RO-011

オープン・システム障害の分類と対策立案方法の提案

Discussion of how to categorize a system trouble and a proposal how to provide an optimized action plan

篠原昭夫[†1]　泉隆[†2]

Discussion of how to categorize the open system trouble. We first define the group of trouble. It is a preparation for using the function chain method. The next step divides the types of trouble by function chain. Finally we propose the individual action plan for each type of trouble. This enables to avoid a confusion that we often see when we select the action plan. It also provides a way to eliminate the existence of secondary problem.

1. はじめに*

オープン・システムで発生した障害の原因調査と対策実施は，ユーザ毎に行われているのが実情である．またその障害記録や調査結果は外部に公開されることが殆どなく，障害事例をまとめて論理的に分析する試みは行われていない．さらに障害対策実施にあたっては，二次障害が発生する例が多く見られる．しかし，これらの原因究明もなされていない．

一方で多くの障害事例を俯瞰すると，障害はいくつかの型に分類できることに気付く．さらに分類された障害型に最適な対策方法を関連付けることで，二次障害の発生を防ぐことができると考えられる．

以上の理由からオープン・システムの障害を論理的に分析し，これに適した対策方法を関連付けるための手法が必要である．同時に二次障害の原因を究明することと，これへの対処方法が求められている．

本報告では階層を利用してはじめに障害を分類し，これを論理的に分析するための準備をする．この分類により分析対象を，障害発生部位が一か所である障害に絞り込む．その後に，機能線[1]を利用してここから障害型を抽出する．

つぎに抽出した各障害型に対し，最適な対策方法を関連付ける．ここで提案する対策方法は，二次障害の発生に対処することを考慮したものである．

以上により，障害発生から対策実施の完了までを，分岐のない流れとすることができる．この定形化により不適切な対策方法が選択されることが回避可能となる．また提案する対策方法では，二次障害発生の防止とその発生検出を容易にすることが考慮されている．

2. 現状における課題と対象システム

2.1 現状における課題

オープン・システムでは障害原因調査と対策実施はユーザ毎に行われている．このため障害事例を統一した視点で論理的に分析する試みはなされていない．また対策実施にあたり DOA [a]などの問題が発生し，対策作業自体が完遂困難となる事例が多く見受けられる．DOA は二次障害の発生原因である割合が高く，この防止と検出を容易にすることが求められている．さらに発生した障害には不適切な対策が立案される事例が散見され，このことも二次障害の発生原因となっている．

このような理由から，障害を論理的に分析してこれに最適な対策方法を関連付ける方法が必要である．さらにこの対策方法は，二次障害発生の防止と対処が考慮されたものでなければならない．

2.2 対象とするシステム規模

オープン・アーキテクチャ製品で構成されたシステムの多くは中・小規模なものである．小規模システムはサーバが１台から数台程度，中規模はサーバが数台から数十台程度のものを指すとする．本報告ではこれら中・小規模のシステムを対象とする．これは，障害を論理的に分析する際に利用する機能線が対象としている規模と同じである．

3. 機能線の概要

機能線は障害発生部位を特定するために考案された手法であるが，障害を統一した形式で論理的に記述できる．このため,本報告ではこれを障害分析の手段として利用する．本章では機能線の概要を説明し，次章以降の理解の一助とする．

[定義]　障害をシステム内のある二点間で目的データ転送中に問題が発生したものと捉え，データの流れに沿い配置されたコンポーネントを結んだグラフ構造を「機能線」と定義する[1].

機能線は以下の原則に従って記述する．

１．分岐のない１本の線である

２．始点は目的データが格納された場所である

３．終点は目的データを要求した場所である

障害は機能線上における目的データ転送の失敗であると考える．つまり障害とは目的データの転送が正しく行われなかった，またはそれが期待する場所に存在していなかったために発生したと解釈できる．この考えに基づくと機能線上には障害発生原因となった部位が必ず存在する．１つ

* †1 日本大学　Nihon University
†2 日本大学　Nihon University

a) DOA (Death on Arrive):
交換作業で投入した新品部品が問題を持っていること．

の機能線上に複数の障害発生部位が存在するのは，複数の障害が同時に発生している場合であり通常は一か所である．

図 1 の中で発動点とは，最初に障害発生を報告して来た部位である．殆どのユーザはここからのエラー報告により障害発生を認識する．発動点は障害発生部位と必ずしも一致しているとは限らない．

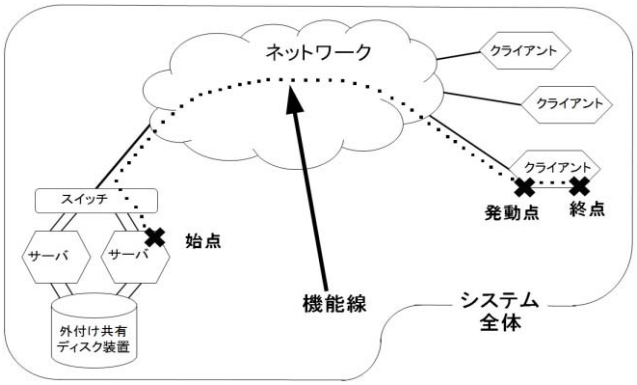

図 1　システム全体と機能線の関係

3.1 機能線とコンポーネント

機能線上にはコンポーネントが配置される．コンポーネントとはシステムを構成している要素で，機能線上で目的とするデータ転送に関与するものである．

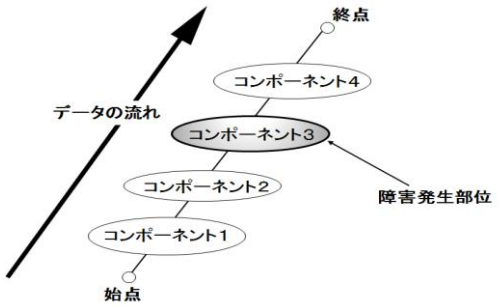

図 2　機能線とコンポーネント

図 2 はコンポーネント 4 個からなる機能線を視覚的に表記したものである．コンポーネントの例をあげる．

・ネットワーク・カード（ハードウェア）
・デバイス・ドライバ（ソフトウェア)
・SCSI ケーブル（ハードウェア）
・ディスク・アレイ装置（ハードウェア）
・ftp クライアント・ソフト（ソフトウェア）

これらの例からわかるようにコンポーネントはシステムを構成している要素（製品）そのものである．機能線ではこれらをハードウェアとソフトウェアの違い，製品規模の大小に関係なく全て一つのコンポーネントとして扱う．

3.2 コンポーネントの分解

コンポーネントの規模は大小様々であるため，実際の障害調査では障害発生部位であるコンポーネントが判明しても対策立案には不十分であることがある．そこでコンポーネントをより詳細に分解し，さらにその中でどのコンポーネントが被疑部位であるかを絞り込む必要がある．

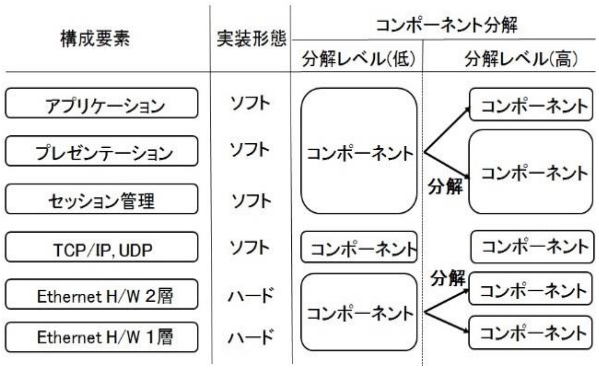

図 3　コンポーネントと分解レベル

図 3 はコンポーネントと分解レベルの関係を示したものである．分解レベルが低い場合は 3 個の，より高めた場合には 5 個のコンポーネントに分解されている．

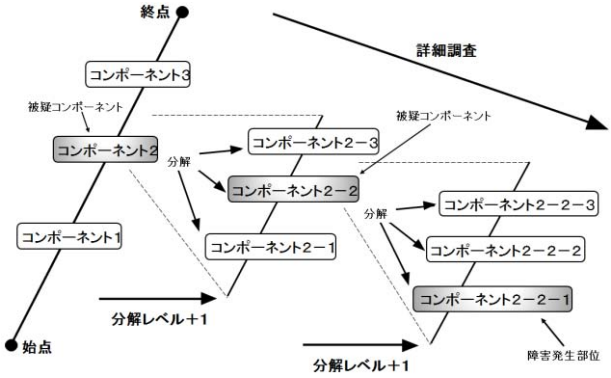

図 4　障害発生部位が判明した機能線

図 4 に障害発生部位のコンポーネントが判明した機能線を示す．分解レベルを用いて効率的に障害発生部位を判定している様子がわかる．

4. 障害分析のための準備

障害を論理的に分析するための準備として，分類階層を導入する．これにより障害を三つの階層に分類する．

階層 A: 技術的，非技術的な視点での分類
階層 B: 障害発生部位の個数による分類
階層 C: 機能線を利用した障害型による分類

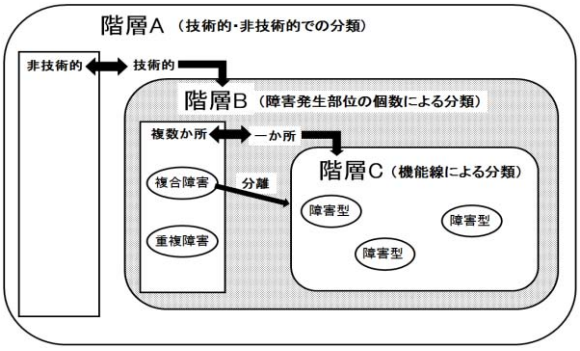

図 5　階層による障害の分類

階層 A から C は，A⊃B⊃C の関係にある（図 5）．障害発生部位が複数ある場合，対策立案には各障害固有の条件を加味する必要があることが多い．これを避けるため，階

層 B により障害箇所が複数あるものを分離する．

4.1 階層 A: 非技術的な障害の除外

図 5 の階層 A では，障害が技術的または非技術的であるかで分類する．分類の基準は次の(1)から(3)である．

(1) 必ずしも製品の不具合や故障などが発生してないが，システム利用者が問題と指摘している事象
（例）パフォーマンス問題

(2) 作業誤りによって発生した事象

(3) 製品の不具合や故障などの問題が表面化している事象

(1)には利用者の主観が，(2)は人為的な要因が含まれるため，非技術的な障害に分類する．残る(3)は技術的要因による事象である．よって次の階層 B では(3)を対象にする．

4.2 階層 B： 障害発生部位の個数による分類

図 5 の階層 B では，一つの障害に原因部位が複数あるかにより分類する．この状態は次の(1)と(2)が考えられる．

(1) 複合障害

複合障害とは，互いに依存関係にない複数の障害が，システム内で同時または同じ期間に発生した状態をいう(図 6)．この障害は独立した複数の障害に分離できる（図 6 の障害 1 と障害 2）．したがって，障害発生部位が一か所である複数の障害として扱うことができる．

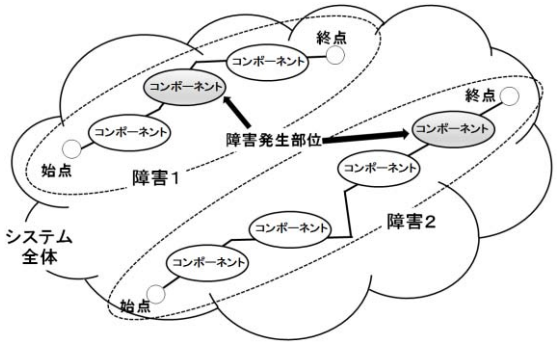

図 6　複合障害

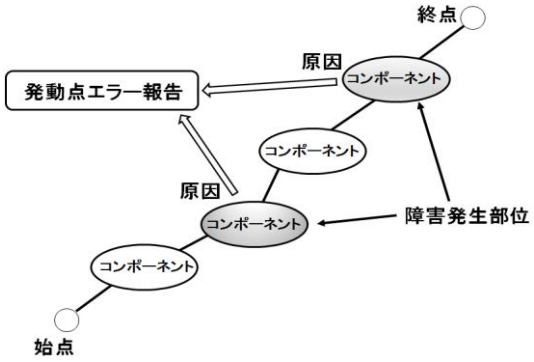

図 7　重複障害

(2) 重複障害

重複障害とは，機能線の発動点エラー報告の原因となった障害発生コンポーネントが，複数存在する状態をいう(図 7)．これは複数のコンポーネントで独立して障害が発生し，かつそれらが同じ発動点エラー報告の原因となったことを意味する．したがって発生確率は極めて低と考えられるため，本報告では考慮しない．

4.3 階層 C: 障害発生部位が一か所の障害

図 5 の階層 A と B の二層で分類され，残った階層 C に属する障害は，障害発生部位が一か所のものである．実際に発生する障害の殆どはこの階層に属する．

障害発生部位が一か所に絞られたことにより，異なる障害を統一された視点で分類可能となる．

5. 機能線を用いた障害分析

本章では機能線を利用することで，階層 C に属する障害を 3 種類の型に分類する[2]．提案する障害型は筆者らが障害事例を分析することで抽出したものである．なお、現在までにこれ以外の型の存在は確認されていない．

[抽出された障害型]

・通常型障害
・両端決定不可型障害
・タイムアウト階層違反型障害

5.1 通常型障害

図 8 に通常型障害を示す．この型では障害発生部位が唯一に判定できており，最も一般的な形態である．

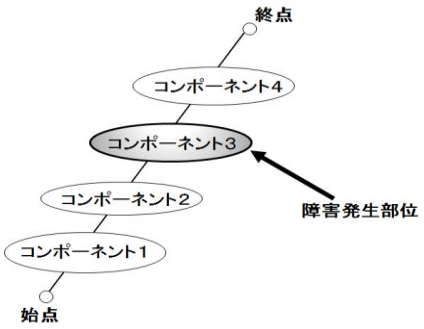

図 8　通常型障害

図 9 に通常型障害の実例として，ディスク・ドライブに読み出し不可ブロックがある障害をあげる．この型では、たとえ障害発生コンポーネントの分解レベルを上げたとしても，検出される障害発生部位は一か所のみであることが重要である．

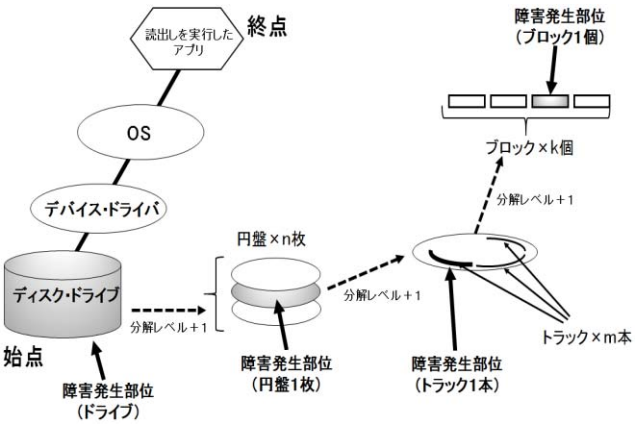

図 9　通常型障害の実例

5.2 両端決定不可型障害

図 10 に両端決定不可型障害を示す. この型では被疑コンポーネントが隣接する 2 個まで絞り込まれたが，そのどちらで障害が発生したかを判定できない．図 10 を用いれば，コンポーネント 3-1 と 3-2 はともに被疑部位であるが，どちらが障害発生部位かを判定できない．

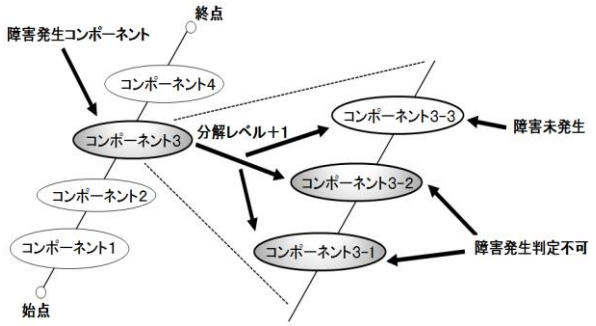

図 10　両端決定不可型障害

図 10 からわかるように, この型は分解レベルが低い場合には通常型障害である．しかしコンポーネント 3 の分解レベルを上げたために，両端決定不可型障害となる場合があることに注意を要する．通常型障害ではこの状況は起こらず，分解レベルを上げた後でも，容易に障害発生部位が判定できる．

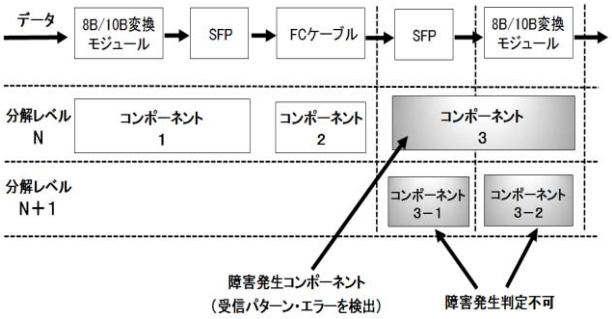

図 11　両端決定不可型障害の実例

図 11 に両端決定不可型障害の実例として，Fibre Channel 接続の光転送で 8B/10B 変換[3][4][5]の受信パターン・エラーが検出された障害をあげる．図中で分解レベル N の状態ではコンポーネント 3 が障害発生部位で，これは通常型障害である．しかしコンポーネント 3 の分解レベルを上げると SFP [b](コンポーネント 3-1)はログ機能がないため，障害原因が SFP の出力部エラーであるか，8B/10B 変換モジュール（コンポーネント 3-2）の受信部エラーであるかを判定できなくなる．すなわち両端決定不可型障害は，製品仕様の制約から障害発生部位が判定できない状態であるといえる．

5.3 タイムアウト階層違反型障害

図 12 にタイムアウト階層違反型障害を示す. この型では，あるコンポーネントが隣接コンポーネントで障害が発生したと記録しているにも関わらず，当該のコンポーネントでは障害発生記録が見られない(図 12 の中のコンポーネント 2 と 3)．この原因は 2 つ考えられる．

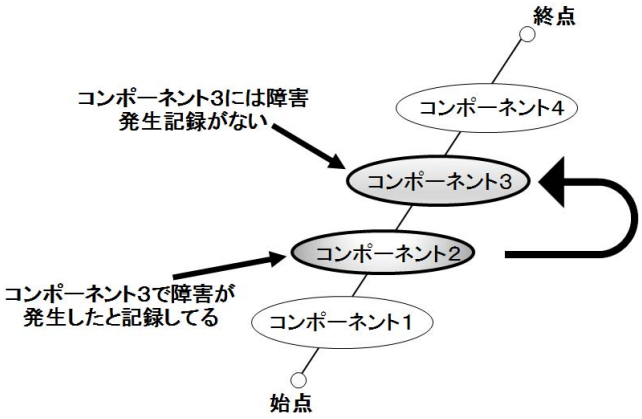

図 12　タイムアウト階層違反型障害

原因 1:コンポーネント 3 の記録が誤っている

この場合は，誤った記録をしているコンポーネント 3 が障害発生部位である. これは通常型障害に分類できるため，以降は考慮しない．

原因 2:コンポーネント 2 が障害記録を生成している

この場合は，障害記録を生成したコンポーネント 2 に論理的不具合があり，製品やシステムの設計不良が原因である．この状態は通常型障害に分類することもできる．しかし，障害発生コンポーネントの判定に際して通常型障害とは著しい違いがあるため別な型とした．なぜなら，通常型障害であればコンポーネント 3 が障害発生部位となるからである．

なおこの型の障害は，殆どの場合タイムアウトに起因して発生するためこの名称とした．

タイムアウト階層違反型障害は，オープン・システムの特徴と考えられる．これは，システムを構成するコンポーネントが異なるベンダから提供されたために，それらの連動確認が不十分であることが原因で発生したからである．

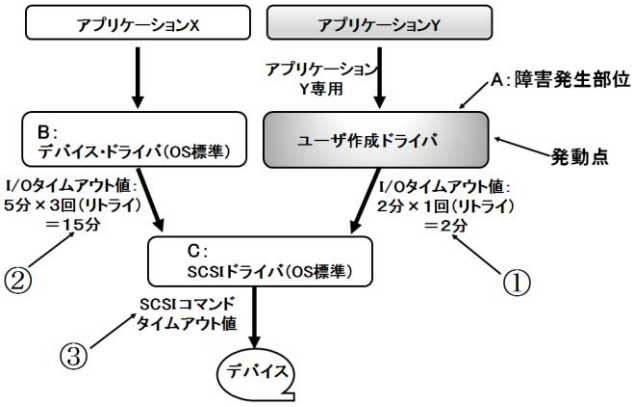

図 13　タイムアウト階層違反型障害の実例

図 13 にタイムアウト階層違反型障害の実例をあげる. 図中で B(デバイス・ドライバ(OS 標準))の I/O タイムアウト値 15 分に対し，A(ユーザ作成ドライバ)での値は 2 分と短

b) SFP: Small Form Factor Pluggable
(電気信号と光信号の変換モジュール)

い．しかし A はこの下位層で C(SCSI ドライバ(OS 標準))を流用するため，ここでタイムアウト判定値の逆転状態が生じる(①＜③)．すなわち上位層の A で検出されたタイムアウトが，下位層の C ではタイムアウトでない状態となる．これは①の値が③の値を考慮せずに，独自に決定されているためである．

この型の障害では，機能線での発動点が階層違反を起こしたコンポーネントと一致することに注意を要する（図 13 で A は発動点かつ障害発生部位である）．

タイムアウトを検出する仕組みを持つコンポーネントは，検出後の処理と障害調査を考慮して設計されるべきである．また，この型の障害はタイムアウト値の設定誤りによっても起こるため，製品自体が不具合を持つ割合よりも多く発生している．

6. 二次障害の発生原因

オープン・システムでは障害対策実施中にしばしば二次障害が発生し，このために対策作業自体が完遂できなくなる事例が見受けられる．二次障害が発生する原因は次の 4 つが考えられる．

(1) 障害対策で変更する部位が他に及ぼす影響の考慮不足によるもの
(2) 作業誤りによるもの
(3) DOA によるもの
(4) 不適切な対策方法の選択によるもの

(1)，(2)は共に人為的要因であると考えられるため，ここでは触れない．

6.1 DOA による二次障害

(3)の DOA（Death on Arrival）とは，ハードウェア障害で交換した新品部品に問題があることをいう．発生した問題が交換前と同じである場合には，交換箇所が障害発生部位でないと誤判断する原因となる．発生した問題が交換前と異なる場合には，結果判断はさらに困難となる．DOA は二次障害の発生原因である割合が高く，この防止と検出を容易にすることが求められている．

DOA はハードウェアに関連した用語である．しかし機能線を用いた障害原因調査では，ハードウェアとソフトウェアの違いを意識する必要がない．ここで，ソフトウェアの DOA を定義する．

[定義 1] 「ソフトウェア DOA」とは，障害対策で変更したソフトウェアで変更前と同じ問題が発生する，またはあらたな問題が発生することをいう．

この定義は，ハードウェアの DOA と同じである．したがって，DOA の定義はソフトウェアまで含むものとする．

[定義 2] 「DOA」とは，ハードウェアとソフトウェアの違いに関係なく，障害対策による変更後のコンポーネントで問題が発生することをいう．

本報告では，以降 DOA には定義 2 を用いる．

6.2 不適切な対策方法の選択による二次障害

ここでは，DOA 検出が増加する傾向が見られる原因について述べる．近年では活性交換[c]可能なハードウェア部品が増加し，このことが DOA の誤検出率を高める原因となっている．これは，活性交換を自動検出する部位に不具合が内在した製品が多いためである．

6.3 不適切な対策方法の選択による二次障害

(4)の不適切な対策方法が選択される原因は，機能線でのコンポーネント分解が正しくないためである．図 11 を再び用いる．分解レベルが N の状態で調査を終了すると，通常型障害と判定される．ここで対策としてコンポーネント 3-2 のみを変更すると，DOA の誤検出などの二次障害を招く．この例では，コンポーネント 3 を変更するのが正しく，分解レベル N+1 で現れるコンポーネント 3-2 のみを変更することは不適切な対策方法となる．

両端決定不可型障害と DOA は，オープン・システムの障害対策を困難にしている．

7. 各障害型の対策方法

障害が属する型を判別した後に，その障害型に最適な対策方法を割り当てる手法を提案する[2]．障害型と対策方法を関連付けることで，不適切な対策方法が選択されることを防止できる．なお，提案する対策方法では DOA の発生防止と，その検出を容易にすることが考慮されている．

7.1 通常型障害の対策方法

通常型障害では，障害発生コンポーネントのみを変更する．変更するコンポーネントが周辺に及ぼす影響以外は考慮する必要はない．対策実施後に同じ問題が発生，またはあらたな問題が発生した場合には DOA と判定する．

7.2 両端決定不可型障害の対策方法

両端決定不可型障害の対策方法は次の 2 通りがある．

(1) 同時投入法

この対策方法では，被疑コンポーネントの両方を同時に変更する．どちらのコンポーネントが障害発生部位かを判定することはできないが，一回の作業で対策を完了できる利点がある．

(2) 順次投入法

この対策方法では，被疑コンポーネント一方のみを変更したのちに経過観察を行う．この方法は対策完了までに時間を要するが，障害発生部位を判定できる利点がある．

[1 手順-1 変更の原則]

同時投入法と順次投入法のいずれの場合にも，一度の対策作業では一か所だけを変更する(1 手順-1 変更の原則)．同時投入法を例とすれば，被疑コンポーネントの両方（図 11 のコンポーネント 3-1 と 3-2）を，常に一体として扱う．この原則は DOA の検出を容易にする効果がある．

c) 活性交換(Online Replacing): 装置稼働中に部品を交換すること

7.3 タイムアウト階層違反型障害の対策方法

タイムアウト階層違反型障害の対策方法は次の２通りがある．

(1) タイムアウト階層違反をしているコンポーネントを変更する（図 12 のコンポーネント 2）．すなわち，タイムアウトを生成している論理不良を修正する．

(2) タイムアウト階層違反をしているコンポーネントが，エラー発生源と記録している隣接コンポーネントを変更する（図 12 のコンポーネント 3）．これは問題のないコンポーネントを変更することとなるが，実際の障害事例ではコンポーネント変更に制約がありこの方法が選択される例が見受けられる．

(1)，(2)のどちらの場合も，対策実施後の DOA 判定方法は通常型障害と同様である．

8. 他の障害分類事例

システム障害を分類する試みには[6][7][8]などが見られる．しかし[6]での分類はシステム管理、ユーザ影響、などの視点のみで行われており、障害自体の論理的構造には言及していない。また[7]では障害事例の原因について技術的な部分にまで触れた分析を行っているが、異なる障害同士の論理的構造を直接比較するまでには至っていない。一方[8]では特定の製品で発生する可能性のある障害を予測し，これの分類を試みている．しかし対策法への関連付けは試みられていない。

障害事例をユーザが外部に公開することは稀であり，障害事例を論理的に分析することを困難にしている．このような理由から，抽出された障害型に対策方法を関連付ける試みは他では見あたらない．

9. まとめ

本報告では階層を用いてオープン・システムで発生した障害を分類する方法を提案した．ここでは障害が技術的，非技術的であるかどうか，検出された障害発生個所が複数個であるかどうかの二段の階層で分類を行った．これにより障害発生部位が一か所である障害に絞り込みを行った後に，機能線を利用して障害型を抽出した．障害型は今後の調査で新しいものが見いだされる可能性がある．

つぎに各障害型に最適な対策方法を関連付ける方法を提案した．これにより，オープン・システムの障害対策実施中に発生する二次障害の一つである，不適切な対策方法が選択されることを防止できる．さらに両端決定不可型障害と DOA の関係についても述べ，DOA が二次障害の主な原因であることを究明した．

本報告の成果をまとめると次のようになる．

- オープン・システムの障害事例を論理的に分析した
- 障害発生部位が一か所である障害事例を分析し，ここから共通の障害型を抽出した
- 不適切な対策方法が選択される原因を究明した
- 各障害型に最適の対策方法を関連付けることで，不適切な対策方法が選択されることを防止可能とした
- 二次障害の主な原因が DOA であることを究明した
- DOA の定義をハードウェアから，ソフトウェアにまで拡張した
- 提案した対策方法では DOA の発生防止と，その検出を容易にすることが考慮されている

オープン・システムの障害対策方法は，修正プログラムの投入決定指針[9]までを考慮しながら，今後も研究を進めてゆく必要があると考える．

参考文献

1) 篠原昭夫，泉隆：「オープン・システム上での障害発生部位特定方法の提案」，情報処理学会第 75 回全国大会，5A-2 （2013-03）

2) 篠原昭夫，泉隆：「システム障害発生部位判定方法と障害の分類」，第 12 回情報技術フォーラム，B007（2013-09）

3) Robert W. Kembel:
Fibre Channel A Comprehensive Introduction, Northwest Leading Associates, Inc, chapter10, (2001)

4) 8B/10B encoding:
http://www.snia-j.org/dictionary/storage_network_keywords/2.html, (2013)

5) Byte oriented DC balanced (0,4) 8B/10B partitioned block transmission code:
http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.htm&r=1&f=G&l=50&s1=4,486,739.PN.&OS=PN/4,486,739&RS=PN/4,486,739, (1982-06)

6) www.ipa.go.jp/files/000026797.pdf
海外における IT 障害の影響及び 対応策に関する事例調査 - 報告書，添付資料１ ：障害事例集 pp.1-29, (2013-04)

7) www.ipa.go.jp/files/000038843.pdf
http://www.ipa.go.jp/sec/reports/20140513.html
重要インフラ障害情報の分析に基づく「情報処理システム高信頼化教訓集（IT サービス編）」 ～障害の再発防止のため、業界を越えて幅広く障害情報と対策を共有する仕組みの構築に向けて～ pp.I-18-39, (2014-05)

8) 8.2. Failure classification:
https://access.redhat.com/documentation/en-US/JBoss_Enterprise_SOA_Platform/4.2/html/SOA_ESB_Programmers_Guide/SOA_ESB_Programmers_Guide-_Fault_tolerance_and_Reliability_-_Failure_classification_.html

9) 篠原昭夫，泉隆：「オープン系システム保守の現状報告」，情報処理学会第 76 回全国大会，4ZE-4 （2014-03）