

メールのスレッドをチケットとして用いた トラブルチケットシステムの構築

99H003 赤松康介

目次

1	はじめに	1
2	従来のトラブルチケットシステム	2
2.1	トラブルチケットシステムの概要	2
2.2	既存のシステム	3
3	本研究で提案するシステム	5
3.1	チケットの運用	5
3.2	メール	7
3.2.1	メッセージヘッダ	7
3.2.2	スレッド	8
3.3	メールリスト	8
3.4	他のシステムとの連携	9
3.5	実装	10
4	考察	11
4.1	利点	11
4.2	問題点	11
5	まとめ	13

1 はじめに

今日では、ネットワークで多様なアプリケーションが利用されるようになり、コンピュータを利用する上でネットワークは無くしてはならないものとなっている。そこで管理者は安定したネットワークサービスを提供する必要がある。そのために、ネットワークにおける障害の把握と迅速な対処が求められる。また、ある程度の規模をもつネットワークの管理においては、複数の管理者が協調して作業を行なうことが多い。そのためには障害情報や作業履歴の共有が必要となる。このような障害情報の把握や共有にトラブルチケットシステムが利用されることが多い。

OSUNET¹には、既に大垣齊講師によって構築された OSUNET Monitor [1] が存在する。これは OSUNET を構成するルータやサーバマシンおよびその上で提供されるサービスの稼働状態を監視するシステムである。このモニタシステムは ICMP²[2] を用いて監視対象機器へのネットワークの到達性を確認する。また、監視対象機器で稼働しているサービスの応答を調べる。そして監視対象が停止した (応答が無くなった) 場合、および復旧した場合、その情報をメールで管理者に通知する。

本研究ではこの OSUNET Monitor を利用して、メールのスレッドをチケットとして用いたトラブルチケットシステムを構築した。

¹大阪産業大学学内 network

²Internet Control Message Protocol. IP(Internet Protocol) の機能を補完するためのプロトコル。IP はコネクションレス型のプロトコルであり、IP パケットの送信時にエラーが起きても送信元ホストはそれを知る事ができない。このようなエラーメッセージや制御メッセージを送信元に通知するのが ICMP の役割。RFC792 で定められている。

2 従来のトラブルチケットシステム

2.1 トラブルチケットシステムの概要

トラブルチケットシステム (Trouble Ticket System) とは大規模システムの保守管理やヘルプデスクサービスに利用されている情報管理システムである [3][4][5][6][7][8][9][10]。おもに障害情報の管理や、ソフトウェアのバグの管理などに使われている [11][12][13]。後者はバグトラッキングシステム (Bug Tracking System) と呼ばれる。

チケットは以下のような流れで運用される。

1. チケットの発行

障害が発生すると、チケットが発行される。この段階のチケットの状態をオープン (Open) という。同時に、障害について以下のような情報が入力される。

- 障害の発見者
- 障害の発生時刻
- 障害内容

2. 作業担当者への割り当て

発行されたチケットは作業担当者へ割り当てられる。この段階でチケットは割り当て済み (Assigned) となる。実際に障害に対応する段階で他の作業担当者へ割り当て直されることもある。

3. 作業記録の記述

作業担当者は対処の作業状況を記録する。

4. チケットの終了

障害の復旧作業が終了し、障害が解消されればチケットはクローズ (Close) される。

クローズされていないチケットを探すことによって、未対応の障害を把握することが可能になる。またクローズされたチケットは保存されるので、障害への対応のノウハウを蓄積できる。

トラブルチケットシステムを用いてネットワーク等の管理を行なうことを考える。その場合同じ作業が同時に行なわれないように、管理者の中から作業担当者を決め、作業を行なう。そして作業担当者が交代する際は引き継ぎを行なわなければならない。その際には該当するチケットを参照することで、作業の進行状況を把握することができる。また過去のチケットを利用して、障害に関する統計を出すことも可能である。同様に、過去のチケットの作業記録から管理者の作業を評価することも可能になる。

2.2 既存のシステム

トラブルチケットシステムとして以下のようなシステムが既に存在する。

- 影舞 [3]

Ruby で記述されたバグトラッキングシステムである。日本人の作者によって開発が進められている。GNU GPL³[14] によってライセンスされ、配布されている。日本語をおおむね正しく扱える。ウェブベースでプロジェクトの作成・設定変更・削除が可能である。また、新規アートの投稿、リプライ、状態・カテゴリ管理もウェブから行なうことができる。投稿・リプライをメールで通知することができる。そして、メールでも送信・リプライが可能である。アートの文字列検索も可能である。

- GNATS [4]

C 言語で記述されたバグトラッキングシステムである。GNU GPL によってライセンスされ、配布されている。実際に利用されている例として NetBSD GNATS Bug Database Summary [11] が挙げられる。コマンドラインやメールによる操作が可能である。さらに、ウェブインタフェースやグラフィカルユーザインタフェースも用意されている。メールでの入力が可能だが、多くの項目が決められており、それに従って入力しなければならない。

- Request Tracker [5]

Perl で記述されたバグトラッキングシステムである。バックエンドに MySQL⁴[15], PostgreSQL⁵[16], Oracle⁶[17] のいずれかのデータベースサーバを用いる。メールとウェブインタフェースにより、チケットへの記入ができる。

- JitterBug [6]

ウェブベースのバグトラッキングシステムである。C 言語で記述されており、CGI⁷として動作する。単一のプログラムである。GNU GPL によってライセンスされ、配布されている。メールによるバグ報告も可能。HTML⁸3.2 に準拠したウェブページを生成する。バックエンドにデータベースサーバは用いない。

³GNU GENERAL PUBLIC LICENSE

⁴高速性と堅牢性を追及したマルチユーザ・マルチスレッドの SQL データベースシステム。

⁵POSTGRES, Version 4.2 をベースにしたオブジェクトリレーショナルデータベース管理システム。POSTGRES はカリフォルニア大学バークレー校のコンピュータサイエンス学科で開発された。

⁶Oracle Corporation の製品である、リレーショナルデータベース管理システム。

⁷Common Gateway Interface の略。ウェブサーバが、ウェブブラウザからの要求に応じて外部プログラムを起動するための仕組み。

⁸HyperText Markup Language の略。文書の論理的構造を記述するためのマークアップ言語。ウェブページを記述するのに使われる。

- Bugzilla Project [7]

Perl で記述されたバグトラッキングシステムである。Mozilla Public License [18] に基づいて配布されている。実際に利用されている例として mozilla.org Bugzilla [12] が挙げられる。バックエンドに MySQL を用いている。バグの間の依存性を示すことができ、依存性を図示することもできる。メール、XML⁹、コンソール、ウェブ等のインタフェイスを備えている。

- Debian bug tracking software [8]

主に Perl で記述されたバグトラッキングシステムである。GNU GPL によってライセンスされ、配布されている。実際に利用されている例として Debian バグ追跡システム [13] が挙げられる。全ての操作をメールで行なうことができる。最低限の入力項目としてパッケージ名とバージョンを記述するだけで、他は自由文での記述が可能である。また、ウェブインタフェイスによりバグを参照することができる。

- BugCommunicator [9]

Ruby で記述されたバグトラッキングシステムである。日本人の作者によって開発が進められている。GNU GPL によってライセンスされ、配布されている。バックエンドに MySQL を用いる。

- Teacup [10]

ウェブベースのトラブルチケットシステムである。Perl で記述されており、CGI として動作する。GNU GPL によってライセンスされ、配布されている。バックエンドに PostgreSQL を用いる。

⁹eXtensible Markup Language の略。文書の論理的構造を記述するためのマークアップ言語。文書独自のマークアップ方法を定義できる。

3 本研究で提案するシステム

以下では、本研究で提案するトラブルチケットシステムについて説明する。

本研究ではメールのスレッドをチケットとして扱うことを提案する。そこでメールリストを中心としたシステムを構築した。このメールリストに流れるメールを管理者で共有する。そして、障害発生を通知するメールから始まるスレッドを一つのチケットとして扱う。

3.1 チケットの運用

本研究で構築したトラブルチケットシステムでは、以下のようにしてチケットを運用する。

1. OSUNET Monitor によって監視対象の停止が検知された場合、トラブルチケットシステムに停止を通知するメールが送信される。このメールの送信をもってチケットの発行 (Open) とする。以下、このメールをオープンメールと表記する。
2. 作業担当者へのチケットの割り当ては、作業担当者が自ら宣言することで割り当てを行なう。
3. 管理者および作業担当者は、原因の分析、復旧作業等に関する情報をオープンメールへの返信として送信する。
4. 監視対象のマシンもしくはサービスが復旧すると OSUNET Monitor が検知し、トラブルチケットシステムへ復旧を通知するメールが送信される。このメールはオープンメールへの返信として送信される。このメールの送信をもってチケットの終了 (Close) とする。

通常、一つの障害について一つのスレッドが発行される。本システムではチケットの発行を OSUNET Monitor だけが行なう。そのため一つの障害に複数のチケットが発行されることはない。

図 1 にチケット運用時のメールの流れを示す。

図 2 にスレッドによるチケット表示の例を示す。図の例では、三つのスレッドが存在する。つまり三つのチケットが存在することになる。そして全てのチケットは、復旧を通知するメールによってクローズされている。

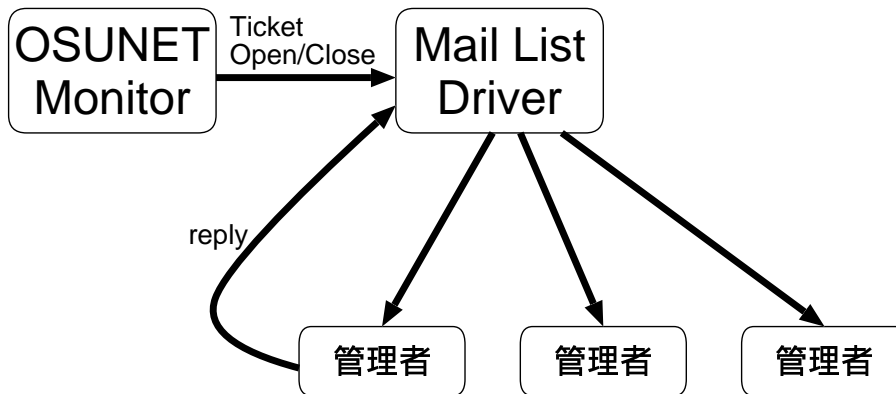


図 1: チケット運用時のメールの流れ

```

kterm
74 11/26(火) 14:30 [ ml@ds5.fken.ise.o ] #00077 NetMaster: SST Report <10382
75 11/26(火) 14:42 [ ml@ds5.fken.ise.o ] #00078 NetMaster: SST Report <103
76 11/26(火) 18:20 [ ml@ds5.fken.ise.o ] #00080 NetMaster: SST Report <10383
81 11/26(火) 18:35 [ AKAMATSU Kosuke <
82 11/26(火) 20:08 [ ml@ds5.fken.ise.o ] #00085 NetMaster: SST Report <103
77 11/26(火) 18:18 [ ml@ds5.fken.ise.o ] #00079 NetMaster: SST Report <10383
78 11/26(火) 18:30 [ ml@ds5.fken.ise.o ] #00081 NetMaster: SST Report <103
371 11/29(金) 13:36 [ AKAMATSU Kosuke <
-かなな-EE=:%%-F1 [ON] Wanderlust: +fken/tts-test {T} (0 new/0 unread) (SU
To: tts-test@ds5.fken.ise.osaka-sandai.ac.jp
Subject: [tts-test:00085] NetMaster: SST Report <1038308702.ds5.fken.http@warbi
rd.ise.osaka-sandai.ac.jp>
From: ml@ds5.fken.ise.osaka-sandai.ac.jp
Reply-To: tts-test@ds5.fken.ise.osaka-sandai.ac.jp
Date: Tue, 26 Nov 2002 20:08:48 +0900
Message-Id: <1038308702.ds5.fken.http@warbird.ise.osaka-sandai.ac.jp>
ds5.fken.http _Up_ at Tue Nov 26 20:08:48 JST 2002
  
```

図 2: スレッドによるチケット表示の例

3.2 メール

電子メールのメッセージはメッセージヘッダとボディから成る。メッセージヘッダとボディは空行によって区切られる。

電子メールの形式については RFC2822 [19] で定められている。

3.2.1 メッセージヘッダ

メッセージヘッダには以下のようなフィールドが含まれる。

- Message-ID フィールド

メッセージの識別子 (Message-ID) を記述する。各メッセージの Message-ID はそれぞれユニーク¹⁰ でなければならない。必須ではないが、すべてのメッセージは Message-ID フィールドを持つべきである。Message-ID は ‘<’ で始まり ‘>’ で終わる。一般的な Message-ID の形式は “<(ホストにおいてユニークな ID)@(host の FQDN¹¹)>” である。

例 : Message-ID: <20021209191606.qmail@ds5.fken.ise.osaka-sandai.ac.jp>

- In-Reply-To フィールド

メッセージが他のメッセージへの返信である場合に付けられる。メッセージ A に対して返信する場合、メッセージ A (以下、親メッセージ) の Message-ID が含まれる。必須ではないが、返信時には適切な In-Reply-To フィールドを付けるべきである。

- References フィールド

メッセージが他のメッセージへの返信である場合に付けられる。親メッセージの References フィールドの内容と、親メッセージの Message-ID が続けて含まれる。もし親メッセージに References フィールドが含まれておらず、Message-ID を一つだけ含む In-Reply-To フィールドがあれば代わりに In-Reply-To フィールドの内容が用いられる。

¹⁰唯一無二。同じものは他に存在しないという意味。

¹¹Fully Qualified Domain Name. ホスト名に対して使う場合は、ホスト名およびドメイン名部分の文字列を省略せずにすべて記述する表記方法。

3.2.2 スレッド

スレッド (Thread) とは「(話・議論の) 筋道、脈絡」という意味で、メールを話の筋道に従ってツリー状に並べたものである。メールにおいてはヘッダの Message-ID フィールド・In-Reply-To フィールド・References フィールドを用いてスレッドを生成する。これはメールを表示する時に MUA¹² によって行なわれる。

スレッド形式で表示することにより、メールによる議論をたどりやすくなる。図 3 にスレッドの例を示す。ここでは一番左の数字が 631 から 640 までのメールが一つのスレッドである。

```
kterm
629 10/17(木)21:40  L[ OHGAKI Hitoshi An ] #40622 fken.a4w:40622: Re: Good
630 10/17(木)21:42  [ OHGAKI Hitoshi An ] #40620 fken.a4w:40620: Week 42, DS9
631 10/18(金)14:48  [ OHGAKI Hitoshi An ] #40623 fken.a4w:40623: Books
632 10/18(金)16:00  L[ AKAMATSU Kosuke < ] #40624 fken.a4w:40624: Re: Books
633 10/18(金)17:57  L[ OHGAKI Hitoshi An ] #40625 fken.a4w:40625: Re: Book
634 10/18(金)22:23  L[ SUGITA Toshinori ] #40626 fken.a4w:40626: Re: Bo
635 10/19(土)09:25  L[ OHGAKI Hitoshi An ] #40627 fken.a4w:40627: Re:
636 10/19(土)13:49  L[ SUGITA Toshinori ] #40630 fken.a4w:40630: Re
637 10/19(土)19:47  L[ OHGAKI Hitoshi An ] #40631 fken.a4w:40631:
638 10/19(土)20:22  L[ SUGITA Toshinori ] #40632 fken.a4w:40632
639 10/19(土)10:30  L[ NAKAJI Takahiro < ] #40628 fken.a4w:40628: Re: Bo
640 10/19(土)13:08  L[ OHGAKI Hitoshi An ] #40629 fken.a4w:40629: Re:
641 10/21(月)10:34  [ andrew@osaka-sand ] #40633 fken.a4w:40633: Good mornin?
-かな-EE=:??-F1  [ON] Wanderlust: +fken/a4w.ml {T} (0 new/0 unread) (Summ
To: fken.a4w@h-ps006.ise.osaka-sandai.ac.jp
Subject: fken.a4w:40624: Re: Books
From: AKAMATSU Kosuke <akamatsu@h-ps011.ise.osaka-sandai.ac.jp>
Reply-To: fken.a4w@h-ps006.ise.osaka-sandai.ac.jp
X-Beat: Fri Oct 18 @342 2002
X-StarDate: [-30]9241.50
Date: Fri, 18 Oct 2002 16:00:01 +0900
Message-ID: <m3u1jkfczy.wl@yaa.dtdns.net>
User-Agent: Wanderlust/1.1.1 (Purple Rain) SEMI/1.13.7 (栗津) CLIME/1.13.6
(中ノ庄) Emacs/20.7 (i386-vine-linux-gnu) MULE/4.1 (葵)

赤松です。

At Fri, 18 Oct 2002 14:48:16 +0900,
OHGAKI Hitoshi Andrew <andrew@osaka-sandai.ac.jp> wrote:
```

図 3: スレッドの例

3.3 メールリスト

メールリストとは、専用のメールアドレスにメールを送ると、そのメールがメールリストのメンバー全員に配信されるシステムである。メールを受信し、それをメンバーに対して送信するソフトウェアをメールリストドライバという。

¹²Mail/Message User Agent. メールを読み書きするためのソフトウェア。メールクライアントや、メイラと呼ばれる。

3.4 他のシステムとの連携

既存のソフトウェアを利用したり自作のフィルタを利用することで、ネットニュースクライアントやウェブブラウザを用いてシステムへアクセスできるようになる。本研究では実際に以下のように連携できることを確認した。

- ネットニュースシステムとの連携
 - Mail2News [20] によってネットニュースへの変換を行なった。これによってネットニュースクライアントを用いてチケットを読めるようになった。(図 4)
 - 自作のフィルタによってヘッダを書き換え、ネットニュースクライアントを用いてチケットに書き込めるようにした。(図 5)
- ウェブシステムとの連携
 - MHonArc [21] によってウェブコンテンツへの変換を行なった。これによってウェブブラウザを用いてチケットを読めるようになった。(図 4)
 - Ruby を用いてメール送信 CGI を作成し、ウェブブラウザからチケットに書き込めるようにした。(図 5)

システム内でチケットの整合性を保つために、入力全てはメールリストドライバを経由して行うこととした。

図 4 に本システムからネットニュース・ウェブへの出力の流れを示す。図 5 にネットニュース・ウェブから本システムへの入力の流れを示す。

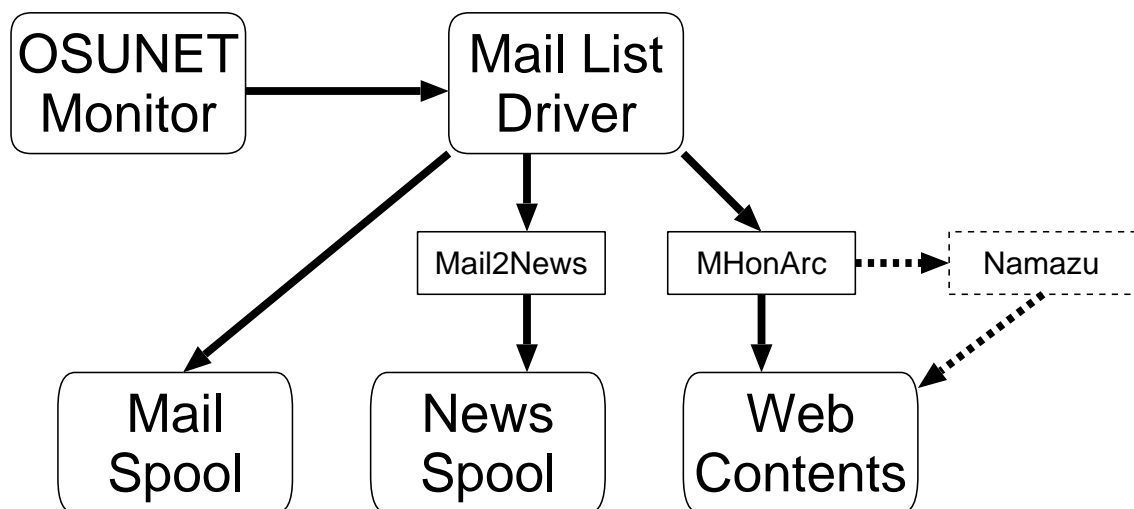


図 4: ネットニュース・ウェブへの出力

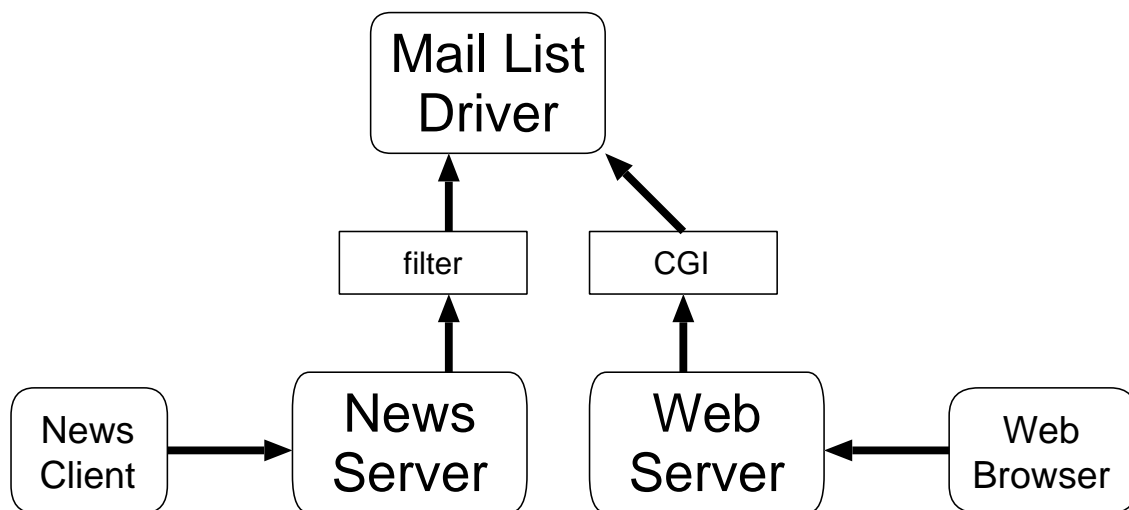


図 5: ネットニュース・ウェブからの入力

3.5 実装

システムの構築には Debian/GNU Linux をインストールしたマシンを用いた。本システムの最小構成は MTA とメーリングリストドライバの組み合わせである。MTA¹³として qmail [22] を用いた。メーリングリストドライバとして CML [23] を用いた。

さらに、ネットニュースシステムとウェブシステムとの連携を行なった。ネットニュースサーバとして INN¹⁴[24] を用いた。ウェブサーバとして Apache [25] を用いた。メールからネットニュースへの変換に Mail2News を用いた。メールからウェブコンテンツへの変換に MHonArc を用いた。ネットニュースからメールへの変換のためにフィルタを作成した。ウェブからメールを送信するために CGI を作成した。

以上のように構築を行ない、実際にトラブルチケットシステムとして稼働することを確認した。

¹³Mail/Message Transfer Agent. メーリングシステムにおいて、メールの配送を担当するソフトウェア。

¹⁴Internet Net News の略。

4 考察

トラブルチケットシステムを構築し、実際に動作させてみた。本システムについて以下のような利点と問題点が挙げられる。

4.1 利点

本システムでは既存のメールシステムを利用した。これは RFC¹⁵によって標準化されている。そのためウェブ・ネットニュースなどの他のシステムとの親和性が高く、拡張性が高い。本研究でウェブ・ネットニュースシステムと容易に連携できることを確認した。

このシステムではチケットへの記入には、メールを用いている。そのため自由文の入力が可能である。これにより管理者は項目を意識することなくチケットへ情報を入力することが可能となる。そして、MUA という使いなれたツールを利用してチケットを扱うことができる。そのため管理者にとって、利用時のオーバヘッドが小さいと考えられる。

チケットがクローズされた後もスレッドを利用して議論を続けることが可能である。例えば図 2 の例において、一番下のスレッド(チケット)は復旧を通知するメールによってクローズされている。しかし、チケットがクローズされた後もそのスレッドで議論が続いている。

GNATS 等の多くのシステムでは日本語を扱うことができない。日本語を扱うためにはソフトウェアの変更が必要となる。Bugzilla や JitterBug 等、一部のシステムは日本語化されたものが公開されている。しかし本システムは言語に依存せず、日本語を扱うことができる。

4.2 問題点

まず、作業担当者の有効な割当方法がないことが挙げられる。本システムでは作業担当者が自ら宣言することで割当を行なうこととした。

また、過去のチケットの検索機能がないことが挙げられる。これについては Namazu¹⁶[26]を利用することで実現できる。

本システムでは管理者の能力に依存する点がある。まず、本システムではチケットの内容は自由文で記述される。つまり、書式を固定していない。そのため入力内容が記入者の能力に依存する。また、メールを利用しているので、他のメールと混ざる可能性がある。これは既存のツールを用いて適切に設定を行なうことで避けられる。この点も管理者の能力に依存する。

メールシステムの性質上、メールの順番が前後してしまうことがある。メール A に対してメール B を返信した場合、メール A よりもメール B の方が先に到着することがある。その場合、MUA によってはスレッド表示が切れてしまう。多くの MUA においては、並び換え等を行なうことによって正しく表示することができる。しかし手間がかかる。

¹⁵Request For Comment の略。インタネットに関する技術情報や仕様などを定めた文書。現在は IETF(Internet Engineering Task Force) が管理している。

¹⁶日本語全文検索システム

本システムでは管理者の利用する MUA の機能がシステムの一部となる。ゆえに利用できない MUA も存在する。例えばまれに、ヘッダに Message-ID フィールド・In-Reply-To フィールド・References フィールドを付けない MUA が存在する。また、ユニークでない Message-ID フィールドを付加する MUA も存在する。本システムは、そのような MUA からは利用できない。中には、正しく設定を行なうことによりユニークな Message-ID フィールドを付加するよう設定できるものもある。それらについては正しく設定を行なうことで利用できるようになる。

5 まとめ

本研究では、OSUNET Monitor と既存のメールシステムを組み合わせ、トラブルチケットシステムを構築した。本システムではチケットとしてメールのスレッドを利用した。これにより、管理者にとって使いなれたメールを用いてチケットを運用することが可能となった。またウェブ・ネットニュースシステムと容易に連携できることを確認した。

今後、このシステムを利用して fken-net¹⁷ の管理を行ない、運用上の問題点を洗い出し、改良を加えていく予定である。管理者の支援機能として、同様の障害に関する過去のチケットの検索機能が考えられる。チケットが発行されると自動的に検索を行ない、結果をスレッドに加えることも考えられる。また、より大きな規模のネットワークで利用できるように、作業担当者の割当方法等についても検討する必要がある。

¹⁷情報安全工学研究室内のローカルネットワーク。

謝辞

本研究を進めていく上で、御指導及び御協力を頂いた大垣 斉講師・藤井 信夫教授・中村 孝講師・fken.a4w メールリストのメンバーの方々、そして情報システム工学科の卒研究生の方々に深く感謝の意を表します。

参考文献

- [1] OSUNET Monitor, [<http://www-fken.ise.osaka-sandai.ac.jp/SST/>]
- [2] RFC792, Internet Control Message Protocol
- [3] 影舞, [<http://www.daifukuya.com/kagemai/>]
- [4] GNATS, [<http://www.gnu.org/software/gnats/>]
- [5] Request Tracker, [<http://www.bestpractical.com/rt/>]
- [6] JitterBug, [<http://samba.anu.edu.au/jitterbug/>]
- [7] Bugzilla Project, [<http://www.mozilla.org/projects/bugzilla/>]
- [8] Debian bug tracking software, [<http://www.chiark.greenend.org.uk/ian/debbugs/>]
- [9] BugCommunicator, [<http://www.nongnu.org/pupa/bugcomm.html>]
- [10] Teacup, [<http://www.altara.org/teacup.html>]
- [11] NetBSD GNATS Bug Database Summary, [<http://www.netbsd.org/Gnats/>]
- [12] mozilla.org Bugzilla, [<http://bugzilla.mozilla.org/>]
- [13] Debian バグ追跡システム, [<http://www.debian.org/Bugs/>]
- [14] GNU GENERAL PUBLIC LICENSE, [<http://www.gnu.org/licenses/gpl.txt>]
- [15] MySQL, [<http://www.mysql.com/>]
- [16] PostgreSQL, [<http://www.postgresql.org/>]
- [17] Oracle Corporation, [<http://www.oracle.com/>]
- [18] Mozilla & Netscape Public Licenses, [<http://www.mozilla.org/MPL/>]
- [19] RFC2822, Internet Message Format
- [20] Mail2News Mini-Howto,
[<http://osaka-sandai.www.linux.or.jp/JF/JFdocs/archive/Mail2News.html>]

- [21] MHonArc, [<http://www.mhonarc.org/>]
- [22] qmail: Second most popular MTA on the Internet, [<http://www.qmail.org/top.html>]
- [23] CML, [<http://www.wkj.forus.or.jp/CML/>]
- [24] INN: InterNetNews, [<http://www.isc.org/products/INN/>]
- [25] The Apache HTTP Server Project, [<http://httpd.apache.org/>]
- [26] 全文検索システム Namazu, [<http://www.namazu.org/>]

※ URL はすべて 2003 年 2 月 1 日時点のもの

付録

ネットニュースヘッダ加工フィルタ

```
grep -v '^Newsgroups:' | grep -v '^NNTP-Posting-Date:' \  
| grep -v '^Followup-[Tt]o:' | grep -v '^Xref:' \  
| grep -v '^NNTP-Posting-Host:' | grep -v '^Path:'
```

メール送信 CGI

```
— input.rb begin —  
#!/usr/bin/env ruby  
  
to = 'tts-test@ds5.fken.ise.osaka-sandai.ac.jp'  
sendmail = '/usr/sbin/sendmail'  
nkf = '/usr/bin/nkf'  
  
require "cgi-lib"  
cgi = CGI.new  
  
print "Content-type: text/html\n\n"  
print %Q[<html>  
<head>  
<title>test</title>  
</head>  
<body>  
]  
  
if (cgi["mid"] =~ /^<.*@.*>$/) or (cgi["mid"] == "") then  
  
print cgi["mid"]  
print "<br>\n"  
print cgi["from"]  
print "<br>\n"  
print cgi["subject"]  
print "<br>\n"  
print cgi["mailbody"]  
  
open("| #{nkf} -j -Lu | #{sendmail} -t -i", "w") do |fh|  
  fh.print <<"HEADER_END"  
From: #{cgi["from"]}  
To: #{to}
```

```
Subject: #{cgi["subject"]}
HEADER_END
```

```
  if cgi["mid"]!="" then
    fh.print "References: #{cgi["mid"]}", "\n"
    fh.print "In-Reply-To: #{cgi["mid"]}", "\n"
  end
```

```
  fh.print <<"HEADER_END"
```

```
#{cgi["mailbody"]}
HEADER_END
```

```
  end
else
  print "error."
end
```

```
print %Q[</body>
</html>]
— input.rb end —
```

— input.html begin —

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html lang="ja">
<head>
<link rev="made" href="mailto:akamatsu@h-ps011.ise.osaka-sandai.ac.jp">
<link rel="index" href="."/>
<link rel="stylesheet" type="text/css" href="./yaa.css">
<meta http-equiv="Content-Type" content="text/html ;charset=ISO-2022-JP">
<title>Input Form</title>
</head>

<body>

<h1>input from web</h1>

<form method="POST" action="./a.cgi">
Message-Id : <input type="text" name="mid"><br>
From : <input type="text" name="from"><br>
Subject : <input type="text" name="subject"><br>
mailbody : <textarea name="mailbody"></textarea><br>
<input type="submit" value="send">
</form>

<hr>
<address>
Last modified: 2002-09-05T15:36:00+09:00
<br>
<a href="mailto:akamatsu@h-ps011.ise.osaka-sandai.ac.jp">
akamatsu@h-ps011.ise.osaka-sandai.ac.jp
</a>
</address>

</body>
</html>
— input.html end —
```