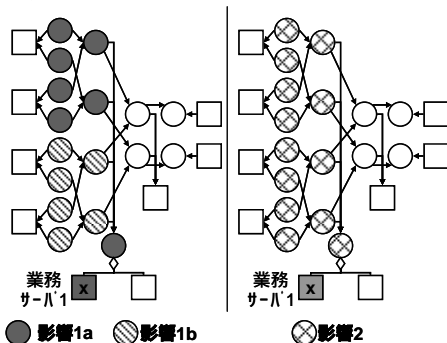


図 8 分析結果例

次に、各構成要素の詳細な影響を実際のシステムを考慮して検討する。業務サーバ1に障害が発生した場合、各構成要素に対して図9のような影響があると考えられる。しかし、多重系システム上で動作する業務処理は同等な関係としてモデル化されるため、図9のような詳細な影響を分析することは不可能である。



影響 1a: 障害発生時に系切替時間の業務停止
 影響 1b: 待機系停止による信頼性(冗長度)低下
 影響 2: 対応端末増加による性能低下の可能性
 および待機系停止による信頼性低下

図 9 詳細な影響

以上で示した課題をまとめると以下の通りである。

- 課題 1: 単純に関係情報を探索するのみでは正確な影響範囲を特定できない。
- 課題 2: 単純に発生した障害を基に探索するのみでは詳細な影響を把握できない。

4. 解決策

4.1. モデル拡張

3.3. で記した課題を解決するために次のような対策を提案する。

- ある条件が成り立った場合に依存関係を表わす条件付依存関係を導入する。
- 多重系システム(Cluster)クラスを、従属する構成要素に条件付依存関係があった場合に、各構成要素の状態(OperationalStatus 属性)を再評価できるように拡張する。

次に対策の詳細について示す。図10のような構成が表現可能なように多重系システム(Cluster)クラスを拡張(MEL_ClusterSystem クラスとして定義)し、条件付依存関係(サービスの実行可能な計算機を表わす関係クラスとして MEL_ServiceRunnableCS クラスを定義)、および従属関係(多重系システムを構成する計算機を表わす関係クラスとして MEL_ClusterComponentCS クラスを定義)で関係を表わす。

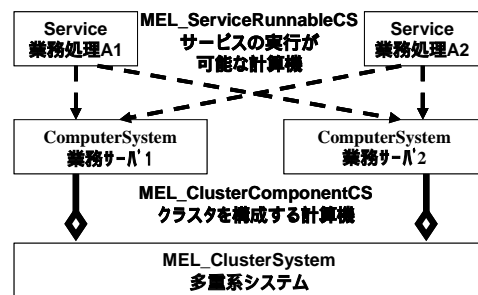


図 10 拡張した多重系システムの構成例

4.2. 分析アルゴリズム

障害の業務への影響を分析するアルゴリズムを以下の通りとする(図11)。

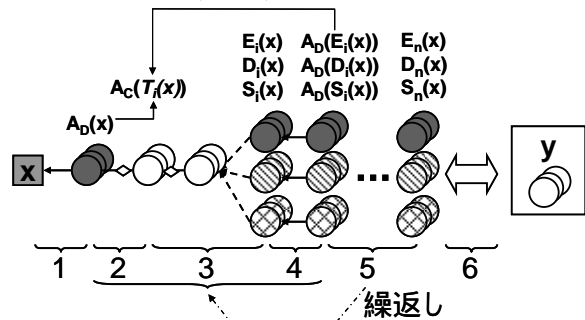


図 11 分析アルゴリズム

1. 障害が発生した構成要素 x の依存関係を探索し、影響のある構成要素の集合 $A_D(x)$ を得る。
2. 得られた構成要素の集合 $T_i(x)$ から従属関係が結ばれている構成要素の集合 $A_C(T_i(x))$ を得る。なお、 $i=1$ のとき $T_i(x) = A_D(x)$ である。
3. 得られた集合内の各構成要素 $A_C(T_i(x))$ と従属関係にある多重系システムクラスの情報を基に多重系システムクラスに従属する計算機と条件付依存関係にある構成要素の状態を再評価し、障害

が発生(状態がError)する構成要素集合を $E_i(x)$ 、信頼性(多重度)が低下する(状態がDegraded)構成要素集合を $D_i(x)$ 、性能低下の可能性のある(状態がStressed)構成要素集合を $S_i(x)$ と表わす。

4. 各状態の構成要素毎に依存関係を探索し、影響のある構成要素の集合を得る。
 - 障害が発生構成要素： $E_{i+1}(x) = A_D(E_i(x))$
 - 信頼性が低下する構成要素： $D_{i+1}(x) = A_D(D_i(x))$
 - 性能が低下する構成要素： $S_{i+1}(x) = A_D(S_i(x))$
5. 2～4 の処理を $E_{i+1}(x)$, $D_{i+1}(x)$, $S_{i+1}(x)$ を基として各構成要素の状態が変化しない状態になるまで繰り返し、 $E_n(x)$, $D_n(x)$, $S_n(x)$ を得る。
6. 業務 y に与える詳細な影響はそれぞれ次のように求めることが出来る。
 - 障害が発生する構成要素： $E_n(x) \ B(y)$
 - 信頼性が低下する構成要素： $D_n(x) \ B(y)$
 - 性能が低下する構成要素： $S_n(x) \ B(y)$

4.3. プロトタイプシステム

4.1 .にて提案したモデルの拡張により、詳細な障害影響分析が可能なことを確認するためにプロトタイプシステムを作成した。作成したプロトタイプシステムの構成は図 12の通りである。

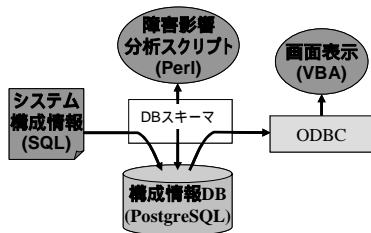


図 12 プロトタイプシステム構成

作成したプロトタイプシステムにより、4.1.で提案したモデルの拡張により正確な、および詳細な障害影響の分析が可能なことを確認した(図 13)。

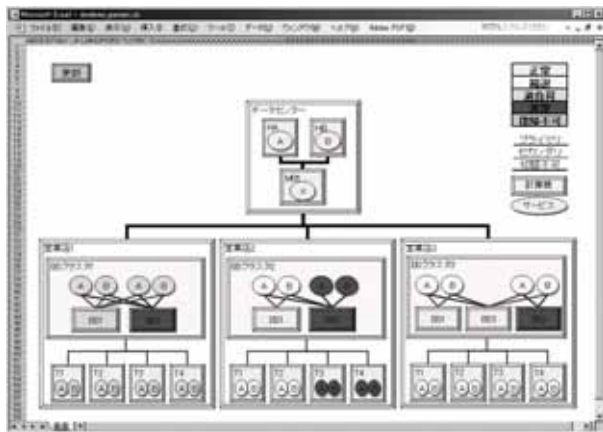


図 13 表示画面例

5. おわりに

本稿では、運用管理システムにおける障害影響範囲分析を実現するにあたり、管理対象システムをCIM を用いてモデル化し、関係情報を探索する方式を検討した。また、多重系システムなどの高可用システムにおいても、正確な、および詳細な障害影響範囲分析を実現するための拡張について提案し、プロトタイプシステムを作成することにより確認した。

しかし、現在のプロトタイプシステムでは構成情報を手入力で行っているため、実システムへの適用を考慮した場合、これらの構成要素情報、関係情報の(半)自動収集を行う仕組みが必要である。今後対応するベンダが増えることにより、自動収集可能な情報が増えていくことが予想される。このとき、収集可能な情報から本稿で提案したモデルへのマッピングが必要になると考えられる。

今後は、対応モデル(技術)の拡充、分析アルゴリズムの改善などについて検討していく予定である。

参考文献

- 1) DMTF : Common Information Model(CIM) Specification Version 2.2 (1999).
- 2) DMTF : CIM Schema v2.8 (2004).
- 3) DMTF : Web-Based Enterprise Management (WBEM) Initiative, <http://www.dmtf.org/standards/wbem>.
- 4) DMTF : CIM Tutorial, <http://www.wbemsolutions.com/tutorials/CIM/>.
- 5) OSMIC : MAXI (MAnagement information eXchange over Internet) Ver 2.0 仕様書, http://www.net.intap.or.jp/INTAP/osmic/business/mxi_2_0.pdf.
- 6) OSMIC : システム運用管理の相互接続に対する OSMICの取り組み(2004), <http://www.net.intap.or.jp/INTAP/osmic/archives/20040616data/wakasugi.pdf>.
- 7) JSP : Java Management Extensions(JMX™) v1.2 Specification(2002).
- 8) 相浦利治, ほか : 運用管理の高度化,2004 信学会総全大, A-9-3, 電子情報通信学会(2004).
- 9) 宮内直人, ほか : ネットワーク障害の故障診断方式に関する検討, TM 研究会, 電子情報通信学会(2004).
- 10) 森一, ほか : サーバ依存関係を利用したシステム構成管理の支援方法, DSM 研究会, 2003-DSM-31, 情報処理学会(2004).
- 11) NRI 技術創発 Vol.3 P78 「CIM 知識ベースを活用したシステム運用管理のインテリジェント化」(2004).