RO-010

オープン・システム上での障害発生部位特定方法の提案

Discussion of how to reach to a root failure point in open systems problem

篠原 昭夫†
Akio Shinohara
日本大学

泉 隆‡
Takashi Izumi
日本大学

An introduction for how to create a function chain, and how to reach to a root failure component. Using a function chain method enables to draw down a problem without considering the difference of hardware and software. Also it provides a thinking for logical structure of the function chain.

１．はじめに

オープン・アーキテクチャ製品で構成されたシステムで発生した障害の原因部位特定方法として機能線を提案する。提案する機能線はデータの流れに沿って配置されるコンポーネントを結んだグラフ構造であり，障害をシステム内のある二点間でデータの転送に問題が発生したと捉えることに基づくものである。すなわち機能線上にシステムを構成するコンポーネントを配置する図表現により，視覚的に障害を記述し障害調査の対象，方向性を明確にすることを可能とする。また機能線は，その分解レベルを順次上げてゆくことで調査をより詳細かつ絞り込んだ段階へと効率的に運ぶことを可能とする。本報告では提案する機能線の考え方，作成方法などについて述べる。

次に作成した機能線を用いて各コンポーネント内で問題が発生したかを判定する方法として比較法を提案する。オープン・システム向けの製品は多機能化が進んだため，同一の製品を選択していてもシステムが異なればその使用法が大きく異なる。このため他システムでの障害調査事例を別システムの障害調査に用いるのが困難なことがある。比較法はこの状況下で各システム固有の使用条件を反映しながら障害原因究明を行うための手法である。

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・システムは様々なメーカーの製品を統合した形態のため，障害発生時に原因部位を特定することが困難なことが多い。さらに個々の製品は商品寿命が短く，未知障害の発生率が低下しない傾向が続いている。このような場合の障害原因究明は人手により行われるがその手法は確立されておらず，多くの場合は経験則などに頼っているのが実情である。

２－２対象システム

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その規模が中・小規模なものである。小規模システムはサーバ台数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。本報告ではこれらの規模を対象として扱う。一般にシステム障害は以下の２つに大別されると考えられる。

A：必ずしも製品の不具合，故障等の問題が発生していないが，システム管理者，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

B：ソフトウェアまたはハードウェアで不具合，故障等の問題が表面化している状態

Ａは使用者の視点で，Ｂは技術的な視点の障害と考えられる。障害の定義には更に深い検討が必要と思われるが，本報告ではＢのみを対象とする。

障害発生時に使用者は何らかのエラーメッセージを受けこれを認識するのが一般的である。殆どの場合このエラーメッセージを発出した箇所から調査が開始される。このとき調査開始箇所を発動点と呼ぶ。発動点は必ずしも障害発生部位と一致しているとは限らない。発動点と障害発生部位の関係例を以下に挙げる。

例１： ftpクライアントがファイルget操作途中で転送が中断し失敗した(図 1a)。
原因：ftpクライアントPC内蔵のFirewallに設定された同時転送制限量を超え強制中断された。
例２： ディスク・ドライブからの読み出し中にOSのシステム・ログにASC/ASCQ＝0x3/11 が記録され，読み出しが失敗した(図 1b)。
原因：ディスク・ドライブ内で該当ブロックが読み出せない状態となっている。

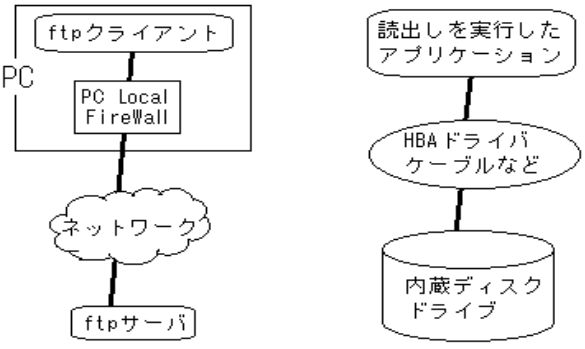

図１ 異なる２つの障害の視覚的概略表記

例１での発動点はftpソフトウェア自体であり，障害発生の原因となったFirewallと異なる部位である。例２ではASC/ASCQを報告したのは問題が発生したディスク・ドライブ自体で発動点と障害部位が同一である。

３．機能線による障害部位特定手法の提案

[定義] 障害をシステム内のある二点間でデータ転送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

†篠原昭夫 日本大学，Nihon University
‡泉 隆 日本大学，Nihon University

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]。

図 1(a)と図 1(b)を比較すると様相の異なる障害の視覚的概観が類似していることに気付く。図 1(a)の中でftpクライアントからftpサーバ上の目的ファイルのデータに至る経路が機能線である。同様に図 1(b)ではアプリケーションから内蔵ディスク・ドライブ上の目的ブロックに至る経路が機能線である。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．始点は目的データの格納場所とする
３．終点は目的データを要求した場所とする

障害は機能線上におけるデータ転送の失敗であると考える。この考えに基づくと機能線上には障害原因部位が必ず存在する。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正しく格納されていなかったために発生したと捉える。上記規則により作成された機能線で記述できない障害は時間的要素を含んだ事象である(データ遅延到達で障害が発生している場合など)。機能線に時間的要素を含ことも可能であるが，本報告ではこれには言及しない。

図 2 は調査対象のシステム全体と機能線の関係を示したものである。この図から明らかなように機能線での障害調査対象はシステム全体の及んでいない。このことが機能線を用いることにより障害調査対象を削減可能とする最初の段階となる。

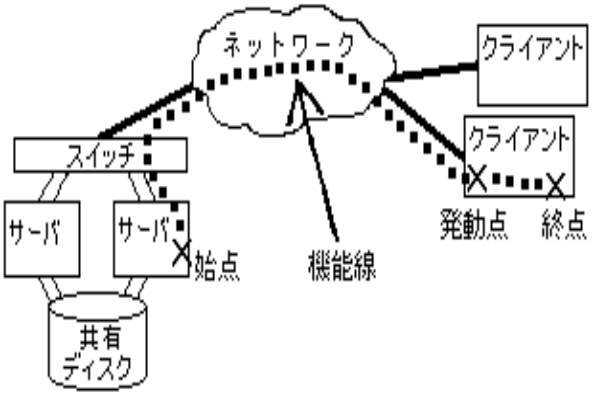

図 2 システム全体と機能線の関係

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目的とするデータ転送に関与するものである。

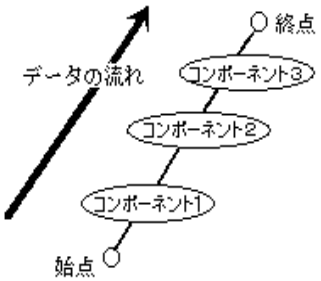

図 3 機能線とコンポーネント

図 3 は３個のコンポーネントからなる機能線を視覚的に表現したものである。図 1aを用いればこの機能線上のコンポーネントは、PC、ネットワーク、ftpサーバとなる。コンポーネントの実例をさらに挙げる。

・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)

この例からもわかるように、コンポーネントとはオープン・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，その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捉える。コンポーネントの具体例としてiSCSI構成を挙げる[2]，[3]。 図 4 はiSCSIを構成する要素を示したものである。iSCSIではハード基盤となるEthernetカードの上位にTCP/IPが動作するためのソフトウェアがあり，さらにその上位にSCSI転送に関連したiSCSI Coreが実装されている。図中で製品AはiSCSI Coreの部分をハードウェアとして実装しているが，製品Bではソフトウェアで実装されている。これらは製品ベンダの設計方針の違いであるが，機能線上ではこの差異を意識せずにどちらも単一のコンポーネントとして扱うことができる。

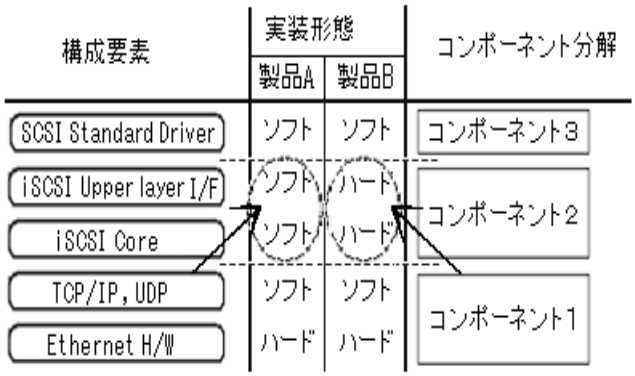

図 4 コンポーネントと製品実装の関係

３－２ コンポーネントの分解レベル

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，あるコンポーネントが障害発生部位と特定できても具体的な対策立案には不十分であることがある。このためコンポーネントをより小さいコンポーネントに分解し，それらの中で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。このことを分解レベルの引きあげと呼ぶ。

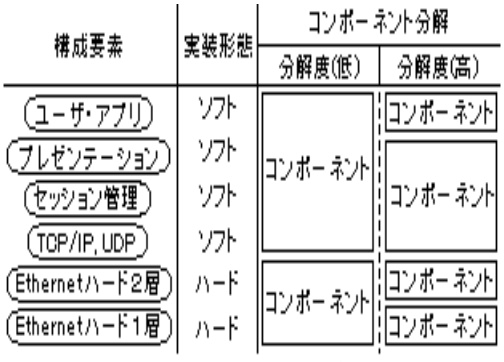

図 5 コンポーネントと分解レベル

図５はコンポーネントの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２個の，高い場合には４個のコンポーネントに分解されている。前出のコンポーネント例であげたディスク・アレイ装置を分解レベルにあてはめると、分解レベルが一番低い場合にはアレイ装置全体が１個のコンポーネントである。分解レベルをあげるとRaidコントローラ，ディスク・ドライブ，外部接続インターフェース，電源モジュールなどのコンポーネントに細分できる。この中でRaidコントローラはさらにメモリー、Firmware、Raid Coreチップなどに分解される。コンポーネントは調査の深度要求に応じ分解レベルをあげることが可能である。

図６は機能線と分解レベルの関係を表したものである。図中左側のコンポーネント２は分解レベルを一段階あげることによりコンポーネント２－１、２－２、２－３に分解される。さらにコンポーネント２－２はコンポーネント２－２－１、２－２－２に詳しく分解される。

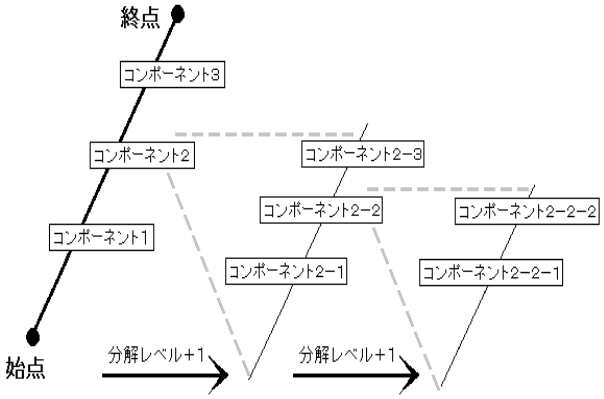

図 6 機能線と分解レベル

３－３ 分解レベルと調査深度の関係

コンポーネントの分解レベルを上げる際には，製品構成を意識する必要がある。これは障害調査の最終目的が対策立案であることが殆どのためである。例としてSCSIカードと周辺ファームウェア，ソフトウェアのコンポーネント分解を考える。 一般にバス・スロット追加方式のカード製品はFirmware提供がCodeとNVRAMで一体となっている。障害部位の詳細に到達するためFirmwareを更に分解しての調査を行うことは可能である。しかしながら障害対策投入時にCodeとNVRAMを個別に変更・更新することは困難なため，分解レベルを高くしたことが必ずしも効果的となりえない。また分解レベルが比較的高い状態では，隣接するコンポーネント同士は相互接続互換等の制約が厳しいことが多い。このことも高い分解レベルのコンポーネント単体での対策投入を困難にしている。この例ではCodeとNVSRAMは事実上リリースが連動し，単体での変更を行えない。一方で再発防止までを考慮した深々度の障害部位特定が必要な場合には，分解レベルを相応に高める必要がある。このことはハードウェア製品の障害調査時に多い傾向がある。

分解レベルを厳密に定義して機能線を作成する必要はない。また，ある機能線上に存在する全てのコンポーネントが同一分解レベルである必要性もない。障害調査の初期段階では分解レベルは低く抑え，調査の進展にあわせて順次レベルを上げるのが最適かつ最短な方法である。この際も全てのコンポーネントの分解レベルを一律に上げず，被疑性の高いコンポーネントのみを対象とする。また図７(a)と図７(b)の比較からわかるように，分解レベルを上げることは調査対象コンポーネント数の急激な増加を招く。このことからも分解レベルは可能な限り低い状態から調査を始めることが望ましい。

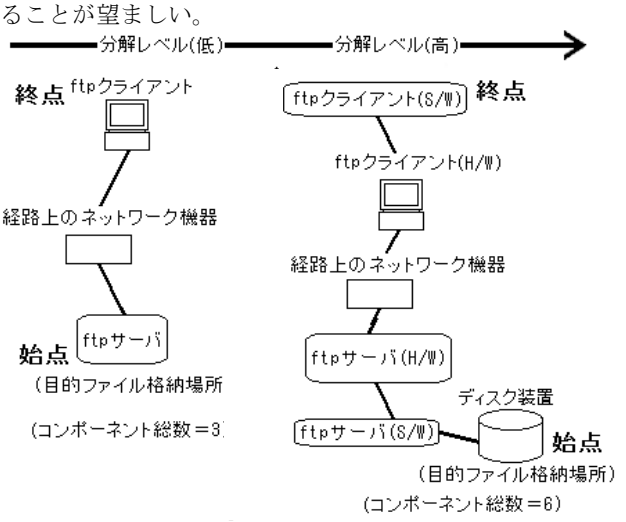

図 7 分解レベルとコンポーネント総数の関係

３－４ 補助機能線

[定義]作成した機能線上の調査対象を削減するため「補助機能線」を定義する。

補助機能線とは調査対象の機能線と一部が重複する別の機能線で，この線上では障害が発生していないものをいう。図８では障害の発生していないクライアントBから点３までの区間が補助機能線である。この例では補助機能線を用い点１から点３の区間を調査除外区間と判断し，調査対象点３から点２に絞り込むことができる

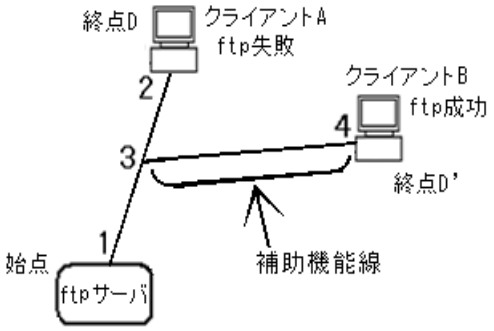

図 8 補助機能線

３－５ 機能線の生成方法

機能線の生成は以下の手順で行う。

step1－発動点での情報を元に始点と終点を決定する

step2－始点－終点間に存在しうるコンポーネントを可能な限り低い分解レベルにて列記する

step3－列記されたコンポーネントを，データの流れ(終点方向)に向かって並べる

さらに補助機能線情報が存在する場合にはこれを併せて作成する。図８の例では補助機能線は１つであるが、これは複数存在することもありうる。その際は効用の予測にとらわれることなく調査に有効な情報である前提で作成しておくことが好ましい。

３－６ 機能線を用いた障害部位特定

機能線作成後は始点から終点方向に向かい順次各コンポーネント内での障害発生有無を判定する。調査方向はこれと逆でも問題ない。実際には始点，終点のどちら側の近傍に障害発生コンポーネントが存在するかを経験則で判断し,結果的に調査時間を短縮している例が多く見られる。初期段階では総コンポーネント数を少なくしておくことが重要である(図 7(a),図 7(b))。これにより被疑コンポーネント総数が減り，効率よく調査を進めることができる。被疑コンポーネントが確定した後は，必要に応じそのコンポーネントのみ分解レベルを上げ更に詳細な調査を行えばよい。

３－７ 機能線を用いた障害部位特定の例

機能線を用い障害部位を特定する例とし再び２章の ftp によるファイル取得失敗を挙げる。この例では補助機能線を活用し障害発生部位がクライアントＡに近い側にあることを判定している。図 8 はその概略である。図中で ftp が失敗したクライアント PC は A で，調査対象の機能線は区間 1-3-2 となる。また付帯情報で ftp クライアント B は,サーバーからの同じファイルの取得に成功したことが判明しているとする。この補助機能線は区間 3-4 となる。補助機能線の定義から区間 1-3 は調査対象から除外でき，この結果対象は区間 3-2 に絞り込まれる。実際の環境では区間 3-2 の間にネットーク機器等が存在する可能性が高いが,図 8 の例での分解レベルでは ftp クライアント A 自体が次の調査対象となる。もしこの結果 ftp クライアント A に問題がないと判定されれば，次の調査対象は区間 3-2 上のネットーク機器となる。よって以降は区間 3-2 の分解レベルを１段上げ調査を進める。

実際の機能線を用いた障害部位特定では補助機能線を多用する。またこれを意識していなくとも結果的に補助機能線の情報を利用して調査している事例は数多く見受けられる。

４．機能線の考察

４－１ オープン・システム障害調査における有効性

オープン・システムでは同一の機能が必ずしも同じ実装形態で提供されるとは限らない。あるベンダがハードウェアで提供している機能を，別のベンダではこれをソフトウェアで提供していることがある。機能線ではこの差異を意識することなく調査を行うことが可能である。プログラマブルASICによる回路実装，ハードウェアを特定のソフトウェアで限定用途に使用するアプライアンス製品，これらが共に増加する状況では機能線のこの性質は非常に有効である。

４－２ 機能線と分解レベルの関係

図 6 から明らかなように，機能線の分解レベルを上げることは，レベルを上げようとしているコンポーネント自体を新しい機能線全体とみなすことと同値となる。すなわち機能線は分解レベルを通じて階層的な構造を持っている。

さらに機能線のこの性質をその規模を拡大する方向に用いれば，大規模ネットワークで相互接続される複数の個別システム間で発生した障害調査にも応用可能である。この場合は各システム内での機能線をより大きな規模のシステム内での個々のコンポーネントと解釈することとなる。

４－３ 複数組織間の障害調査時の機能線の有効性

オープン・システムは独立して開発された製品を統合して構成されているため、障害調査が複数のベンダ、調査組織を跨いで行われることがある。システム開発・保守を請け負ったベンダが障害調査を実施する場合にもそのベンダ内で個々の製品調査に特化した部署が連携して一件の障害調査することが多い。このような状況に機能線を用いると個々の製品のハード・ソフト構成、特徴を意識することなく障害を同一の様式で記述できるため、組織間の連携を行いやすくなる。また各組織の調査担当範囲は、機能線上のコンポーネントに対応していることが殆どである。更に連動組織が複数ベンダを跨ぐ場合には、機能線により障害が統一様式で記述されていることで情報交換を円滑に行うことができる。

５．各コンポーネントでの障害発生有無の判定方法

３章では機能線の定義を行いコンポーネントとの関連について述べた。また障害調査量を削減する方法としてコンポーネントの分解レベルと補助機能線を導入した。本章では機能線を作成したのちに実施する各コンポーネント内での障害発生有無の判定方法を提案する。提案する比較法は調査対象システムが正常に稼働している際に採取しておいた情報を、障害発生時の各コンポーネントが正常に稼働していたかを判断するための基準として用いるものである。

５－1 比較法

機能線上の各コンポーネント内で障害が発生したかを判定する方法として比較法を提案する。これは対象システムの正常稼動中に事前収集しておいた情報と，これと同じ対象を障害発生時に収集したものと比較，不一致部分があればそれを障害発生部位と判定する方法である。一般にオープン・システムでは同一バージョンの製品で構成された異なるシステムは，正常稼動時に必ずしも同じ挙動をしていない。このためあるシステムでの障害事例を別システムの障害調査の参考にすることが困難なことがある。比較法では当該システムで正常時に収集しておいた情報を各コンポーネント内での障害発生有無を判定するための指標として用いる。更にオープン・システム向けの製品は動作仕様の詳細が公開されていないことが多く，このことも比較法が有効な理由となっている。

５－２ 比較法を行うための準備

以下はシステム正常稼働時に採取しておくと有効と考えられる情報の例である。

A:システム起動中の各コンポーネントからの報告
B:各コンポーネント独自のログ採取機能からの情報
C:対象システムの使用目的であるサービスの正常時の処理時間
D:各コンポーネントで自動記録される定期・不定期情報
E:ネットワーク，I/O パスでの帯域使用率
F:単位時間あたりで計測かつ数値化可能な情報

例１：システム正常稼動中サーバーの１日当たりのログ情報量(MB/day)
例２：ディスク装置に格納されるユーザー・データ量の１ヶ月あたりの増加量

比較法に利用可能な情報の種類は多岐にわたるが，次の方針で収集情報を決定することが望ましいと考えられる。
(1) 対象システムの使用目的に注目する（前出：C,E,F）
(2) 対象システムで使用されているコンポーネントの特徴に注目する(前出：A,B,D)

これらの情報は個別に収集されるため，機能線上のコンポーネント構成とは一致しない。OS のシステムログ・メッセージを例とすれば，これは複数コンポーネントから個別に報告された情報が報告順に記録されているだけであり，各コンポーネント間の相関情報，因果関係の情報等は含まれていない。このため収集された情報を比較法で使用するには，個々の情報がどのコンポーネントに対応するかを分別し，コンポーネント毎に時系列に再配置する作業が必要となる。図 9 は OS のシステムログとのコンポーネントの対応を示した例である。

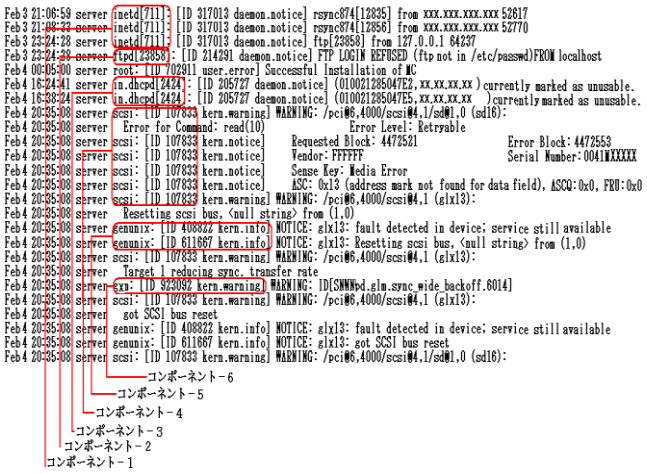

図 9 システムログとコンポーネントの関係

５－３ 比較法での留意点

比較法ではシステム正常稼働時に情報を収集しておくだけでなく、それらを事前に調査、分析しておくことが必要である。以下の(1),(2)は事前実施する項目である。

(1) 採取情報の時差修正

各コンポーネントから個別に収集した情報は記載時刻に相対的な時差を含んでいる。例えばサーバ本体に外部接続されたディスク装置はサーバとは独立した時計内蔵している。このためディスク装置内部から採取されたログ情報はサーバ本体の時計を基準に採取された他の情報と時差が生じる。こさらにこれらの外部接続装置は開発国の標準時刻のみを基準にログ情報を記録している場合がある。以上の理由から機能線全体での基準時刻を定め、個別に採取された情報の時差を修正する作業が必要である。更に事象が発生した実時刻とそれが記録されるまでの時差にも注意を要する。また事象が何秒かに渡って継続するものである場合には，ログに記録される時刻はその開始時刻か終了時刻のどちらかが考えられる。このような挙動の判定は正常稼動時に採取した情報を事前に分析、調査しておくことで判別可能となる。

(2) 調査対象の障害に関連しないエラー記録の除去

あるコンポーネント内で検出された障害発生記録が調査対象の障害と関連していないことがある。これは収集された情報が調査対象の障害の発生前、後を網羅しているためである。経験的には前出[1]の時差修正が完了しているならば，長時間に渡り同じ障害での事象が繰り返し記録されている場合の除き，調査対象の障害発生時刻の前後数分より長い時間離れた時刻のエラー記録は対象外と判断できることが多い。

５－４ 比較法の実例

システムが正常に稼働している記録と、調査対象の障害記録が混在している例を図 10 に示す。図中で繰り返し記録されている①、および１件のみの記録のある②はシステム正常稼働に伴い定期的に記録される情報である。③が障害を記録している部分であり調査の対象部分である。調査対象と関連のない①、②を除去するためにはシステム正常稼働時に採取された情報との比較作業が必要となる。

付記：事前情報収集の副次効果

比較法での使用を目的に事前収集された情報は，発生した障害への対策に有効利用できることがある。例えば障害の影響でシステムの稼働に必須となる設定・構成情報などが消失した際に，これを修復するため事前収集情報を用いることがある。また製品によってはその設定、構成情報をバックアップする手段が提供されていない、またはそれらの情報は手動でのみ採取可能なログに含まれる仕様となっていることがあり、事前採取情報が唯一の保管手段となっている。このことは比較法の本質とは無関係であるが，オープン・システムでの障害調査・対策実施の過程では重要であり障害の規模が大きいほど事前収集情報が利用される頻度が高い傾向が見受けられる。

図 10 正常情報と障害情報

６．まとめ

オープン・システムで発生した未知の障害は人手による調査が行われるが、その手法は一般化されていない。これらの障害を調査する手法として機能線を提案した。これはシステム内で発生した障害を記述する機能線を提案した。

システム内の隣接コンポーネント同士が通信を行っていることに着目し，目的データが要求された地点に到達しないことを障害と考える。この定義により機能線上には必ず障害発生部位を含んだコンポーネントが存在することになる。また補助機能線を効果的に用いて調査対象区間の削減を可能とする。クライアント・サーバモデルであるオープン・システムの障害調査では補助機能線は活用機会が多い。さらに機能線では複数ベンダーから提供される製品をハードウェア，ソフトウェアの違いを意識せずに，一様にコンポーネントとして扱えるという特徴がある。

次に機能線上のどのコンポーネント内で障害が発生したかを判断する方法として比較法を提案した。これはシステム正常稼動時に事前収集された情報を障害発生後に同様の方法で採取の情報と比較，通常時には記録されない情報を障害発生の痕跡であるとして注目する。オープン・システム向けの製品はその内部仕様の詳細が公開されていないことが多く、このことも比較法が有効な手段である理由のひとつとなっている。また比較法に用いる目的で事前収集されていた情報は，設定・構成情報が破壊，消失された障害への対策にも効果的に利用できる。

オープン・システム向け製品が市場へ投入される速度は依然衰える傾向はなく、さらにバージョン・アップ、修正プログラムのリリースは間隔が短くなる傾向もある。これらの要素は未知の障害の発生率をたかめることとなる。このことは人手による障害調査の必要性につながり、機能線を活用する機会は今後も減少しないと考えられる。

機能線を用いることにより異なる製品で構成された異なるシステムで発生した関連のない障害を統一された形式で記述することができる[4]。この特徴を利用し今後は機能線により独立した個々障害を統一された視点で分析、分類することを考案してゆきたい。

[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] John L.Hufferd: “iSCSI The Universal Storage Connection”, Addison Wesley(2003)

[3] Robert W.Kembel: “Fibre Channel A Comprehensive Introduction”, Northwest Leading Associates,Inc. (2001)

[4] 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）