

LAN 環境における Trouble Tracking Ticket System(T3) の構築

泉 裕† 山本 茂† 山口 英† 山本 平一†

†奈良先端科学技術大学院大学情報科学研究科
†大阪大学大学院基礎工学研究科

内容梗概

本研究では、LAN 内で発生した障害の復旧支援を目的とした Trouble Ticket System を開発を行っている。Netlog や Concert Ticket System 等、従来のシステムでは、複数の管理者による管理体制下で運用する際、障害復旧の作業の分配や障害履歴の検索、長期的な運用が効果的に行えないので、刻々と障害が起こる状況に対応できない。そのため、複数の管理者が存在する LAN において長期的な運用でも効率的にネットワーク管理を行うことができる Trouble Ticket System が必要とされている。本稿では、従来の Trouble Ticket System の問題点について述べ、LAN 内で Trouble Ticket System を長期的かつ効果的に運用するための必要条件や機能について考察し、それらを満足する機能を備えたシステムである T3(Trouble Tracking Ticket System) を提案する。

Design and Implementation of Trouble Tracking Ticket System for LAN management.

Yutaka Izumi† Shigeru Yamamoto† Suguru Yamaguchi† Heichi Yamamoto†

†Graduate School of Information Science Nara Institute of Science and Technology, Ikoma, Nara, JAPAN
† Faculty of Engineering Science, Osaka University

Abstract

It becomes popular to use Trouble Ticketing Systems for trouble shootings in the network management. However, since currently available systems like Netlog or Concert Ticket System do not support functions for scheduling of trouble shooting tasks, communicating among network managers, and systematically analyzing registered tickets, while these functions are considered to be essential for network trouble shootings. In this paper, we address problems around currently available trouble ticket systems in the case that they are applied for network management tasks, and requirements on the trouble ticket system are pointed out. Finally, we propose the T3 (Trouble Tracking Ticket System) and show its design and implementation.

1 はじめに

現在、大学や研究機関で導入されている LAN の管理体制には、障害の復旧を行う管理者が複数で構成されている形態がよく見られる。しかしその体制下で管理を行う場合、障害の復旧作業を効果的に分配することができないために一部の管理者に高負担がかかってしまう等の問題がある。

このような問題が発生する背景として、管理者間の意志の疎通を十分に行えないために LAN 内の管理状況の把握が困難になるという状況がある。管理者の人員不足が指摘される折、管理者自体も資源と

して考えた場合の資源の有効利用は重要である。そのため Trouble Ticket System という障害復旧作業におけるプロセス管理を行うツールを導入した管理を試みているサイトが増えている。しかし、既存の Trouble Ticket System には、依然として複数の管理者に効率良く作業を分配できないことや記録の検索が十分でないことなどの問題点がある。

本稿では、Trouble Ticket System の概念とその問題点について考察し、新たな Trouble Ticket System として、T3(Trouble Tracking Ticket system) を提案する。T3 は従来の Trouble Ticket System

に比べて効果的な作業分配や検索が行えることを目的としており、現在のネットワーク管理体制下での長期的な運用を考慮している。

2 Trouble Ticket System の概念

2.1 ネットワーク管理での位置付けと役割

大規模化かつ複雑化してきたネットワークを運営していくためには、ネットワークを構築しているハードウェアならびにソフトウェアの整備や保守が必要である。ネットワークを運営していくための一連の作業をネットワーク管理という言葉で表現するが、その作業は多岐に渡っており、計算機ならびにネットワーク技術全般の知識を必要とする。その管理のカテゴリには、構成管理、障害管理、性能管理、セキュリティ管理、アカウント管理などが挙げられる。Trouble Ticket System は複数の管理者間で協調作業を実現するために考案されたもので、グループウェアの一線としてとらえることができる。Trouble Ticket System は障害管理に焦点を当てたシステムであるが、システムに蓄積された障害記録を分析することにより、その結果を構成管理や性能管理に利用することができる。

障害管理の一環と考えた場合の Trouble Ticket System は、一連の管理作業において障害の履歴を記録する役割を担っている。障害管理の詳しいプロセスをフェーズ毎に分けると、障害の発見、障害の診断、障害の復旧という3つがある。基本的に Trouble Ticket System はそのすべてのフェーズにおいて障害の症状や対処法を記録する。すなわち、障害を発見した時点から復旧に至るまでの記録をとることが障害管理におけるシステムの役割となる。

Trouble Ticket System を構築する際には一般的な指標が必要となるが、「良い Trouble Ticket System」という定義はされていない。しかし、文献 [1] にはその理想とする目標が示されている。その内容にはいくつか項目が挙げられており、整理すると以下ようになる。

ネットワーク管理状況の把握 現状のネットワークにおいて、未解決の障害や誰がどの障害に対処しているかを把握できる。

複数の管理者による参照 障害復旧にあたる管理者

は、障害が以前にも発生したものであればその記録を参照することで復旧作業を効率良く行うことができる。

作業のスケジューリング 誰も対処していない障害があれば、障害復旧作業を担当者に分配できる。

統計的な分析 長期に渡って情報を蓄積し分析することで、障害発生頻度の高い箇所を特定したり、障害復旧作業を行う管理者の評価等の統計的な分析ができる。

現在、上記の項目の実現を目指して様々な Trouble Ticket System が考案されている。その基本的な機能と利用形態について述べる。

2.2 基本的機能と利用形態

現在、一部のサイトで利用されている既存の Trouble Ticket System は、図 1 に示される運用形態を持っている。

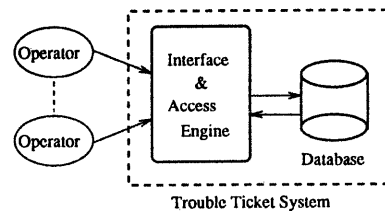


図 1: 運用形態

複数の管理者 (Operator) はそれぞれの管理作業を行い、システムに対し同等の権限で記録を行う。しかも分配された作業に対し復旧するまで責任を持って対処する。

Trouble Ticket System の機能とは、基本的に「Trouble の症状や対処法を書いた Ticket を発行することと、その Ticket を検索すること」である。その書式や機能は文献 [1] において基本的な形式が示されており、Netlog Trouble Ticket System¹や Concertnet Trouble Ticket System²など既存の Trouble Ticket System はその形式に従って設

¹JvNCnet で運用されている

²Concertnet で運用されている

計されている。以下に基本的な Ticket の書式、発行と検索の機能について述べる。

Trouble Ticket System で用いられている Ticket には一定の書式 (構造) があり、その書式に従って管理者が記述していくことになる。基本的に Ticket は header と body で構成されている。その構成は以下の通りである。

- header(paramater)

number	Ticket の番号
time	発行された日時
status	作業状況 (Open/Update/Close)
operator	作業担当者の名前
subject	障害内容の表現

- body(free text)

header の各項目に、後に Ticket を検索するためのパラメータを入力し、body には障害の具体的症状およびそれに対する作業の内容やその後の変化した状況等を詳しく記述する構成になっている。

上記のパラメータはごく基本的な項目に限って記述しているが、Trouble Ticket System はこの書式に従って Ticket を発行する。発行は障害復旧への作業毎に行われ、Ticket 内の status がその作業の区切りを示している。よって以下に示した発行の時期は、status の項目を用いて表現されている。

- Open Ticket : 障害の発見
- Update Ticket : 復旧作業の実行
- Close Ticket : 障害から復旧

一連の Ticket の発行は、具体的にその Ticket をデータベースに格納する動作を示している。従来の Trouble Ticket System 動作モデルを図 2 に示す。

図 2 で、横軸は時間軸を示している。つまり、障害を自分で発見した時や報告を受けた時点から、その作業の内容と経緯、復旧までのプロセスを各 status を持った Ticket に書き込むという機能を Trouble Ticket System は持っている。

既存の Trouble Ticket System の検索は「現在 Open している Ticket のリストを表示する」と「特定のパラメータに該当する Ticket を検索、表示する」機能を持っている。前述の Ticket のパラメー

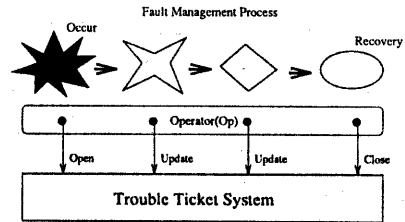


図 2: Trouble Ticket System

タは基本的な項目に限られているが、より効果的な検索を行うためには、これだけでは不十分である。

現在利用されている Trouble Ticket System の多くは、上記の形態と機能を持っている。しかし、今後増加するであろうネットワークの管理体制下では効果的に利用できないことが考えられる。以下にその問題点の抽出と考察を行う。

3 従来の Trouble Ticket System の問題点

前述の Trouble Ticket System の運用形態では、複数の管理者が同等の権限を持ってシステムを利用することが前提となっており、少数の統括管理者と多数の復旧作業で構成されるサイトでの長期的な運用は困難であると考えられる。その問題点を以下に述べる。

- 少数の統括管理者への高負担
- 管理者の育成が困難
- 長期における運営が困難

現在、大学や研究機関等のネットワークでは、ネットワーク全体の管理状況を把握する少数の統括管理者 (Master Operator:MOp) と、障害に対処する多数の作業管理者 (Operator:Op) で構成された体制を持つサイトが多い。しかし、既存の Trouble Ticket System ではこれらの管理者の区別ないため、作業分配の効率が悪い。従って MOp は障害管理の作業全体に関わらなければならない、結果的に高い負荷がかかる。また、ネットワークを管理する人間の構成

が年々変化していく大学等の環境において、Opとして経験の浅い管理者は、現在までの管理状況やその障害の履歴を十分把握できないために管理者に必要な知識や技術を取得することが困難である。

更に、ネットワークサイトごとに環境が違うため、Ticketに必要なパラメータが異なる場合や、長期に渡って運用して行く際にネットワーク内の機器の構成が変更される場合、それらを管理対象に加える必要が生じる。既存の Trouble Ticket Systemでは、固定されたパラメータしか持たないため、後者のように、運用を継続することが困難である。

これらの問題点の発生は、以下の点に起因していると考ええる。

- P1:限られた検索方法
- P2:作業分配のトレードオフ
- P3:固定されたスキーマ

既存のシステムは、statusがOpenのTicketを検索するか、限られたパラメータでTicketを検索する機能(P1)しか持っておらず、関連を持つ複数のTicketの表示(同じnumberを持ったTicket間の関連)など、一つの障害に関する詳しい履歴を効果的に見ることが出来ない。

障害復旧作業を、担当する管理者に強制的に振り分けて行く形では、効果的に作業の分配が行えるが、作業を分配する管理者にそのオーバーヘッドがかかり、結果的に一部の管理者の負担が一部大きくなる。反対に、作業を受け持つ人間が自主的に自分の受け持つ作業を獲得する形にすると、TicketをOpenした人間がそのまま作業をすることになり、少数のMOpがネットワークの管理状況を把握するというシステムの目的の達成できない。すなわち、既存の Trouble Ticket Systemでは、このトレードオフを抱えたままでしか運用できない(P2)。

自サイトのネットワーク管理を長期的に行う場合、機器や管理者の構成変化に伴って、Ticketの持つ項目(データベースにおいてはスキーマ)を変更する必要が出てくる。現在の Trouble Ticket Systemにはスキーマを変更する仕様が無い(P3)ため、新たにスキーマを変更した Trouble Ticket Systemを立ち上げた時に、変更されたデータと整合性を保つことが難しい。

上記の問題点とその原因(P1,P2,P3)は、実際には自サイト³で従来のシステムを運用した結果、更に明らかになった。これらの問題を抱えたままでは、今後増加していくネットワークの管理体制下での長期運用は望めない。そこで、これらの問題点の解消を目的とした新たな Trouble Ticket Systemを提案する。

4 T3の構築

前章で述べた問題点を考慮した新しい Trouble Ticket Systemとして、T3(Trouble Tracking Ticket System)を提案する。このシステムは、単一サイトでの管理運用を支援するシステムとして、従来の Trouble Ticket Systemに比べ以下の点で機能的に優れている。

作業分配の効率化 復旧作業の状況を把握する

検索の効率化 様々な項目で効率良く検索が行える

長期運用に必要な機能 ネットワークの状況変化に柔軟に対応する

これらの動作(Trackingとよぶ)を行うことのできるシステムを構築する上で、対象となる管理形態と新たに追加した機能、実装方法について述べる。

4.1 利用する形態

本システムでは、現在の多くの大学や研究機関での管理体制を考慮して、図3の形態で運用することを前提としている。

Opの役割は、障害復旧のための作業を行う人間として、作業を行う際にTicketを発行(Open,Update)することである。よってデータベースへのアクセス権と、必要なカテゴリ(後述)の追加、削除を行う権限を持つ。

MOpは、TicketをCloseする権限と障害の復旧作業をOpに分配する権限、データベース内のスキーマ(後述)を変更する権限を持つ。それ以外はOpの権限に準じているので、Opと同様の作業を行うこともできる。

³奈良先端科学技術大学院大学における管理体制下で運用

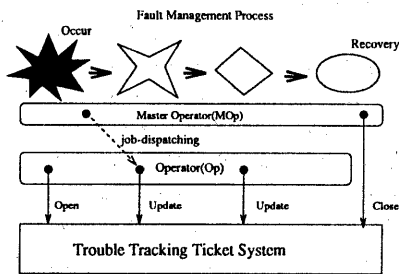


図 3: Trouble Tracking Ticket system(T3)

4.2 新しい機能

この形態において、前述の問題点を意識した運用を可能とする T3 の特徴を示す (括弧中の記号は、対応する前述の各問題点を示す)。

- Inner status による効果的検索 (P1,P2)

従来の Open/Update/Close による status の表現だけの検索のアルゴリズムでは「現在 Open されている Ticket」を検索する際、status が Open の Ticket number を検索すると、既に Close されている Ticket も該当するため、Close されていない Ticket に関しては、別に index を作成してテーブルの中に該当する Ticket number を記録し、Close された時点でそのテーブルから削除する方法が用いられている。しかし、この方法では、「現在誰も担当していない Open された Ticket」等の複合条件で検索をする場合に対応できないので、以下の flag を導入することでより効果的な検索ができる。

recovery flag	: 現在の復旧状況
treatment flag	: 担当者の有無
reason flag	: 原因判明の有無
finish flag	: Close 待ち

これらのパラメータは同じ番号すなわち一つの障害に関する Ticket 全てに書き込まれるもので、これによって index による照合も必要なくなる。様々な複合検索が可能となり現在の状況や一つの障害履歴に関する情報を効率良く得ることができる。

treatment flag で担当者の有無を確認した後、誰もその作業を行っていない場合はその作業を獲得す

る機能を持たせることで、Op の復旧作業獲得の自主性を保持しながら、MOp の負担を軽減できる。また、この情報を集計することで「誰がどれだけ仕事を受けたのか」等の統計情報を収集できる。

recovery flag は、復旧作業が終了したか否かと、復旧できなかった場合を示すフラグである。作業管理者によって復旧ができなかった場合、このフラグを立てることで他の作業管理者に引続き対処を依頼できる。

他にも、管理作業の中には原因が特定できないうちに復旧してしまうこと象も起こり得るため、MOp は reason flag を確認し Close する。原因が判明していないまま Close flag が立っている Close 待ちの Ticket を Close するか否かは MOp の判断で行う。これらの flag より、作業分配の際のトレードオフを解決し、効果的に作業の分配や検索、障害履歴の参照が行える。

- Category free による効果的な検索 (P1)

同様の障害が生じた場合、以前の記録を参照するために障害を予め分類しておくことが重要となる。しかしその種類はサイトによって種類が異なるため、将来において必要となるものやそうでないものが生じると考えられる。これに対し、Ticket の中に Category field を設定して、障害のカテゴリとしてキーワードを登録するためのデータベースを作成することで対応する。カテゴリの設定によって検索が効率良くでき、各項目内の障害に対する管理技術のノウハウを効率良く取得することができる。更にデータベースへの登録削除により、カテゴリに自由性を持たせ長期運用に対応できれば、そのデータを集めて「何処によく障害が起こるか」等の統計情報の分析も行える。

- Schema free による長期運営への対応 (P3)

header や inner-status の項目は、システムの長期運用で変更する可能性がある。その際の項目 (データベースにおけるスキーマ) を追加削除できる機能を持たせることで、長期運用に必要な項目の変更を行うことが出来る。この項目の変更は、管理の際に必要な項目がサイトによって異なることを考慮した機能で、時間的推移と共にそのサイト内の管理状況が変化しても、柔軟に対応することが可能となっている。

4.3 実装

4.2.2 章で述べた機能の実装はデータベースへのアクセスエンジン内で行う。その実装部分の位置づけを図4に示す。

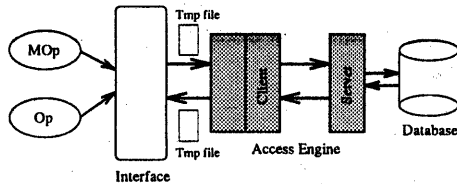


図4: implemetation

データベースとして、現在 Relational Data Base(RDB)である POSTGRES を用いている。アクセスエンジン(C言語により記述)はユーザインターフェース(tkで作成)とデータベース間に位置しており、ユーザインターフェースが作成したテンポラリファイル(項目名とパラメータ、フラグ、テキストを書き込む)をデータベースに格納する形を取っている。検索の場合は逆を行う。

5 将来への展望

本システムの運用による利点は大きいですが、まだ発展させるべき点は多くあると考える。以下にその将来性についての考察を述べる。

Category と記述内容の構造化 本システムにおけるカテゴリは、本来「何処が悪いのか」を正確に特定するために必要な項目である。現在のT3においては、そのカテゴリを複数指定することで障害の発生場所を特定することになっているが、そのカテゴリを構造化することでより詳しく、より正確に発生場所を特定することができる。また、記述内容の具体的な内容は全て Free Text となっているために、必要最小限の情報を記述しても、文法を特定していないと作業の自動化に向けての改良が十分にできない。従って、その記述法についても新たな方法を提案する必要がある。

他のシステムとの連動 障害管理には前述のように3つのフェーズがあり、各々のフェーズにおいて様々なツールが開発されているが、それぞれのフェーズで利用されるツールはお互いに連動していない。本システムがその障害の発見から復旧にいたるまでのプロセスのノウハウを提供することができれば、そのプロセスの自動化を図ることができる。また、本システムは将来においてそれを提供するために十分機能することができると考える。

Internet での利用 基本的に本システムは LAN 環境を前提としており、Internet での利用は今回考慮していない。実際に各サイトでの情報というのは他のサイトにおいても有効である場合が多いと考えられるので、サイト間での情報の交換は必須である。

6 おわりに

本研究では既存の Trouble Ticket System の実験運用を通じ、従来のシステムにおける問題点を明らかにすると共に、これらの問題点を解消する新たな Trouble Ticket System として T3 を提案した。T3 には Ticket の発行や検索をより効果的に行うための機能を持ち、長期的なネットワークの運用に柔軟に対応できる。

今後は T3 を奈良先端科学技術大学院大学内ネットワークに実装して運用を行うことにより、T3 のさらなる拡張について検討していく予定である。

参考文献

- [1] D.Johnson. NOC Internal Integrated Trouble Ticket System Functional Specification Wishlist("NOC TT REQUIREMENTS"). RFC1297, January 1992