

# 特別研究報告

## 題目

**Web 共同編集機能を実装したメール検索システム**

## 指導教員

井上 克郎 教授

## 報告者

Choy Kho Yee

平成 18 年 2 月 20 日

大阪大学 基礎工学部 情報科学科

## Web 共同編集機能を実装したメール検索システム

Choy Kho Yee

### 内容梗概

近年では，ソースコードなどが広く公開され，誰でも自由に開発に参加できるオープンソースソフトウェア開発手法が注目を集めている．開発者とユーザ間での意思疎通のために，メーリングリストシステムが利用される．メーリングリストに送信された電子メールは全てメーリングリストアーカイブに保存されており，誰でも参照できるように公開されている．

しかし，現状のメーリングリストのアーカイブに参照する方法は，基本的なキーワード検索機能と閲覧機能を利用した方法のみである．これらの機能を提供するシステムはユーザからのフィードバックを許さないため，メールが提供している情報の品質は一定であり，同じ疑問を持ったユーザ間で経験を共有することもできない．

一方，最近では，インターネットなどを通して利用者が共同に編集，レビューを行うことで質の高い文章の作成を目指すプロジェクトが普及し始めている．この Web 共同編集という概念を利用すればメーリングリストアーカイブの問題点が解決できると考えられる．

そこで，本研究では，Web 共同編集機能を実装したメール検索システムの試作を行った．本システムはメーリングリストアーカイブに蓄積された電子メールを，利用者が編集，評価及び分類できる機能を提供する．これらの操作によって，メーリングリストアーカイブの検索結果の精度を上げることと，提供する情報の品質を改善することを目指す．

また，実際のオープンソースソフトウェア開発で用いられたメーリングリストアーカイブを使って，評価を行った．その結果，本システムを用いることで，メーリングリストアーカイブの利用者が，有益な情報を効率よく入手することが可能となり，現状の問題点が改善されることを確認した．

### 主な用語

オープンソースソフトウェア開発 (Open Source Software Development)

メーリングリストアーカイブ (Mailing List Archive)

## 目次

<b>1</b>	<b>はじめに</b>	<b>4</b>
<b>2</b>	<b>オープンソースソフトウェアとその開発環境</b>	<b>6</b>
2.1	オープンソースソフトウェア	6
2.2	オープンソースソフトウェア開発環境	6
<b>3</b>	<b>メーリングリストアーカイブの問題点</b>	<b>10</b>
3.1	電子メールの仕様	10
3.2	メーリングリストアーカイブの問題点	12
3.3	既存参照手法の問題点	13
<b>4</b>	<b>システムの設計</b>	<b>15</b>
4.1	概略	15
4.2	バックエンド	16
4.3	フロントエンド	16
<b>5</b>	<b>システムの実装</b>	<b>18</b>
5.1	開発環境と実装言語について	18
5.2	検索対象	18
5.3	バックエンドの実装	18
5.4	検索機能	27
5.5	Web 共同編集機能	29
5.6	ユーザインターフェイス	30
<b>6</b>	<b>利用例</b>	<b>38</b>
6.1	操作の流れ	38
6.2	実例	38
<b>7</b>	<b>評価</b>	<b>41</b>
7.1	システムの処理速度の評価	41
7.2	再現率，適合率及び $f$ 値の測定	43
7.3	アンケート	44
<b>8</b>	<b>まとめ</b>	<b>46</b>

謝辞	47
参考文献	48
付録	51
A データベースで用いるテーブルの構成	52
B アンケートの質問	56
C アンケートの集計結果	60

## 1 はじめに

ソフトウェア開発の規模の増大につれて、その開発形態が多人数化、分散化している。一つの大規模なソフトウェアの開発を行う際、複数の開発者が、互いにソースコードを共有しながら同時に一つの開発作業に携わることが一般的になりつつある。また、インターネットやイントラネット等のネットワーク環境の発展に伴い、各開発者が国境を超えてそれぞれ異なる場所で作業を行うことも多くなってきた。

その中に、このような分散した開発手法を代表するのは近年注目されてきたオープンソースソフトウェア開発手法である。[1] オープンソースソフトウェア開発では、最終のプロダクトだけではなく、開発過程上に作られた設計情報やソースコード、ドキュメントなどもインターネット上に公開され、誰でも自由に開発に参加することができる。全ての情報が公開されているため、開発者だけではなく、興味と能力を持ったユーザであればソフトウェアの開発に関わることができる。実際のところ、ユーザが開発者にソフトウェアのパッチを提供することも珍しくない。[1][2][3]

オープンソースソフトウェア開発では、開発者とユーザ間での意思疎通や質疑応答にメーリングリストシステムが使われる。開発者とユーザが送信した電子メールは全てメーリングリストアーカイブに保存されており、公開されている。このアーカイブから、開発者は過去に行われた開発に関する議論を知り、ユーザはプロダクトに関する疑問に対する解答を入手できる。

ところが、蓄積された膨大な情報の中から、必要な情報を的確に取得することは容易ではない。なぜなら、既存の参照方法が限られているからである。メーリングリストのアーカイブに対する既存の参照方法は、基本的なキーワード検索機能と閲覧機能を利用した方法のみである。そのため、利用者が必要な情報を得るまでに、何回も異なるキーワードで検索を繰り返して、リストアップされた多くの結果から欲しい情報を選別しなければならない。また、メーリングリストアーカイブの既存の検索システムや閲覧システムはユーザからのフィードバックを許さないため、検索結果の精度の向上やメールが提供している情報の品質の改善、同じ疑問を持ったユーザ間で経験を共有することもできない。

一方、近年では多数の利用者が Web を利用した共同編集作業で質の高い文章の作成が注目を集めている。共同編集作業では、ネットワーク上のどこからでも、いつでも、誰でも Web サーバ上にある文章を編集し保存することができる特徴を持つ。この手法を用いて実際に質の高い文章の作成に成功した例は数多く存在している。

そこで、本研究ではメールの内容の品質と検索結果の精度を向上させるために、Web 共同編集機能を実装したメール検索システムの試作を行った。ここで Web 共同編集機能とは、従来のシステムでは不可能だったユーザによる電子メールの編集、評価、分類及び電子メー

ルに対するコメントの追加などの操作を Web インターフェイスを通して行う機能である。

また、FreeBSD の開発で用いられるメーリングリストアーカイブを用いて本システムの評価を行った。まずにメーリングリストアーカイブからデータベースを作成するのにかかる時間と検索にかかる時間を測定した。その結果、十分に実用的な時間で動作し、データ量が増え続けるメーリングリストアーカイブに追隨できることを確認した。次に、既存の全文検索システムの検索結果との比較を行い、ユーザのフィードバックがあまりない初期状態でも同じ程度の情報を提供することができることを確認した。また、システムを FreeBSD のユーザに公開し、アンケートの採取を行った。このアンケートを通して今後の研究の方向を確認することができた。

本システムを用いて、メーリングリストアーカイブの利用者に正確な情報を効率よく届けることができると考えられる。開発者ならより時間を有効に利用して開発作業を行うことができ、ユーザなら必要な情報を早く入手できると考えられる。

以降、2 節では、オープンソースソフトウェアとその開発環境について述べる。3 節では、メーリングリストアーカイブと既存の参照手法が抱えている問題点を説明する。4 節では、システム的设计について、5 節では、その実装について説明する。6 節では、本システムを利用する際の操作の流れ及び本システムを用いた場合に解決する問題の例を示す。そして、7 節では、システムに対して評価を行う。最後に、8 節では、本研究のまとめと今後の課題について述べる。

## 2 オープンソースソフトウェアとその開発環境

本節では，オープンソースソフトウェアとその開発環境について述べる．

### 2.1 オープンソースソフトウェア

ソフトウェア開発のプロダクトであるソースコードやドキュメント等を，インターネットに代表される大規模ネットワークを利用して広く公開し，世界中に分散した複数の開発者がそれらを共有しながら並列的にソフトウェアの開発作業を行う開発手法はオープンソースソフトウェア開発と呼ばれる [1][5]．そのため，インターネットなどのネットワークにアクセスすることができれば，開発者はいつでも自由に開発作業に参加することが可能である．

近年，オープンソースソフトウェア開発手法で，高品質で多機能なソフトウェアを開発できるとして注目を集めている．そのオープンソースソフトウェア開発によって開発されたソフトウェアをオープンソースソフトウェアと言う．FreeBSD[6] や Linux[7]，Mozilla[8]，Apache[9]，OpenOffice.org[10] 等は，オープンソースソフトウェア開発で開発されたオープンソースソフトウェアである．

### 2.2 オープンソースソフトウェア開発環境

オープンソースソフトウェア開発では，各開発者がそれぞれ分散して並列的に開発作業を行うことが可能である．その一方で，開発中のソースコードやドキュメントなどのプロダクトを広く公開するため，それらの管理を行う必要がある．そこで，オープンソースソフトウェア開発に参加する開発者は，オープンソースソフトウェア開発環境と呼ばれる環境の中でプロダクトの管理を行う．

オープンソースソフトウェア開発環境の構成例を図 1 に示す．オープンソースソフトウェア開発環境は，一般に複数の既存システムから構成される．図 1 の構成例の場合，ソースコードやドキュメント等のプロダクトは，版管理システム [11] の一つである CVS(Concurrent Versions System)[12][13][14][15] を用いて管理される．それらのプロダクトは，ある程度安定したと思われれば Web や FTP サーバなどにリリースされ，公開される．一般ユーザは，CVS リポジトリから最新版を入手することもできるし，Web または FTP サーバにリリースされた安定版をダウンロードすることもできる．また，開発者と一般ユーザの間で相互に行われる意思疎通の手段として，電子メールやメーリングリストが用いられる．その内容はアーカイブとして保存され，Web を用いた検索エンジンによって自由に検索や閲覧が可能である．ユーザからのバグ報告等のフィードバックは，バグ追跡システムである Bugzilla によって管理される．また，図 1 には示していないが，最近では，オープンソースソフトウェ

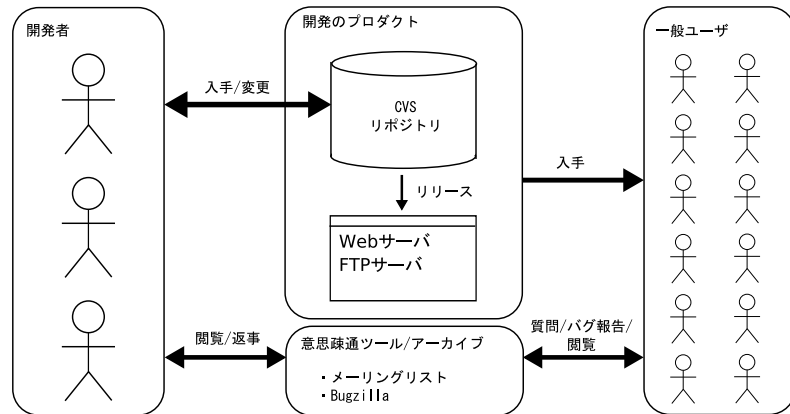


図 1: オープンソースソフトウェア開発環境の構成例

アのドキュメントの作成に共同編集作業を支援するシステムの一つである Wiki を導入する傾向がある。Wiki については後節で説明する。

以下では、電子メールを用いたメーリングリスト及び Wiki について説明する。

#### 電子メールとメーリングリスト

電子メールとは、インターネットやイントラネット等のネットワークを通じて、文章や画像などのデータをやりとりするためのシステムである。電子メールは、インターネットが現れる前に既に存在しており、現在のインターネットの前身である ARPANET[16] の出現によって普及し始め、インターネットの普及とともに爆発的に普及した。今日では、ネットワーク環境の充実に伴い、容易に利用することが可能となり、単に「メール」と称されることも多い。

オープンソースソフトウェア開発では、開発者が世界中に分散して存在し、開発作業を行うことが多い。そのため、互いの意思疎通のための手段として、インターネットを介した電子メールが一般的に利用される。また、ユーザと開発者の間でのやりとりも、電子メールを通して行われることが多い。

しかし、不特定多数の開発者やユーザの一人一人にメールを送ることは不可能である。また、電子メールだけを利用すれば、送られた電子メールの集中管理はできない。従って、電子メールのやりとりを一括して管理するために、メーリングリストと呼ばれるシステムが利用されることが多い。メーリングリストへの参加、退出は一般的に自由である。ある参加者がメーリングリスト宛に電子メールを送信すると、同じ電子メールが参加者全員に配信される。また、誰かがその電子メールに対して返信すると、その電子メールも参加者全員に配



信される。配信と同時に、送られたメールは全てアーカイブとして保存され、管理される。オープンソースソフトウェア開発の場合、このアーカイブはインターネット上に公開されることが多い。このため、他の開発者間での現在と過去の議論内容を、各開発者とユーザが容易に捕捉することが可能となる。また、Web を通じて、アーカイブの中から電子メールによる議論の内容を検索するシステムも存在する。

これらのシステムを利用することにより、分散している開発者間で様々な情報を共有することが可能となり、開発作業の促進につながる。

## Wiki システム

Wiki あるいは WikiWiki とは、Web ブラウザを利用して Web サーバ上のハイパーテキスト文書を書き換えるシステムの一つである。[4] 一般的に、このシステムに使われるソフトウェア自体や、このシステムを利用して作成された文書群自体を指して Wiki と呼ぶ。混乱を起さないために、ここではシステムを構成するソフトウェアを Wiki システムと呼び、作成された文書を Wiki ページと呼ぶ。

Wiki システムは、誰でも、ネットワーク上のどこからでも、文書の書き換えができる特徴を持ち、共同作業で文書の作成に有効である。ソフトウェアとしては、初めに登場した WikiWikiWeb[17] に改良を加えたり参考にしたりして、現在では、オープンソースソフトウェアである MediaWiki[18]、Twiki[19] などの多くの Wiki システムが存在する。これらの Wiki システムの共通点は以下に列挙した。

- ネットワーク上のどこからでも、いつでも、誰でも文書を書き換えて保存することができる。
- 文書の書き換えには Web ブラウザしか必要としない。
- Wiki システム特有の文書マークアップは HTML などと比べて簡単なので覚えやすい。
- 同じ Wiki 内の文書間にリンクが張りやすくなっていて、高度に関連しあった文書群を作成しやすい。

以下、Wiki システム特有の文書マークアップについて簡単に説明する。WikiWiki はハワイ語で「速い」を意味し、ページを速く作成できることを目標とする。従って、従来のように HTML マークアップで Web ページを作成することを放棄して、HTML マークアップより簡単なマークアップで Web ページが作成できるように工夫した。ユーザは Wiki システム固有のマークアップで記述したテキストデータを Web インターフェイスを通して編集する。Web ブラウザで Wiki ページを表示する際に、Wiki システムは、Wiki システムの固有マークアップ

表 1: Wiki システムのマークアップと HTML マークアップとの対応の例

Wiki システム	HTML
//emphasis text//	<em>emphasis text</em>
+++ Level 3 Heading	<h3>Level 3 Heading</h3>
----	<hr>
[http://pear.php.net PEAR]	<a href="http://pear.php.net">PEAR</a>
http://c2.com/sig/wiki.gif	

プから HTML マークアップへの変換を行う。Wiki システムによって使えるマークアップが異なるが、代表的なマークアップはキャメルケース (CamelCase) による他の Wiki ページへのリンクや、表 1 にまとめたものがある。なお、キャメルケースとは、例えば”CamelCase”のように、複合語の各単語の先頭を大文字で書くスタイルのことである。

オープンソースソフトウェアのドキュメントの作成に Wiki システムを利用した例は、SpamAssassin[20](<http://wiki.apache.org/spamassassin/>) や、GCC[21](<http://gcc.gnu.org/wiki>) などがある。

### 3 メーリングリストアーカイブの問題点

オープンソースソフトウェア開発では意思疎通や質疑応答の重要な手段としてメーリングリストが用いられる。開発の方向性から実装の詳細までに関する議論や様々な設定と使い方に関する質疑応答がメーリングリストを通じてなされる。オープンソースソフトウェアの開発者にとってもユーザにとっても、メーリングリストアーカイブの持つ情報は価値の高いものであり、無くてはならないものである。しかし、オープンソースソフトウェアの開発規模が増大するにつれて、メーリングリストアーカイブに蓄積された電子メールの数が膨大になり、必要な情報だけを引き出すことが困難になる。

本節では電子メールの基本仕様について述べ、現在のメーリングリストアーカイブと既存参照手法の問題点を指摘する。

#### 3.1 電子メールの仕様

電子メールの仕様は RFC 2822[23] と、RFC 2045[24] から RFC 2049 までのいくつかの RFC によって定められている。ただし、2001 年に RFC 2822 が公表される前に電子メールの仕様は RFC 822[22] に定義されていた。これらの RFC によると、電子メールは「ヘッダ」と「本文」と呼ばれる 2 つの部分に分けられている。

ヘッダには、送信者、受信者や送信日などの電子メールに関する情報のフィールドから構成されている。各フィールドは名前と値の対からなっている。本研究において特に重要なフィールドは、Message-ID と In-Reply-To フィールドである。その詳細については後に説明する。この他にも、From、To、Subject、Date などのフィールドがある。

一方、本文には、メールの本文が記述されている。ヘッダと本文は、空行によって分けられている。なお、電子メールの例は図 2 に示す。

#### 電子メールとスレッド

メーリングリストを利用して議論を行う際、送信されたメールに対して返信することの繰り返しで議論が成立する。一通のメールに対して複数のメールが返信されるため、メールとそれに対する返信メール間の関係は一对多の関係である。この関係に基づいて、メールとそれに対する返信メール間で線を引くと、メールの返信関係を表す木構造のグラフを得ることができる。

その具体例は図 3 に示す。図中には、一通のメールを一つの四角形で表している。また、矢印は返信先のメールを指す。メール a1、a2、a3 はそれぞれ親メールへの返信メールである。また、メール b1 とメール b2 はそれぞれメール a1 とメール a2 への返信メールである。

<p>From - Tue Jan 31 16:17:40 2006          ... (中略) ...          Message-ID: &lt;43DF0F01.9070201@ics.es.osaka-u.ac.jp&gt;          Date: Tue, 31 Jan 2006 16:17:21 +0900          From: Choy Kho Yee &lt;k-choy@ics.es.osaka-u.ac.jp&gt;          User-Agent: Thunderbird 1.5 (Windows/20051201)          MIME-Version: 1.0          To: k-choy@ics.es.osaka-u.ac.jp          Subject: A Sample of an E-mail          Content-Type: text/plain; charset=ISO-2022-JP          Content-Transfer-Encoding: 7bit</p> <p>Good morning.</p> <p>This is a sample of an e-mail.          An e-mail is made up of two parts, namely the header and the          body. All information about the e-mail go into the header, while          the intended message goes into the body.</p> <p>See you.</p>	<div style="border-left: 1px solid black; border-right: 1px solid black; border-top: 1px solid black; border-bottom: 1px solid black; width: 20px; height: 150px; margin: 0 auto;"></div> <p style="text-align: center;">ヘッダ</p> <div style="border-left: 1px solid black; border-right: 1px solid black; border-top: 1px solid black; border-bottom: 1px solid black; width: 20px; height: 100px; margin: 0 auto;"></div> <p style="text-align: center;">本文</p>
---	--

図 2: 電子メールの例

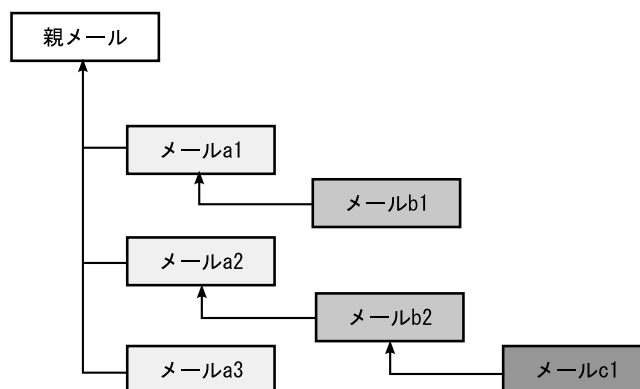


図 3: メール返信関係を表す木構造

最後に、メール c1 はメール b2 に対する返信メールである。これらのメール全体から一つの木が構成される。

このグラフに含まれるメールの集合をスレッドといい、スレッドの木構造をスレッド構造という。議論の流れを把握するためには、スレッドのトップである親メールから、最後まで順に見ていかなければならない。従って、スレッド構造を視覚的にわかりやすく表現することが重要である。

次に、メーリングリストのアーカイブからこのスレッド構造を再現する方法を説明する。前節で述べたように、電子メールはヘッダと本文の二つの部分に分かれている。ヘッダには電子メールに関する色々な情報が記述されていて、その中で、Message-ID と In-Reply-To というフィールドがある。Message-ID とは、各メールに付けられる識別子であり、重複する Message-ID は存在しない。一方、In-Reply-To フィールドには、返信先のメールの Message-ID が記述されている。従って、この二つのフィールドから、スレッド構造の復元は理論上可能である。

しかし、これはあくまでも規格上のことであり、メールを扱うソフトウェアによって実装されていないこともある。例えば、本研究の実験対象とした FreeBSD のメーリングリストのアーカイブには、重複した Message-ID も存在しているし、返信メールであるのに、In-Reply-To フィールドには返信先の Message-ID が記述されていないメールもあった。このような場合には、スレッドの構造を正しく再現することは不可能である。

### 3.2 メーリングリストアーカイブの問題点

メーリングリストアーカイブとは、メーリングリスト宛に送られたメールを全て保存しているデータベースのことを指す。メーリングリストのアーカイブは有益な情報を多く保持しているが、時間が経つにつれて膨大な量のメールから必要な情報を探し出すことが困難になる。メーリングリストのアーカイブには下記の問題点があると思われる。

- メールの内容の品質が保証されない。メーリングリストが公開されて誰でも使えることで便利なシステムではあるが、その一方で、誰からでもメールを受け付けるため、中には提供する情報の品質が低いメールも混ざっている。ただし、ここで情報の品質が低いというのは、その内容が実際の問題解決の参考にならないメールのことを指す。その代表として、「フレーム」と呼ばれるメーリングリストなどのインターネット上の議論の場に起きる過熱的な論争が挙げられる。また、送信された当時では内容が正しかったにもかかわらず、時間の経過とともにその内容が不正確になってしまったようなメールも数多く存在する。このような内容の品質が低いメールは、メーリングリストのアーカイブから有用な情報を引き出す際の大きな障害となる。

- 一通のメールは複数の話題に対応するのが普通だが、既存メーリングリストの管理手法では、その対応関係を再現することが困難である。現在、メールの分類に関して、メーリングリストの管理者が複数の話題に対応したメーリングリストをそれぞれ用意し、利用者は自らの判断で適切なメーリングリストに送信するのが一般である。しかし、メールの内容が複数の話題にまたがる場合、この分類手法ではうまく対応できない。

### 3.3 既存参照手法の問題点

現在、メーリングリストのアーカイブの参照手法は基本的に二つに分かれている。その一つはアーカイブに保存された全てのメールを、送信日、スレッド、送信者、タイトルなどの項目順に閲覧する手法であり、もう一つはキーワードを用いたメーリングリストアーカイブに対する全文検索である。しかし、多くの場合、後者のメール検索という手法が利用される。メーリングリストアーカイブの閲覧は、探したいメールの送信日や送信者などが既にわかっているなどの特殊な場合にのみ使われる。従って、以下では主に全文検索における問題点を述べる。

現在、メーリングリストのアーカイブを検索できる Web サイトは多数存在する。FreeBSD のメーリングリストアーカイブを検索できるサイトだけでも、

- The FreeBSD Project, Search (<http://www.freebsd.org/search/>)
- Rambler: FreeBSD mail archives search (<http://freebsd.rambler.ru/>)

などがある。

これらの検索サイトで主に使われている検索手法は「全文検索」というものである。全文検索とは、文字列で構成されるデータを検索する際に、データの任意の一部分に含まれる特徴を基に検索することである。確かに、全文検索は強力な検索手法であり、検索結果の精度も高い。しかし、3.2 節で述べたメーリングリストアーカイブの問題点のため、メーリングリストアーカイブの検索に全文検索だけを利用すると、理想的な検索精度が実現できない。

また、3.1 節で述べたように、メーリングリストの議論はスレッド単位でまとめられる。議論の流れを追うためには、スレッド構造の再現が極めて重要である。しかし、既存の検索システムの多くは、このスレッド構造の再現がされないものが多い。検索結果がメール単位で表示され、同じスレッドに属しているメールにもかかわらず、何の関係もないように個別に結果として表示される。これでは議論の流れを追うことが難しい。

それに、キーワード検索には一つの大きな欠点がある。それは、キーワードの選び方にある。初心者は、何をキーワードとして検索を行えばいいかさえわからない場合が多いのであ

る．この時，メーリングリストに新たにメールを送信したり，適切でないキーワードを使って検索を行って，少しずつ必要な情報に近づいたりする必要がある．

## 4 システムの設計

本研究ではメーリングリストアーカイブ内のメールの品質改善と検索精度の向上を支援する Web 共同編集機能を実装したメール検索システムの設計と試作を行った。本節では、試作したシステムの設計について説明する。

### 4.1 概略

本システムでは、従来のシステムでは不可能だったユーザによるメールの編集、評価、分類及びメールに対するコメントの追加等の機能を実装し、ユーザがこれらの操作を行うことによって、メーリングリストアーカイブの検索結果の精度を上げることと、提供する情報の品質を改善することを目指す。

本システムは、メーリングリストアーカイブからメールを抽出して検索可能なデータベースを構築するバックエンドと、Web ブラウザを通して表示されるフロントエンドの二つの部分に分かれている。なお、システム全体の概略は図 4 に示す。

システムの動作の流れとして、まずバックエンドでは、登録プログラムがメーリングリストのアーカイブに蓄積されたメールから必要な情報を抽出し、抽出した情報をデータベースに蓄える。ユーザは用意した Web インターフェイスを利用してメールを検索する。必要がある場合、同じ Web インターフェイスを通してデータベース内の情報を更新する。これからバックエンドとフロントエンドの仕様と設計を説明する。

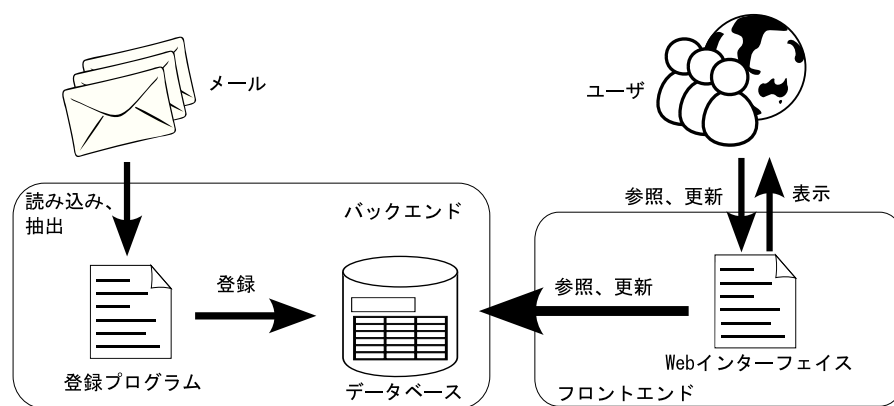


図 4: システムの概略図



## 4.2 バックエンド

バックエンドは、検索・更新処理を簡単にするためのデータベースと、メーリングリストのアーカイブに蓄積されたメールから必要な情報を抽出しデータベースに登録するプログラムから構成される。

登録プログラムを定期的に行うことによって、メーリングリストアーカイブから新たなメールをデータベースに格納することができる。ただし、既に登録したメールは二度と読み込まないようにする。

## 4.3 フロントエンド

ユーザが Web インターフェイスを通して行える操作は以下の通りである。ただし、本システムではメールの編集にあたって 2.2 節で紹介した Wiki システムを導入し、メールと Wiki ページを同じようなものとして扱うことにした。便宜のため、これからはメールと Wiki ページのことをまとめてメール・Wiki と書く。また、タグとはユーザがメール・Wiki に対して任意に付けることのできる文字列である。

- メール・Wiki を検索する
- 既存メール・Wiki を編集する
- 新たに Wiki ページを作成する
- メール・Wiki にランクを付けて評価する
- メール・Wiki にタグを付けて分類する
- メール・Wiki に対してコメントを投稿する

上記の機能に加え、様々な観点からメール・Wiki を絞って表示する機能も提供する。これらの機能は 4 つのインターフェイスを通して提供される。

- メール・Wiki の検索インターフェイス

キーワードの入力フィールドからユーザの入力を受け、データベースを参照して検索を行う。検索結果をスレッド単位にまとめ、スレッドのトップであるメールの一覧を出力する。なお、検索結果の中、ユーザに高くランクされたもの、キーワードに深く関係するものを優先に出力する。そして、検索結果が多い場合、ページごとに分けて出力する。一つのページに表示される結果の数が設定できる。また、データベースの全データではなく、ある期間内に送信・作成されたメール・Wiki のみを検索すること

もできる．タイトルと本文の両方からキーワードを検索するか，タイトルのみから検索を行うことも設定できる．

- メール・Wiki の単体表示インターフェイス

上記の検索結果の一覧から閲覧したいメール・Wiki を選択すればそのメール・Wiki が単独に表示される．メール・Wiki のタイトル，最新バージョンの本文などが表示されるとともに，そのメールが属するスレッド内のメールも表示される．そして，編集されたメール・Wiki の編集履歴と各バージョン間の差分情報も表示される．メールに添付ファイルがあった場合，その添付ファイルへのリンクも表示され，そのリンクを通じてファイルのダウンロードができる．メールにコメントが投稿された場合，そのコメントも本文の下に表示される．メール・Wiki のランクとタグ付けもこのインターフェイスから行える．また，下記のメール・Wiki の作成・編集及びコメント追加インターフェイスへのボタンもここに表示する．

- メール・Wiki の作成・編集とコメント追加インターフェイス

このインターフェイスを通して，既存メール・Wiki を編集することや，コメントを投稿すること，新規に Wiki を作成することができる．なお，強制ではないが，編集者，作成者の名前とメールアドレスを入力することもできる．

- ナビゲーション・サイドバー

常に表示されているナビゲーション・サイドバーから，ユーザは以下の視点からメール・Wiki を絞って表示させることができる．

- よく検索されたキーワード
- 最も多くのメール・Wiki に付けられたタグ名
- 最近追加された Wiki ページ
- 最もランクの高いメール・Wiki
- 最もコメント数の多いメール・Wiki
- 最も多く閲覧されたメール・Wiki
- 最も多く編集されたメール・Wiki
- 最近編集されたメール・Wiki
- 最もよくランクされた回数の多いメール・Wiki

## 5 システムの実装

本節では，前節に設計したシステムの実装について説明する．

### 5.1 開発環境と実装言語について

本システムは，下記の開発環境で開発された．

- CPU: Intel Pentium 4 2.00 GHz
- RAM: 884 MB
- OS: Debian GNU/Linux 3.1(Sarge)
- Web サーバー: Apache 2.0.54-5
- データベース: MySQL 4.0.24
- 実装言語: Perl 5.8.4(バックエンド), PHP 4.3.10(フロントエンド)
- その他: CPAN の Perl モジュール群, PEAR の PHP モジュール群, PECL の xdiff モジュール

本システムの作成に利用した CPAN[29] モジュール群と PEAR[30] モジュール群については表 2 と 3 を参照して下さい．PECL[31] のモジュール xdiff はテキストデータの差分情報を抽出するために用いられた．また，これらのモジュールは更に他のモジュールに依存する可能性があるが，ここではそれらの依存モジュールの説明を省略する．

### 5.2 検索対象

本研究では，検索対象のメーリングリストのアーカイブとして FreeBSD のメーリングリストアーカイブを用いた．ただし，FreeBSD のメーリングリストでは項目ごとに分けているため，開発者と一般ユーザがよく利用される”freebsd-”で始まるリストだけを対象にした [33]．1994 年から 2006 年 1 月 17 日までの期間内のメールからデータベースを構築した．

### 5.3 バックエンドの実装

本節では，システムのバックエンドの実装について説明する．

表 2: CPAN の Perl モジュール群

モジュール名	機能
DBI	Perl スクリプトからデータベースへ接続する際のインターフェイス
Email::Folder	mbox 形式のテキストファイルからメールを一通ずつ抽出し、抽出したメールから各ヘッダフィールドと本文を抽出する
MIME::Parser	メールから添付ファイルを抽出する
Mail::Address	ヘッダの To , From などのフィールドに記述された名前とメールアドレスの分解を行う
Date::EzDate	日付を分析し年月日と時間に分解する
List::Compare	リストの和集合、積集合などを求める
File::Basename	ファイルのフルパスからファイル名を抽出する

表 3: PEAR の PHP モジュール群

モジュール名	機能
PHP_Compat	PHP4 に PHP5 の部分機能を追加する
Text_Wiki	特有のマークアップ [32] に従って記述したテキストデータを HTML へ変換する

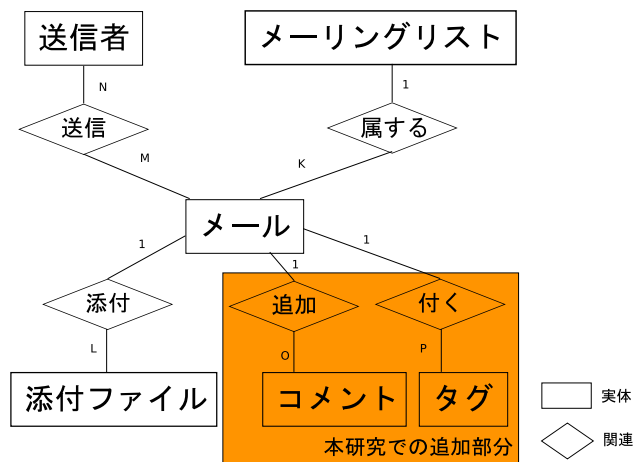


図 5: メールングリストアーカイブの関係を表す図

### メールングリストアーカイブのモデル化

まずに、メールングリストアーカイブをモデル化した。メールングリストアーカイブのモデル図は図 5 に示した。図中の四角形の箱は実体を表し、ひし形の箱は関連を表す。また、ひし形の両辺にある線上の数字、英文字は対応関係 (1 対 1 や 1 対多など) を表す。

メールングリストアーカイブの中心となるのはメールである。送信者はメールを送信し、メールはあるメールングリストに属する。また、メールには添付ファイルが添付されていることもある。これらの実体と関係は、メールングリストアーカイブの既存のものである。本システムの実装のために、新たにコメントとタグという実体を追加した。

図 5 の各実体は属性を持つ。それらの属性は図 6 に示す。図中の四角形の箱は実体を表し、丸は属性を表す。例えば、メールという実体に、Message-Id やタイトル、返信先、本文などの属性がある。ただし、図の中に色付けされたものは本システムの実装のために追加したものである。

### データベースの実装

前節に作成したモデル図はリレーショナルデータベース [25] を用いて実装した。リレーショナルデータベースでは、データはテーブルと呼ばれる表にまとめられ管理される。このようなテーブルが一つのリレーショナルデータベースに複数存在することができる。現在では、リレーショナルデータベースは単にデータベースと呼ばれることが多い。以降、特に明記しない限り、リレーショナルデータベースを単にデータベースと記述する。

本システムの実装にあたって、データベース管理システムには MySQL [28] データベース

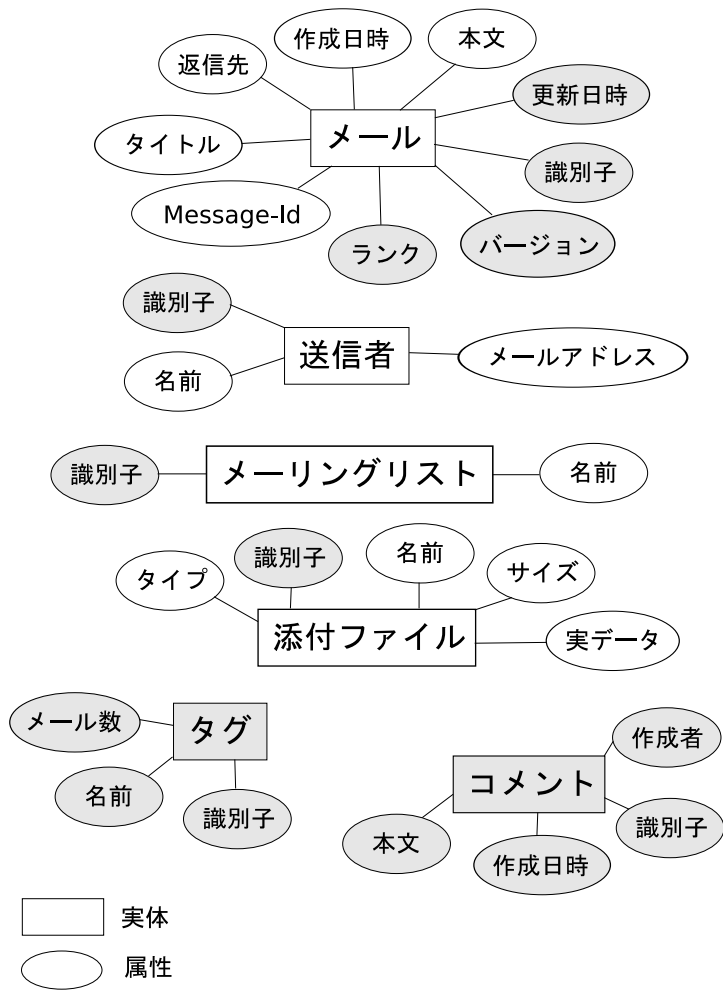


図 6: 各実体の属性

実体テーブル

<b>wiki</b> wiki_id fdate_created time_created last_fdate_updated last_time_updated timestamp_updated timezone subject msg_rank vote_cnt view_cnt comment_cnt fbsd_rel bodytext latest_text latest_version	<b>sender</b> sender_id sender_name rank vote_cnt msg_cnt fav_topic email	<b>mime</b> mime_id wiki_id filename filesize filetype data
	<b>archives</b> wiki_id version old_text fdate_updated time_updated author email	<b>comments</b> id wiki_id comment fdate_created time_created poster email
<b>topic</b> topic_id topic cnt	<b>maillist</b> ml_id ml_name	

関係テーブル

<b>wiki_maillist</b> wiki_id ml_id	<b>wiki_sender</b> wiki_id sender_id	<b>message</b> wiki_id msg_id
<b>reference</b> refer_by refer_to	<b>wiki_topic</b> wiki_id topic_id	

その他のテーブル

<b>topnodes</b> topnode wiki_id	<b>search_results</b> keywords search_type date_range expiry wiki_ids scores	<b>wiki_pages</b> wiki_id page_name
<b>keywords</b> keyword cnt		

図 7: 本システムのデータベースのスキーマ図

管理システムを用いた。メールの情報を蓄えたり、ユーザのフィードバックを保存したりするために、データベースに 16 個のテーブルを設けた。これらのテーブルは図 7 にテーブルの役割別に示した。ここで役割というのは、そのテーブルは実体に対応しているか、関連を表しているかということである。また、実装上の都合で用意したテーブルもあり、必ずしも前節で述べたモデルと対応しているとは限らない。

図中の一つの箱は一つのテーブルに対応する。箱の最上部に書いてあるのはそのテーブルの名前であり、その他はそのテーブルの項目またはカラムの名前である。各テーブルの目的については表 4 を参照して下さい。各カラムの詳細は付録 A にある表 9 から表 24 までの表を参照して下さい。ただし、将来の拡張性を確保するために定義したカラムもあるため、全てのカラムが実装に利用されるとは限らない。

登録プログラムの実装

登録プログラムは Perl[26] で実装され、その名前を mlreg.pl とした。mlreg.pl はメーリングリストのアーカイブに蓄積されたメールから必要な情報を抽出し、データベースに登録するという役割を果たす。なお、ここで「必要な情報」というのは、データベースのテーブル別で表 5 に示す。各カラムの詳細は付録 A にある表 9 から表 24 までの表を参照して下さい。mlreg.pl の起動書式を以下の通りにする。

表 4: データベースの各テーブルの目的

テーブル名	目的
実体テーブル	
wiki	メール・Wiki についての情報を保持する
sender	メール・Wiki の送信者・作成者の情報を保持する
mime	添付ファイルの情報を保持する
archives	メール・Wiki の過去バージョンを保持する
comments	メール・Wiki に付けられたコメントを保持する
topic	ユーザが付けたタグの情報を保持する
maillist	メーリングリストに関する情報を保持する
関係テーブル	
wiki_sender	メール・Wiki とその送信者・作成者との関係を保持する
wiki_topic	メール・Wiki とタグとの関係を保持する
wiki_maillist	メール・Wiki とメーリングリストとの関係を保持する
message	メール・Wiki の Message-ID と内部識別子である wiki_id との対応関係を保持する
reference	メールの返信状況 (スレッド構造) を保持する
その他のテーブル	
topnodes	スレッドのトップノードを早く求めるために各メール・Wiki とそのトップノードとの対応関係を保持する
wiki_pages	新規作成された Wiki ページの情報を保持する
keywords	検索キーワードを保持する
search_results	検索の速度を向上するためのキャッシュである



表 5: 登録プログラムがメールから抽出する必要な情報

テーブル名	カラム
message	msg_id
sender	sender_name , msg_cnt , email
maillist	ml_name
reference	refer_to , refer_by
topnodes	topnode
mime	filename , filesize, filetype, data
wiki	fdate_created, time_created, timezone, subject, bodytext

```
% perl mlreg.pl mailfile
```

引数に複数のメールが入っている mbox 形式のテキストファイルを指定する。mbox 形式のテキストファイルには、複数のメールが連続的に記述されていて、各メールの境界が、一行の空行とそれに続く最初の 5 つの文字が”From ”であることで認識される。このファイルから各メールを抽出して必要な情報をデータベースに登録する。なお、既に登録してあるメールは飛ばされる。これから表 5 の各情報の抽出方法について説明する。

#### 基本データの抽出

msg\_id と subject はそれぞれヘッダの Message-ID フィールドと Subject フィールドから直接入手できる。送信の日時に関する情報(fdate\_created ,time\_created と timezone) は Date フィールドから得られる。メーリングリストの名前である ml\_name の抽出については、FreeBSD のメーリングリストのアーカイブのファイル名に対応するメーリングリストの名前がそのまま記述している特性を利用して、メールからではなく、mbox 形式のテキストファイルの名前から求めて、そのファイル内のメールは全てこのメーリングリストに属すると判断する。また、送信者の名前である sender\_name とその送信者のメールアドレスを表す email はヘッダの From フィールドから取得できる。特定の送信者が送信したメールの数を表す msg\_cnt は、同じ sender\_name から送信されたメールの数をカウントすることで得られる。

この他のデータの取得については、次の 2 項目に分けて説明する。

#### スレッド構造の再現

スレッド構造の再現には、メールのスレッド構造を表す refer\_to と refer\_by を利用する。これらの情報は、メールの In-Reply-To フィールドから得られる。例えば、メール 1 の In-Reply-

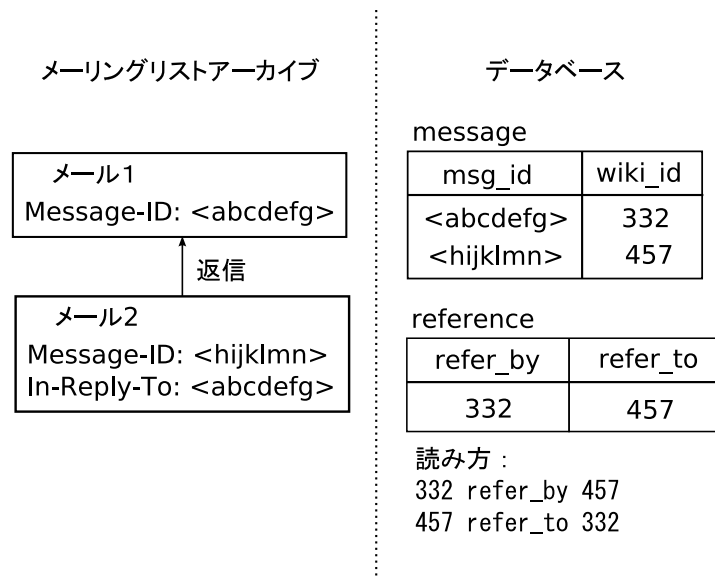


図 8: refer\_to と refer\_by とスレッド構造との関係

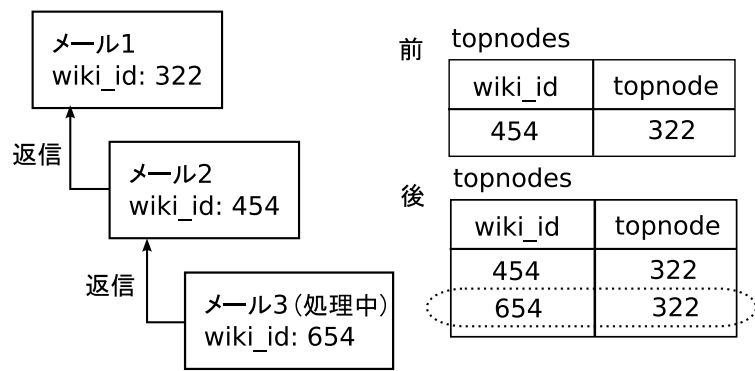
To フィールドにメール 2 の Message-ID が記述された場合，メール 1 の wiki\_id が refer\_by に登録され，メール 2 の wiki\_id が refer\_to に登録される．この例の図的な表現は図 8 である．ただし，wiki\_id は msg\_id をデータベースに登録する際にデータベースによって自動的に生成される．本システムでは，メールを wiki\_id で管理しているため，データベースに存在しない msg\_id を持つメールに関して正しくスレッド構造が再現できないのである．従って，メールの登録は必ず古いものから順に登録しなければならない．

refer\_to と refer\_by だけを用いると，検索を行う際スレッドのトップを求めるにかなり時間がかかってしまうため，各メールに対するスレッドのトップの wiki\_id を topnode に予め記録することによって検索の速度向上を果たした．topnode が下記のアルゴリズムでメールを登録していく過程に更新される．

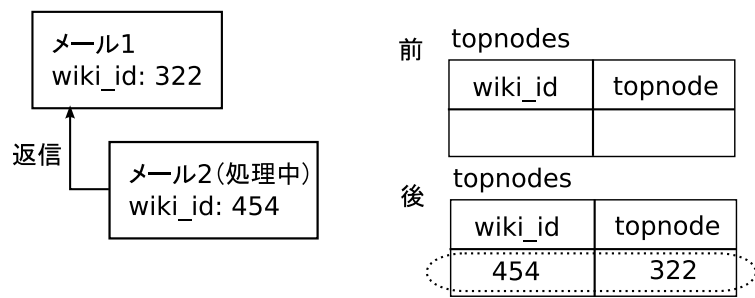
まず，データベースの topnodes テーブルから返信先に対応する topnode を求める．そのような topnode が存在すれば，その topnode をそのまま今処理中のメールの topnode にする (図 9 の (a))．存在しない場合，返信先のメールを今処理中のメールの topnode にする (図 9 の (b))．

#### 添付ファイルの処理

電子メールの規格には，一通の電子メールに複数の異なる種類のデータを格納するために決められた MIME マルチパートというものがある．[24] 本来，電子メールはテキストデータ



(a)



(b)

図 9: topnode を求める方法

しか扱わないため、画像などのデータの送信ができなかったが、この拡張仕様によって、画像などのファイルをテキストデータにエンコードして送信することができるようになった。しかし、元のデータを再現するためにデコードを行わなければならない。本節では、このように電子メールに添付ファイルが付属している場合の処理について説明する。

本システムでは、MIME マルチパートの電子メールを処理するために、Perl のモジュールとして配布されている `MIME::Parser` を利用した。そうすれば、電子メールの各パート(ファイルのこと)が自動的に分離され、指定したディレクトリに出力される。ただし、電子メールの本文もテキストファイルとして出力する。本システムでは、これらの出力ファイルから、テキストのファイルだけをメールの本文としてデータベースの `bodytext` に登録する。その他のファイルについては `MIME::Parser` モジュールに用意されてあるサブルーチンで処理し、ファイル名 (`filename`) とファイルタイプ (`filetype`) を取得する。ファイルサイズ (`filesize`) とデータ (`data`) については Perl の入出力機能で取得する。

#### 5.4 検索機能

本システムでは、データベースに蓄積されたメール・Wiki の内容に対してユーザが入力したキーワードを用いて全文検索を行う。ただし、メール・Wiki が編集可能であるため、全文検索を行う際、メール・Wiki の最初のバージョンと最新バージョンだけを検索の対象にした。全文検索のエンジンとして、本システムが利用したデータベース管理システム MySQL[28] の全文検索機能を利用した。本節で本システムにおけるキーワードの処理と検索結果のスコア計算について説明する。また、検索速度向上のために設けた検索キャッシュについても簡単に説明する。

##### キーワード分析

本システムでは、キーワードの前に書かれた前置記号によって、AND(+), OR(|) と NOT(-) という論理演算によって検索を行うことが可能である。例えば、”+ FreeBSD |Gnome - nautilus” と入力すると、FreeBSD というキーワードは必ず結果に含まれなければならないが、Gnome キーワードは含まれても、含まなくても良いことになる。この中から、更に nautilus キーワードが含まれている結果が除かれる。ただし、記号が記述されていない場合、AND と解釈される。また、”FreeBSD 5.4-RELEASE”のように複数の語からなるキーワードを引用符(「」または「’) で括ることで検索を行うこともできる。従って、最初にユーザが入力したキーワードの分析を行わなければならない。

キーワードの分析のために、まず AND, OR, NOT のために3つの配列が用意される。キーワードの分析は、ユーザからの入力文字列を最初から順に処理していった、記号によって3

検索キーワード： Apache PHP MySQL

検索結果	$S_{my}$	結果1の詳細
結果1	1.00	Subject: Apache Install Tags: Apache, PHP Rank: 5.44
結果2	0.80	
結果3	0.75	結果1のXの算出
.	.	$X = 1.00 + 5.44 + 1 + 2 \times 2$
.	.	$= 11.44$
.	.	

図 10:  $X$  の算出例

つの配列に振り分けることによって行われる。これらの配列から、MySQL データベースへのクエリ文が作成される。

#### 検索結果のスコア計算

前にも述べたように、本システムは MySQL の全文検索機能を利用して検索を行う。この場合、MySQL は内部に作成されたインデックスより検索結果にスコアを付ける。本システムは、この MySQL の内部に付けられたスコアに加え、下記のことも検索結果の最終スコアの計算に取り入れる。なお、これから MySQL 内部に検索結果に付けられたスコアを  $S_{my}$  と記述する。MySQL の検索結果はメール・Wiki 単位で返されるが、本システムでは、検索結果はスレッド単位で表示される。従って、スコアもスレッド単位に付けなければならない。このスレッドに付けたスコアが検索結果の最終スコアであり、 $S_{th}$  と記述する。これから、メール・Wiki に対するスコア、 $X$  の計算を説明してから、スレッド単位のスコア、 $S_{th}$  の計算について説明する。

$X$  は、 $S_{my}$  に加え、ユーザに付けられたランク、 $R$ 、メール・Wiki のタイトルに含まれるキーワードの数、 $N_{title}$  と、メール・Wiki に付けられ、キーワードにも含まれるタグの数、 $N_{tag}$ 、という4つの値から式(1)を用いて算出する。なお、検索結果に属さないメール・Wiki の  $X$  は0とする。算出の具体的な例を図10に示す。

$$X = S_{my} + R + N_{title} + 2N_{tag} \quad (1)$$

次に、スレッドの最終スコア、 $S_{th}$  の算出について説明する。 $S_{th}$  は、スレッドに属する各メール・Wiki の  $X$  の合計である。スレッドのトップのスコアを  $X_0$  に、その下にあるメール・Wiki のスコアをそれぞれ  $X_1, X_2, \dots, X_{n-1}$  にすると、 $S_{th}$  は式(2)で算出できる。な

お,  $n$  はスレッドに属するメール・Wiki の数である .

$$S_{th} = \sum_{i=0}^{n-1} X_i \quad (2)$$

#### 検索キャッシュ

メーリングリストのアーカイブに蓄積されたメールの数は膨大であるため, 毎回検索処理を行うのが効率的ではない . そこで, 検索速度を上げるために簡単なキャッシュを設けた .

このキャッシュでは, 検索結果を識別するために, メール・Wiki の送信日・作成日や, タイトルのみもしくはタイトルと本文を検索対象とするなどの検索範囲と, キーワードから生成した一意的なキーを記録する . これらのキーと同時に, 検索結果とスコアのリストも保存される . 10 分以内に同じ条件での検索があった場合, 検索をもう一度行うのではなく, このキャッシュから結果を取得することによって検索速度の向上を果たす .

### 5.5 Web 共同編集機能

本節では試作したメール検索システムの Web 共同編集機能について説明する . この Web 共同編集機能を通して, ユーザは Web インターフェイスからシステム内のメール・Wiki の内容を編集したり, 新規に Wiki ページを作成したりすることが可能になる . また, 直接的な編集ではないが, メールに対してランクを付けたり, コメントを投稿したりすることもできる . 更に, タグを利用してメールを分類することもできる . 次に, メール・Wiki の編集及び作成について説明してから, ランクとタグについて説明する .

#### メール・Wiki の編集及び作成

4.1 節でも述べたように, 本システムではメールと Wiki ページを同じようなものとして扱うことにした . それらの唯一な違いは, メールは最初からメーリングリストアーカイブに存在してデータベースに読み込まれたのに対して, Wiki ページは本システムの Web インターフェイスを通じて作成されることである . また, メールは規格上スレッド構造を持っているが, Wiki ページではスレッド構造はない .

メール・Wiki を編集・作成するにあたって, 2.2 節で紹介した Wiki システムのマークアップを利用することによって, テキストのみならずより豊富な表現ができるのである . 本システムでは, Wiki システムのマークアップを HTML のマークアップに変換するために PEAR の PHP モジュールである Text\_Wiki[32] を導入したため, 本システムで利用できるマークアップもそれに制限される .

メール・Wiki の編集・作成は利用者に制限なく自由に行うことができる。これは本システムの利点であり、問題点でもある。なぜなら、悪意を持った者によってメール・Wiki の内容に間違った情報などを書かれた恐れがあるからである。これは Wiki システムに共通する問題点であろう。従って、本システムではメール・Wiki の編集・作成履歴は保存されており、内容の復帰などに備えている。

#### ランクとタグについて

ユーザはメール・Wiki に-10 から+10 の間にランクを付けることができる。メール・Wiki の最終的なランクは、ユーザが付けたランクの平均値である。

メール・Wiki に英数字からなる任意の文字列をタグとして付けることもできる。ただし、1 つのメール・Wiki に対して複数のタグを付けることができる。このタグによってメール・Wiki を分類することができる。後に説明するナビゲーション・サイドバーからユーザに付けられたタグの一覧が参照でき、特定のタグでメール・Wiki を絞り込み表示させることができる。

5.4 節で説明したように、メール・Wiki のランクも、メール・Wiki に付けられたタグも検索結果のスコアの計算に反映する。従って、メール・Wiki のランクとタグを変更することによって検索結果の表示順位が変わり、検索の精度を改善することができる。

## 5.6 ユーザインターフェイス

これから本システムのユーザインターフェイスについて説明する。このユーザインターフェイスは PHP 言語 [27] で実装され、任意の Web ブラウザからアクセスできる。なお、そのメインインターフェイスは図 11 に示す。

#### 検索インターフェイス

検索インターフェイスは、ユーザからの検索キーワードの入力を受け取って、データベースの検索結果をユーザに表示する。なお、検索キーワードの入力部は図 12 に示す。キーワードの入力以外にも、メール・Wiki の送信日・作成日によって検索範囲を指定したり、タイトルのみもしくはタイトルと本文の両方を検索の対象に指定したりすることもできる。また、5.4 節で説明したキャッシュを無効にすることもできる。

検索結果の表示は図 13 に示した。検索結果はスレッド単位でまとめられ、ここではスレッドのトップだけを表示する。タイトルの他に、このメールに関する色々な情報とスコアも表示される。



図 11: 本システムのメインインターフェイスの構成

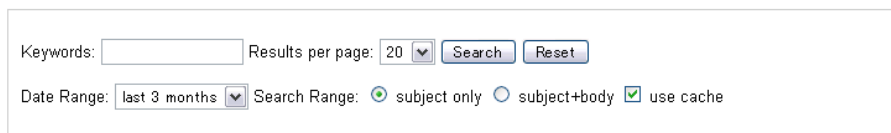


図 12: 検索入力部

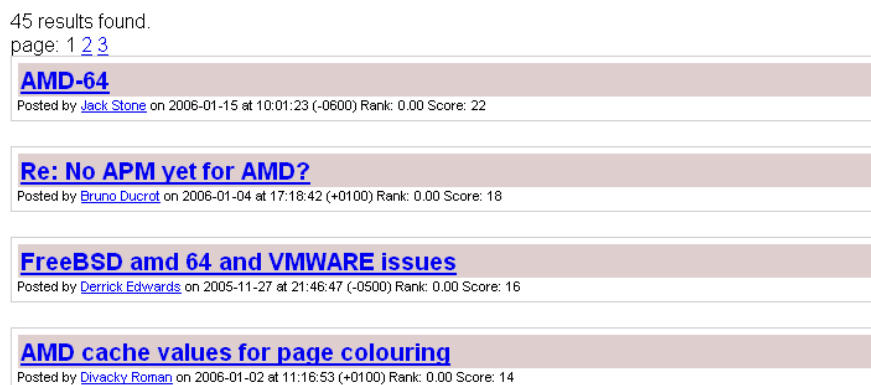


図 13: 検索結果の表示





図 14: メール・Wiki の内容表示

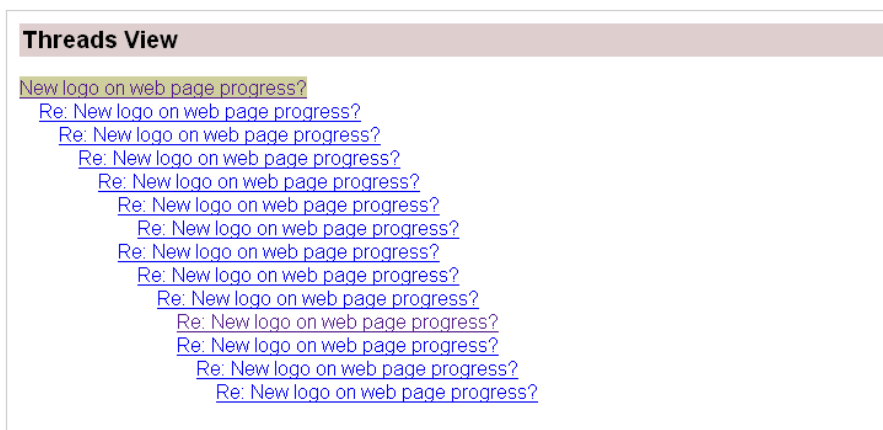


図 15: スレッド構造の表示

## メール・Wiki の単体表示インターフェイス

図 13 から閲覧したいスレッドをクリックすれば、メール・Wiki の単体表示インターフェイスに移る。ここでは、メール・Wiki の内容 (図 14)、表示中のメール・Wiki が属するスレッドのスレッド構造 (図 15)、添付ファイルの表示 (図 16)、編集部 (図 17)、コメント (図 18) と編集履歴 (図 19) が表示される。

メール・Wiki の表示部では、タイトルの他に、このメール・Wiki の送信日・作成日やランク、バージョンなどの情報とその内容が表示される。

スレッド構造の表示部から、このスレッドに属する他のメールへのリンクが表示され、順に閲覧していくこともできればランダムに参照することもできる。

メールに添付ファイルがある場合、その添付ファイルは添付ファイルの表示部に表示される。ここからダウンロードすることもできる。

Filename	Filetype	Filesize	Download
signature.asc	application/pgp-signature	187	<a href="#">download</a>

図 16: 添付ファイルの表示

**Edit**

Rank this message: +10

Add tags(comma separated):

Tags: [freebsd logo](#) (Delete) [freebsd website](#) (Delete)

図 17: 編集部の表示

編集部から、表示中のメール・Wiki に対するランク付けとタグ付けができる他、編集画面 (図 20) とコメントの追加画面 (図 21) へのボタンも表示される。ただし、タグを追加するには、例えば”FreeBSD 5.4-RELEASE, Apache, PHP”のようにタグをカンマで区切ってテキストフィールドに入力してから Submit ボタンをクリックすればよい。また、表示中のメール・Wiki に付けられたタグもここで表示される。既存のタグを外すには、タグの後ろにある Delete というリンクをクリックすればよい。

表示中のメール・Wiki に投稿されたコメントはコメント表示部に表示される。投稿者の名前と投稿した日時も表示される。

表示中のメール・Wiki の編集履歴情報は編集履歴情報表示部で表示される。ここから特定のバージョンを閲覧したり、バージョン間の差分情報を表示させたりすることができる。

**Comments**

Posted by [kovi](#) on 2006-02-08 at 13:15:42

I like this post.

Posted by [AnotherOne](#) on 2006-02-08 at 13:18:12

Another comment.

図 18: コメントの表示



図 19: 編集履歴の表示

## メール・Wiki の作成・編集とコメント追加インターフェイス

本節でメール・Wiki の作成や編集とコメント追加に関するインターフェイスを紹介する。まず、図 20 にメール・Wiki の編集画面を示す。図 17 の”Edit this message”ボタンをクリックするとこの画面に移る。テキストボックスにメール・Wiki の最新の内容が表示され、これを編集してから”Submit changes”ボタンをクリックすれば変更が保存される。編集者の名前とメールアドレスを入力することもできるが、強制ではない。

図 17 の”Add Comment”ボタンをクリックすると、図 21 に示したコメントの追加画面に移る。各入力フィールドに入力して”Add comment”ボタンをクリックすればコメントが投稿される。

データベースに存在しない Wiki ページを作成するには、まずキャメルケース (Camel Case) で作成したい Wiki ページのタイトルを既存メール・Wiki の本文中に書く。すると、図 14 に示したメール・Wiki の内容の第一行目の”?FreeBSD”のように、その複合語の先頭に疑問号が付く。この疑問号をクリックすれば、図 22 に示した画面に移る。ここに作成したい Wiki ページの内容を入力して、”Submit”ボタンをクリックすればその Wiki ページが作成される。これによって、図 23 に示したように疑問語が表示されなくなる。その代わりに、”FreeBSD”という複合語がリンクと変わり、このリンクをクリックすれば作成した Wiki ページが表示される。

## ナビゲーション・サイドバー

最後に、色々な観点でメール・Wiki を絞って表示させることのできるナビゲーション・サイドバーについて説明する。ナビゲーション・サイドバーは図 24 に示す。このナビゲーション

**Edit: New logo on web page progress?**  
Posted by [Siebrand Mazeland](#) on 2006-01-15 at 00:18:20 (+0100) Rank: 2.66

Name:

E-mail:

```
A while ago kuriyama@ proposed a new layout for the FreeBSD webpage using the new
logo.

Any status on getting this in the actual site? It appears as the last update
on this is from about a month ago.

Siebrand

[1] http://people.FreeBSD.org/~kuriyama/www/
[2] http://people.FreeBSD.org/~kuriyama/www/logo.html

Added one line.
```

図 20: メール・Wiki の編集画面

**Add Comment: New logo on web page progress?**  
Posted by [Siebrand Mazeland](#) on 2006-01-15 at 00:18:20 (+0100) Rank: 2.66

Name:

E-mail:

Comments:

図 21: コメントの追加画面

**Add: FreeBSD**

Name:

E-mail:

Contents:

図 22: Wiki ページ追加画面

**New logo on web page progress?**

Posted by [Siebrand Mazeland](#) on 2006-01-15 at 00:18:20 (+0100) Rank: 2.65 Version: 5

A while ago [kuriyama@](#) proposed a new layout for the [FreeBSD](#) web page using the new logo.

Any status on getting this in the actual site? It appears as the last update on this is from about a month ago.

Siebrand

[1] <http://people.FreeBSD.org/~kuriyama/www/>  
[2] <http://people.FreeBSD.org/~kuriyama/www/logo.html>

Added one line.

図 23: Wiki ページへのリンクの表示

The image shows a vertical navigation sidebar with three sections. The first section, 'Top Keywords:', lists ten items: 1. [freebsd](#), 2. [bsd](#), 3. [kernel](#), 4. [USB](#), 5. [php](#), 6. [Jeremy](#), 7. [Peter](#), 8. [Fatal](#), 9. [panic](#), and 10. [trap](#). Below this list is a link [Show all keywords...](#). The second section, 'Top Tags:', lists six items: 1. [freebsd website](#), 2. [freebsd logo](#), 3. [bios](#), 4. [amd](#), 5. [athlon](#), and 6. [USB device](#). Below this list is a link [Show all tags...](#). The third section, 'Recently Added Pages:', lists one item: 1. [FreeBSD](#). Below this list is a link [Show all pages...](#).

図 24: ナビゲーション・サイドバー

ン・サイドバーは常に画面の左側に表示され、4.3 節で述べたように、検索されたキーワードや、ユーザの付けたタグなどでメール・Wiki を参照できる。

## 6 利用例

本節では、メーリングリストアーカイブから必要な情報を引き出そうとするユーザが本システムを利用する際の操作の流れを説明してから、本システムを利用すれば解決できる問題の実例を2つ紹介する。

### 6.1 操作の流れ

図 25 に本システムを用いてメーリングリストアーカイブから必要な情報を検索するための操作の流れを示す。まず、ユーザには2つの選択肢がある。1つは、キーワードを用いて検索を行う方法である。もう1つは、ナビゲーション・サイドバーから他の観点からメールを絞り込む方法である。

検索もしくは絞り込みの結果が表示されたら、閲覧したいスレッドを選択し開く。一般ユーザならば、ここでメールを読んでからそのまま次のステージに移ることが多いであろうが、より高度な知識を持ったユーザならば、そのメールの間違ったところを編集したり、新しい Wiki ページを作成したりすることもできる。また、ここでメールにリンクやタグを付けたり、コメントを投稿したりすることもできる。

閲覧中のメールが必要な情報を含んでいればここで操作が終わるが、そうでなければ検索結果の一覧に戻るか、最初からやり直すことになる。

### 6.2 実例

次に、本システムを利用すれば解決できる問題の実例を2つ紹介する。

#### タグによる絞り込み

多くの場合、FreeBSD の利用者は同じコンピュータに FreeBSD とともに Windows もインストールして運用している。このようなシステム構成はデュアルブート環境と呼ばれる。しかし、Unix ベースの FreeBSD と Windows は技術と実装上あまりにも異なっているため、両者の間での直接的なデータ共有は難しい。ただし、ハードディスクのパーティションを FAT ファイルシステムでフォーマットすれば、FreeBSD と Windows の両方からその中にあるファイルが読み書きできるようになる。しかし、ある程度の経験を持っているユーザならば、“FAT” や “mount” などのキーワードで素早く情報を入手することができるが、初心者にはそうにはならない。“windows” で検索するだけでは情報が多すぎて逆にわからなくなるのである。

そこで、タグ機能を利用すればこの問題が解決できる。これは、上級者によって予めメールにタグを付けることによるメールを分類する手法である。例えば、図 26 に示したメール

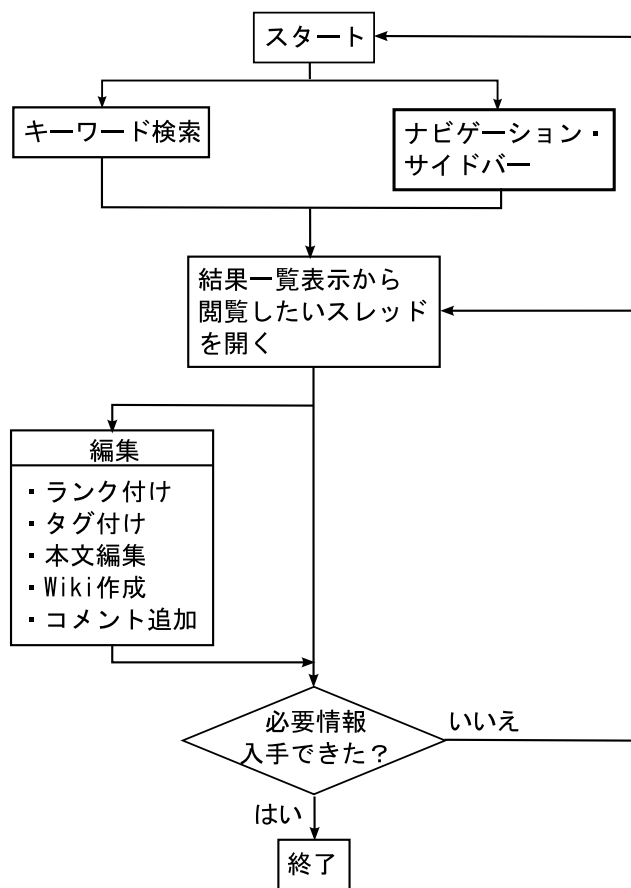


図 25: 操作の流れ



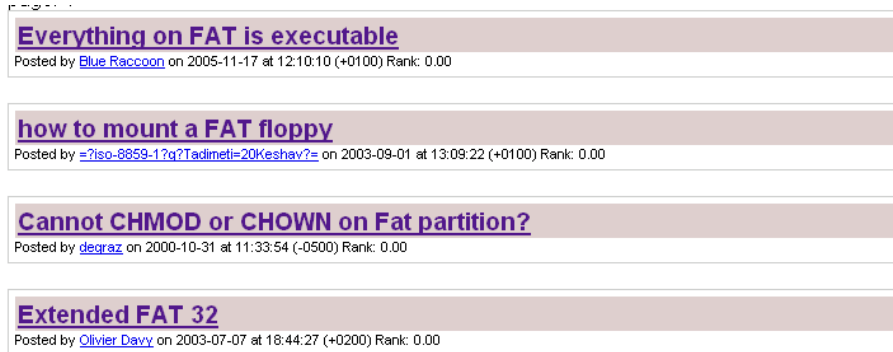


図 26: タグでまとめたメール



図 27: 迷惑メールの例

は、タイトルの中に”windows”というキーワードさえ含んでいないが、その内容は前述したデータ共有に関して大事な情報が入っている。これらのメールに例えば”windows data share”というタグを予め付けておけば、他のユーザはこのタグを選択すればその一覧が簡単に見つけることができる。

### 迷惑メール対策

あるユーザが安いハードウェアについて知りたいため、”cheap”をキーワードとして検索を行った。そこで、図 27 に示した 2 つの迷惑メールが検索結果の 6 ページ目に表示された。明らかに迷惑メールであるため、それらのメールのランクを最低の”-10”にした。そうすると、次回の検索では、これらのメールは検索の最後である 12 ページ目に表示されるようになった。

## 7 評価

本システムの処理速度と検索精度を評価するために、データベースの作成や検索処理にかかる時間と、検索結果についての再現率、適合率の測定及び  $f$  値の計算を行った。また、今後の研究の方向を確認するために、FreeBSD の実際のユーザに簡単なアンケートの採取を行った。本節では、それらの評価の結果を報告する。

### 7.1 システムの処理速度の評価

本システムの処理速度の実用性を評価するために、データベースの作成と、検索処理にかかる時間の測定をそれぞれ行った。次に、その結果について説明する。

#### データベースの作成時間

本実験では、メーリングリストのアーカイブからデータベースを作成する時間が実用的なものかどうかを評価した。対象としたのは、FreeBSD のメーリングリストの中、“freebsd-questions”、“freebsd-stable”、“freebsd-current”などの“freebsd-”から始まるリストに、1994 年から 2006 年 1 月 17 日までの期間内にアーカイブ化されたメールである。測定の結果は表 6 に年別にまとめた。ただし、1994 年と 2006 年については、2006 年 1 月 17 日までにメーリングリストのアーカイブに存在するメールだけが登録の対象となる。また、MySQL を利用した全文検索の速度を上げるために、メールの本文に対してインデックスの作成を行わなければならない。表 6 に表示した登録時間以外に、インデックスの作成に 1 時間 17 分を要した。

表 6 に示した所要時間とインデックスの作成時間を合わせると、全てのアーカイブの登録処理には 22 時間近く、ほぼ一日かかるが、これは最初の一回だけ行えばよいから、実用的に言えるであろう。実際の場合、1 週間単位でメールの追加登録が行われると予想できるため、数十分もあれば十分であろう。

#### 検索処理にかかる時間

試作したシステムの検索時間が実用的なものかどうかを確認した。本節では、その評価の結果を説明する。

検索処理にかかる時間の評価は、検索範囲に分けて行った。検索は、“apache php”、“ufs filesystem”と“firefox plugin”という 3 つのキーワードで行った。3 回の検索結果の数の平均や、検索にかかった時間の平均は表 7 にまとめた。なお、日時検索範囲の下に書いてある期間は、2006 年 1 月 17 日に対する期間である。

表 6: データベース作成に要する時間

年	メール数	総ファイルサイズ (MB)	所要時間
1994	758	1.6	0m 30s
1995	56,896	118	34m 02s
1996	85,206	191	52m 44s
1997	99,204	232	1h 03m 47s
1998	138,186	364	1h 30m 10s
1999	159,060	407	1h 45m 38s
2000	192,126	473	2h 04m 56s
2001	197,446	601	2h 30m 47s
2002	192,103	691	2h 32m 33s
2003	171,685	705	2h 28m 53s
2004	173,022	741	2h 34m 58s
2005	158,806	728	2h 29m 35s
2006	6,510	32	07m 23s
合計	1,631,008	5,284.6	20h 35m 56s

表 7: 検索処理の所要時間

日時検索範囲	メール検索範囲	平均結果数 (スレッド)	キャッシュ状況別 検索所要時間 (秒)		
			作成	利用	無効
3ヶ月以内	タイトルのみ	5	0.13373	0.02880	0.10193
	タイトルと本文	84	0.45158	0.05442	0.40581
6ヶ月以内	タイトルのみ	10	0.11580	0.03058	0.11010
	タイトルと本文	177	0.56749	0.07011	0.50480
1年以内	タイトルのみ	15	0.07014	0.03243	0.05850
	タイトルと本文	364	0.79693	0.09783	0.45728
全部	タイトルのみ	135	0.52060	0.06705	0.27532
	タイトルと本文	2691	6.22061	0.45243	4.94682

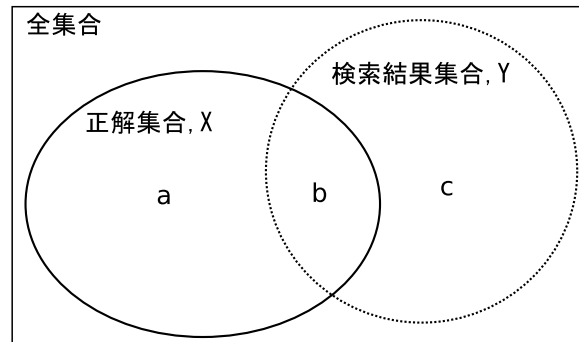


図 28: 再現率, 適合率を表す図

検索時間は検索結果の数に大きく影響される。表 7 からわかるように、結果の数が少ない場合、検索はほぼ一瞬に行われるのに対し、結果の数が増えると、何秒間もかかってしまう。しかし、キーワードの選択は一般的になりすぎない限り、十分実用に耐える時間内に検索が実行されると思われる。

次に、キャッシュの効果について説明する。キャッシュの作成は、キャッシュ内に検索条件にあったものが見つからない場合、もしくはそのキャッシュが作成されてから 10 分経過した場合に行われる。表 7 からわかるように、キャッシュの作成が行われる場合は、キャッシュを利用しない場合よりも検索時間が長くなる。しかし、キャッシュが作成され、利用された場合には、その検索時間の短縮が無視できない。表 7 の最後の行の場合、キャッシュを利用しない場合に比べて、キャッシュの利用によって検索時間が最大 90.85% 短縮される。

## 7.2 再現率, 適合率及び $f$ 値の測定

既存の全文検索システムと本システムで再現率, 適合率を求め,  $f$  値を計算し比較を行った。この評価ではユーザからの入力や編集があまりない初期状態でも一定程度の精度が実現されていることを証明したい。

まず, 図 28 を使って再現率, 適合率について説明する。図中の正解集合,  $X$  は, 検索対象となる全集合のうち, 正解とされるものの集合である。一方, 点線に囲まれた検索結果集合,  $Y$  は, 実際の検索で返された結果の集合である。 $X$  の要素の数を  $N_X$  とすると,  $N_X = a + b$  となる。また,  $Y$  の要素の数を  $N_Y$  とすると,  $N_Y = b + c$  となる。再現率,  $R$  と適合率,  $P$  は以下のように定義される。

$$R = \frac{b}{N_X}, P = \frac{b}{N_Y} \quad (3)$$

表 8: 既存システムとの比較実験の結果

	再現率	適合率	$f$ 値
Namazu	45.0%	40.8%	0.43
本システム	46.5%	43.5%	0.45

- 再現率, R: 正解とされる結果のうち, 検索結果として出された結果の割合
- 適合率, P: 検索結果として出された結果のうち, 正解とされる結果の割合

再現率, 適合率は上記のように定義され, 両方ともに高いほど優れた検索システムであると判断できる. また,  $f$  値とは,

$$f \text{ 値} = \frac{2 \times R \times P}{R + P} \quad (4)$$

と定義され, 情報検索の精度を表す指標として用いられる.  $f$  値が大きいほど, 情報検索の精度が高いとされる.

次に, 評価のために行った実験を簡単に説明する.

freebsd-stable メーリングリストで 2005 年 12 月にやりとりされたメールの中から「デバイスドライバーに関する議論」を探すという課題を設定し再現率, 適合率を求めた. freebsd-stable メーリングリストで 2005 年 12 月にやりとりされたメールの総数は 1017 通であり, 248 スレッドが存在した. そのうち「デバイスドライバーに関する議論」と判断されたスレッドは 40 個存在した. 既存の全文検索システムと, 本システムの両方について検索を行い, 検索結果のスレッドが上記 40 個のスレッドに属するか否かによって分類し, 再現率, 適合率を計算した.

実験では, デバイスドライバーに関する異なるキーワードの組み合わせを用いて 5 回検索を行い, 再現率, 適合率の平均値を求めた. これらのキーワードの組み合わせとは, "device driver", "device support", "driver", "driver problem" と "driver support" である. 既存の全文検索システムには, Namazu[34] を利用した.

実験結果は表 8 に示す. 実験の結果から, ユーザの入力や編集があまりなかった初期状態でも, 本システムは既存全文検索システム Namazu と同程度の検索精度が実現されていることが言えよう.

### 7.3 アンケート

本システムで提案及び実装した各機能の実用性を実際の FreeBSD のユーザにオンラインアンケートを通して評価してもらった. そのアンケートの質問と集計結果については付録 B

と付録 C を参照して下さい。

このアンケートは freebsd-current, freebsd-hackers という開発者向けのメーリングリストと, freebsd-questions, freebsd-advocacy と freebsd-chat という一般ユーザ向けのメーリングリストの利用者に 3 週間にわたって公開された。回収できたのはわずか 14 件しかなかったが、回答者の中で上級者(スキルレベルに”expert”と”experienced”と答えた者)が 9 名, 中級者以下は 6 名(付録 C の図 34 参照)という構成になっているため, その答えは十分に参考にできると思われる。次に, アンケートの結果の中で特に注目すべき点について説明する。

一般の場合, 目的を持ってメーリングリストのアーカイブを検索することが多いが, 今回のアンケートで何となく(”For Fun”)アーカイブを検索しメールを閲覧する利用者が 4 名いることがわかった(付録 C の図 35 参照)。このようなユーザは本システムが提供する機能を利用してメーリングリストアーカイブの検索精度を上げることが期待できる。実際のところ, その 4 名のユーザは次の質問でメールのランクやタグ付けなどの機能を利用しようと思うと答える傾向がある。

次に, 本システムの提供する機能の中, 実際にユーザが使おうとする機能について説明する。付録 C の図 42 にその詳細を示す。14 人の中で「タグ付け」「ランク付け」と「コメント追加」機能を使おうとするユーザがその半分以上を超えたが「メール編集」と「Wiki 作成」を使おうとするユーザは少なかったことがわかった。Wiki の作成には高度な知識が必要なため, 作成しようと思ってもなかなか難しい点が存在するため, 数字の低さは簡単にわかる。一方, メール編集をしようと思うユーザがそれよりも少ない原因はアンケートに書かれている。以下に簡単にまとめた。

- 同じメールの異なるバージョンが存在し混乱が起きかねない。
- コメント追加だけで十分である。
- 悪意の持った利用者にメールに誤った情報を書かれてしまう恐れがある。

これらの問題点に対して, 下記の解決方法も提案された。

- メールが編集された事実をより強調すべき。
- 完全に自由に誰でも編集できるよりはユーザ登録などの仕組みで登録したユーザのみが編集できるようにすべき。

アンケートの結果を全体的に見て, メール編集機能に戸惑う利用者が多かったが, その他の機能はよく実装され, 実用であると言えよう。特に, 付録 C の図 38, 図 39 と図 40 を見れば, ユーザが有益な情報をより効率よく入手できることが言えよう。

## 8 まとめ

オープンソースソフトウェア開発を研究対象として、まずオープンソースソフトウェア開発環境について説明した。オープンソースソフトウェア開発にかかわる開発者とオープンソースソフトウェアの利用者は、問題を解決するためにメーリングリストアーカイブを検索する。しかし、既存のメーリングリストアーカイブには、メールの厳密な分類ができない、及び、メールの内容の品質が保証されないという2つの大きな問題が存在している。また、メーリングリストアーカイブの参照手法にも、スレッド構造が再現されない、キーワード選びが困難であるなどの問題点が存在すると指摘した。

そこで、Web 共同編集機能を実装したメール検索システムの設計と実装について述べた。具体的に、利用者による「メール編集」、「新規 Wiki 作成」、メールに対する「ランク付け」、「タグ付け」及び「コメント追加」によってメーリングリストアーカイブからより簡単に必要な情報を引き出せることを支援するシステムの提案をし、その設計及び実装について説明した。

それから、試作したシステムにおける操作の流れを説明し、このシステムが実際に役に立つ例を2つ挙げた。

試作したシステムの実用性を評価するために、データベースの作成や検索に必要な処理時間の測定と評価を行った。その結果、本システムは大規模なオープンソースソフトウェア開発に適用した場合でも実時間で動作することを確認した。また、既存全文検索システムとの再現率、適合率とf値の比較も行った。比較の結果、利用者のフィードバックが少ない初期状態でも十分に高い検索精度を実現できることを確認した。さらに、今後の研究の方向を確認するために、実際にシステムをFreeBSDのユーザに公開し、アンケートの採取を行った。アンケートの結果から、メールの編集機能以外に良い評価をもらった。

現状では、メールの編集に抵抗を持つ利用者が多いが、その抵抗感を解消するために、例えば編集するための権限管理をより固めたり、実際のメールを編集するのではなく、メールを処理して内容を抽出し、この抽出した内容を編集できるようにしたりするなどの工夫が必要だと思われる。また、より長い期間でシステムを公開しその有効性を検証する必要があると考えられる。

## 謝辞

本研究において、常に適切な御指導および御助言を賜りました大阪大学大学院情報科学研究科コンピュータサイエンス専攻 井上克郎 教授に深く感謝致します。

本研究において、常に適切な御指導および御助言を頂きました同 松下誠 助教授に深く感謝致します。

最後に、その他様々な御指導、御助言を頂きました大阪大学大学院情報科学研究科コンピュータサイエンス専攻 ソフトウェア工学講座 井上研究室の皆様に深く感謝いたします。



## 参考文献

- [1] Eric S. Raymond, “The Cathedral & Bazaar”, O’REILLY, 2001.
- [2] Eric S. Raymond, “The Art of Unix Programming”  
<http://www.faqs.org/docs/artu/index.html>.
- [3] Open Source Technology Group, SourceForge,  
<http://sourceforge.net/>.
- [4] Wikipedia, the free encyclopedia,  
<http://www.wikipedia.org/>.
- [5] 落水浩一郎, “分散共同ソフトウェア開発に対するソフトウェアプロセスモデルに関する基礎考察”, 電子情報通信学会技術研究報告, SS2000-48(2001-01), pp.49-56,2001.
- [6] The FreeBSD Project, The FreeBSD Project,  
<http://www.freebsd.org/>.
- [7] Linux Online Inc., The Linux Home Page,  
<http://www.linux.org/>.
- [8] The Mozilla Project, Mozilla.org,  
<http://www.mozilla.org/>.
- [9] The Apache Software Foundation, Apache Projects,  
<http://www.apache.org/>.
- [10] The OpenOffice Organization, OpenOffice.org,  
<http://www.openoffice.org/>
- [11] Jacky Estublier, “Software Configuration Management: A Roadmap”. The Future of Software Engineering in 22nd ICSE, pp.281-289, 2000.
- [12] Brian Berliner, “CVS II: Parallelizing Software Development”, In USENIX Association, editor, Proceedings of the Winter 1990 USENIX Conference, pages 341-352, Berkeley, CA, USA, 1990.
- [13] Karl Fogel, “Open Source Development with CVS”, The Coriolis Group, 2000.
- [14] 鯉江英隆, 西本卓也, 馬場肇, “バージョン管理システム (CVS) の導入と活用”, SOFT-BANK, December, 2000.

- [15] 大月美佳 , “入門 CVS Concurrent Versions System” , SHUWA SYSTEM Co., Ltd, 2001.
- [16] Wikipedia, the free encyclopedia, ARPANET,  
<http://en.wikipedia.org/wiki/ARPANET>.
- [17] WikiWikiWeb, FrontPage,  
<http://c2.com/cgi/wiki>
- [18] MediaWiki - MediaWiki -,  
<http://www.mediawiki.org/wiki/MediaWiki>
- [19] TWiki - Enterprise Collaboration Platform & Wiki -,  
<http://twiki.org/>.
- [20] The Apache SpamAssassin Project, Welcome to SpamAssasin,  
<http://spamassassin.apache.org/>.
- [21] GNU Project, GCC Home Page,  
<http://gcc.gnu.org/>.
- [22] David H. Crocker, “Standard for the Format of Arpa Internet Text Messages”. RFC 822.  
<http://www.ietf.org/rfc/rfc822.txt>.
- [23] P. Resnick, “Internet Message Format” RFC 2822,  
<http://www.ietf.org/rfc/rfc2822.txt>.
- [24] N. Freed, N. Borenstein, “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies” RFC 2045,  
<http://www.ietf.org/rfc/rfc2045.txt>.
- [25] 増永良文 , “リレーショナルデータベースの基礎” , 東京オーム社 , 1990.
- [26] Randal L. Schwartz, Tom Phoenix, “Learning Perl” , O'REILLY, 2001.
- [27] PHP.net, PHP: Hypertext Preprocessor,  
<http://www.php.net/>.
- [28] 志村伸弘 , “MySQL 徹底攻略ガイド” , 技術評論社 , 2002.
- [29] Comprehensive Perl Archive Network(CPAN)  
<http://search.cpan.org/>.

- [30] PEAR::The PHP Extension and Application Repository  
<http://pear.php.net/>.
- [31] PECL :: The PHP Extension Community Library  
<http://pecl.php.net/>.
- [32] Text\_Wiki, Sample Page,  
[http://wiki.ciaweb.net/yawiki/index.php?area=Text\\_Wiki&page=SamplePage](http://wiki.ciaweb.net/yawiki/index.php?area=Text_Wiki&page=SamplePage).
- [33] The FreeBSD Project, Resources on the Internet, Mailing List,  
[http://www.freebsd.org/doc/en\\_US.ISO8859-1/books/handbook/eresources.html#ERESOURCES-MAIL](http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/eresources.html#ERESOURCES-MAIL).
- [34] Namazu Project, 全文検索システム Namazu,  
<http://www.namazu.org/>.

## 付録

- A. データベースで用いるテーブルの構成
- B. アンケートの質問
- C. アンケートの集計結果

## A データベースで用いるテーブルの構成

ここでは本システムのデータベースの中にある各テーブルの各カラムについて簡単に説明する。

表 9: message

カラム名	説明
wiki_id	メール・Wiki の識別子
msg_id	メール・Wiki の Message-ID

表 10: sender

カラム名	説明
sender_id	送信者の識別子
sender_name	送信者の名前
rank	送信者に対する評価
vote_cnt	送信者に対する評価が行われた回数
msg_cnt	送信者が送信したメールの数
fav_topic	送信者がよく送信する話題
email	送信者のメールアドレス

表 11: topic

カラム名	説明
topic_id	タグの識別子
topic	タグの文字列
cnt	このタグの下にあるメール・Wiki の件数

表 12: maillist

カラム名	説明
ml_id	メーリングリストの識別子
ml_name	メーリングリストの名前

表 13: reference

カラム名	説明
refer_by	下記の refer_to に返信されたメールの wiki_id
refer_to	上記の refer_by への返信メールの wiki_id

表 14: topnodes

カラム名	説明
topnode wiki_id	下記の wiki_id を持つメール・Wiki が所属するスレッドのトップ メール・Wiki の識別子

表 15: wiki\_sender

カラム名	説明
wiki_id	メール・Wiki の識別子
sender_id	上記の wiki_id を持つメール・Wiki の送信者・作成者の sender_id

表 16: wiki\_topic

カラム名	説明
wiki_id	メール・Wiki の識別子
topic_id	上記 wiki_id を持つメール・Wiki に付けられたタグの topic_id

表 17: wiki\_maillist

カラム名	説明
wiki_id	メール・Wiki の識別子
ml_id	上記 wiki_id を持つメールが送信されたメーリングリストの ml_id

表 18: mime

カラム名	説明
mime_id	添付ファイルの識別子
wiki_id	メール・Wiki の識別子
filename	添付ファイルの名前
filesize	添付ファイルのサイズ
filetype	添付ファイルのタイプ
data	添付ファイルの実データ

表 19: wiki\_pages

カラム名	説明
wiki_id	新規作成された Wiki ページの識別子
page_name	新規作成された Wiki ページのタイトル

表 20: archives

カラム名	説明
wiki_id	メール・Wiki の識別子
version	バージョン
old_text	上記バージョンのメール・Wiki の内容
fdate_updated	更新された日付
time_updated	更新された時間
author	編集者の名前
email	編集者のメールアドレス

表 21: wiki

カラム名	説明
wiki_id	メール・Wiki の識別子
fdate_created	メール・Wiki が送信・作成された日付
time_created	メール・Wiki が送信・作成された時間
last_fdate_updated	メール・Wiki が最後に更新された日付
last_time_updated	メール・Wiki が最後に更新された時間
timestamp_updated	更新タイムスタンプ
timezone	タイムゾーン情報
subject	メール・Wiki のタイトル
msg_rank	メール・Wiki のランク
vote_cnt	メール・Wiki がランクされた回数
view_cnt	メール・Wiki が閲覧された回数
comment_cnt	メール・Wiki に投稿されたコメントの数
fsbd_rel	関連する FreeBSD のバージョン
bodytext	メール・Wiki の元の内容
latest_text	メール・Wiki の最新バージョンの内容
latest_version	最新バージョン

表 22: comments

カラム名	説明
id	コメントの識別子
wiki_id	メール・Wiki の識別子
comment	コメントの内容
fdate_created	投稿された日付
time_created	投稿された時間
poster	投稿者
email	投稿者のメールアドレス



表 23: keywords

カラム名	説明
keyword	検索キーワード
cnt	検索された回数

表 24: search\_results

カラム名	説明
keywords	検索に使われたキーワードから生成された一意的な文字列
search_type	検索のタイプ(タイトルのみ, タイトルと本文)を記録する
date_range	検索の時間的な範囲を記録する
expiry	有効期間
wiki_ids	検索結果
scores	検索結果に対応するスコア

## B アンケートの質問

本研究で試作したシステムについて実際の FreeBSD の実際の利用者にアンケートを記入してもらった。図 29 から図 32 はそのアンケートを 3 つのセクションに分けて示す。

### Section 1: Let's hear about you.

Q. What do you use FreeBSD for?

Server  Workstation  Desktop  Other(Please specify):

Q. How do you rate your knowledge/skills in FreeBSD?

Expert  Experienced  Intermediate  Casual user  Beginner

図 29: セクション 1 利用者について

## Section 2: General questions

**Q. In which scenario would you search the mailing list archives?**

- Software installations & configurations
- Software usage information(tutorial, etc.)
- Looking for program candidates
- Looking for solutions for errors encountered
- For fun :D
- Others(Please specify):

**Q. What do you think about the usability/functionality of existing mailing list archive search/management system?**

- Very Good!  Good  Average  Bad  Very bad

**Sub-Q. Why do you think so?**

**Q. What do you think about being able to edit the mails sent to a mailing list?**

- Very good idea  Good idea  Can live without  Bad idea  Very bad idea

**Sub-Q. Why do you think so?**

図 30: セクション 2 一般的な質問

### Section 3: Questions about MLwiki

**Q. Which functions of MLwiki that you are using, or you think is useful to you?**

- Mail search
- Navigation sidebar
- Edit existing mails
- Create new wiki pages
- Rank existing mails
- Apply tags to existing mails
- Comment on existing mails

**Q. Does MLwiki return the information that you are looking for?**

- Definately Yes  Probably Yes  Unsure  Probably No  Definately No

**Q. Do you think this is easier to use than the existing mailing list archive search system?**

- Definately Yes  Probably Yes  Unsure  Probably No  Definately No

**Q. What do you think about the design and implementations of MLwiki regarding to the mail editing functions?**

- Very good!  Good  Average  Bad  Very bad

**Sub-Q. Why do you think so?**

**Q. Would you try to edit the mails if you find errors in them?**

- Definately Yes  Probably Yes  Unsure  Probably No  Definately No

**Q. Would you create new wiki pages?**

- Definately Yes  Probably Yes  Unsure  Probably No  Definately No

**Q. Would you apply tags to the mails?**

- Definately Yes  Probably Yes  Unsure  Probably No  Definately No

**Q. Would you submit comments?**

- Definately Yes  Probably Yes  Unsure  Probably No  Definately No

図 31: セクション 3 の 1 試作したシステムについて

**Q. Would you rank a mail you find useful/unuseful?**

Definitely Yes  Probably Yes  Unsure  Probably No  Definitely No

**Q. What do you think that is bad about MLwiki?**

**Q. If you have any suggestions about improving MLwiki, I would like to hear :)**

**Q. If you any other comments, please write it down here.**

図 32: セクション 3 の 2 試作したシステムについて

## C アンケートの集計結果

ここでアンケートの集計結果を示す。ただし、ここで表示するのは数値的にまとめられるもののみである。意見などのテキストで記入されるものは 7.3 節を参照して下さい。また、複数選択可能な質問については棒グラフで表示し、単一選択の質問についてはパイチャートで表示する。なお、回収できたアンケート回答の合計は 14 件である。

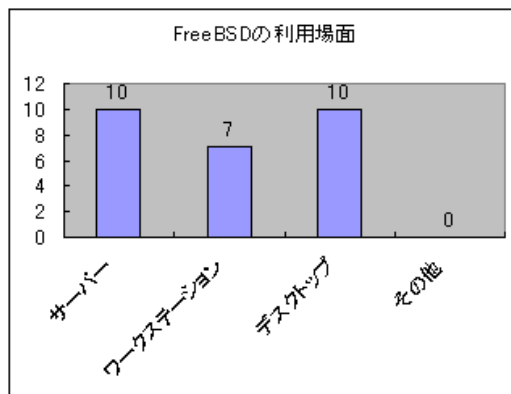


図 33: FreeBSD の利用場面

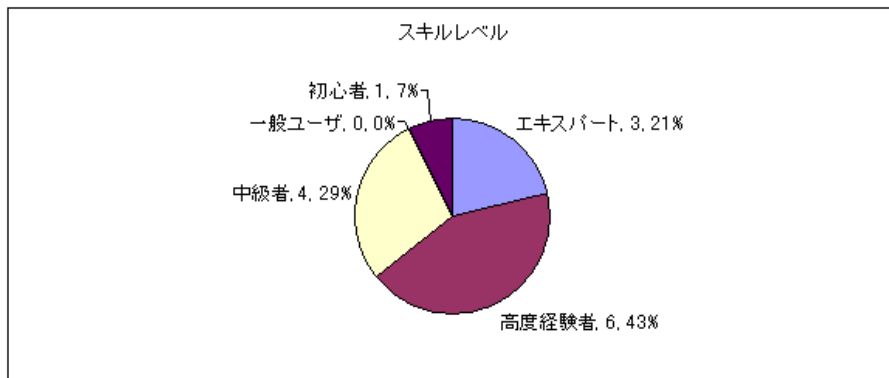


図 34: スキルレベルの自己評価

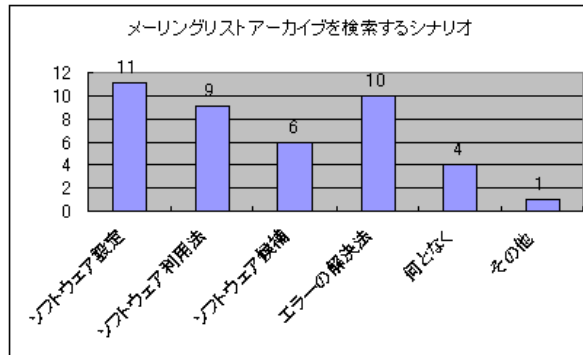


図 35: メーリングリストを検索するシナリオ

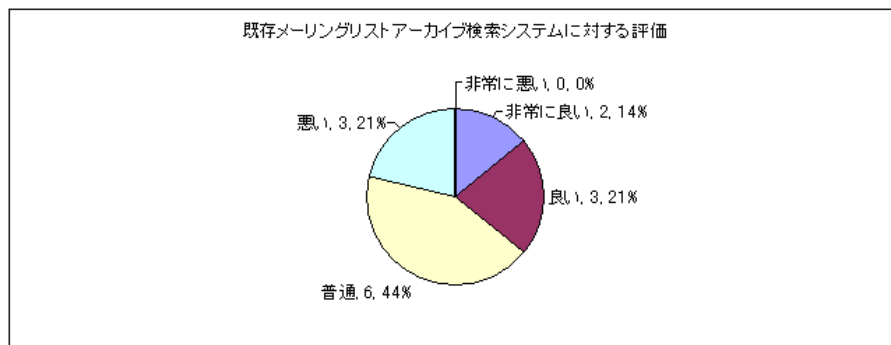


図 36: 既存メーリングリストアーカイブの検索システムに対する評価

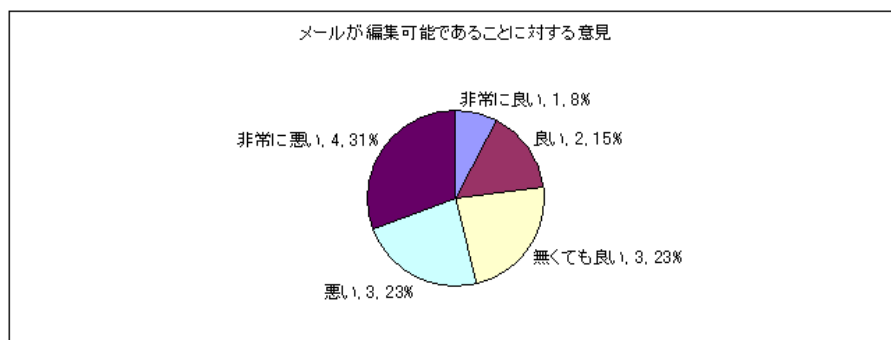


図 37: メール編集可能であることについての意見

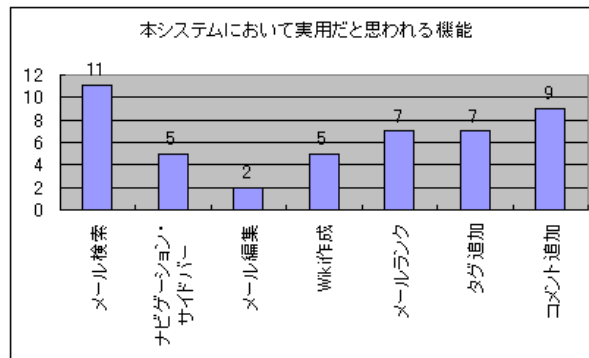


図 38: 本システムにおいて実用的だと思われる機能

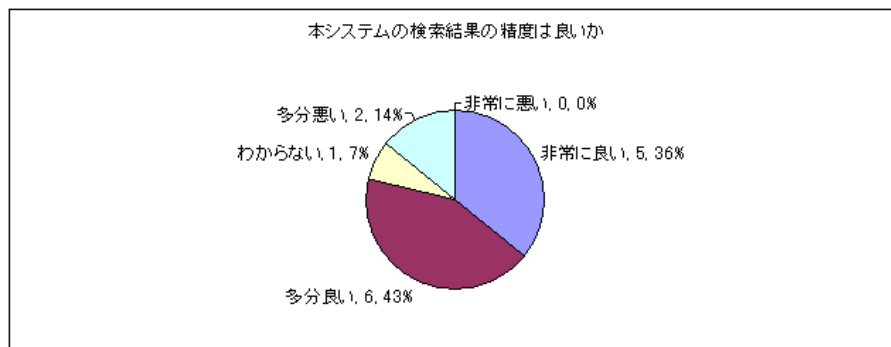


図 39: 本システムの検索精度に対する評価

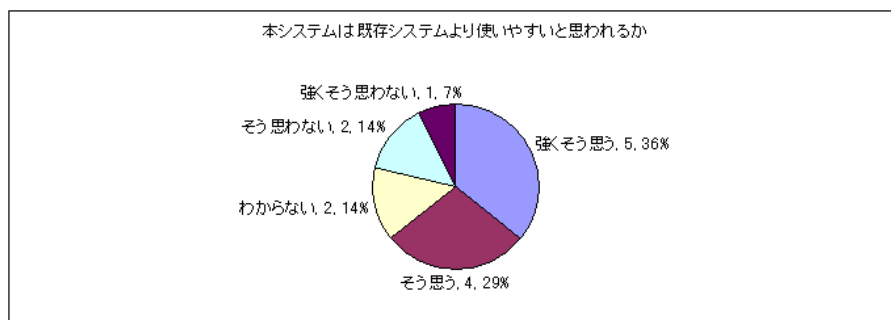


図 40: 既存システムと比較して本システムの使いやすさ

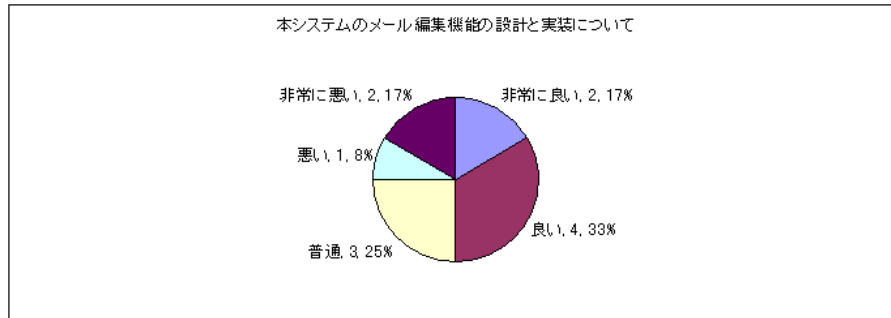


図 41: 本システムのメール編集機能の設計と実装について

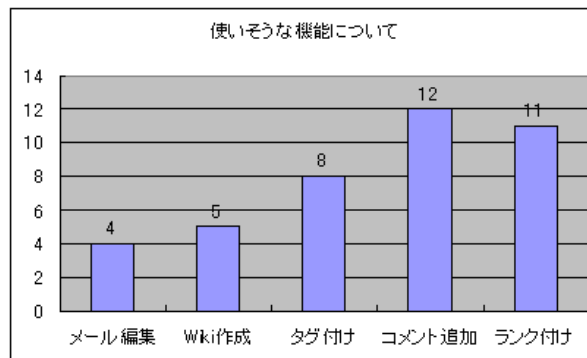


図 42: 使いそうな機能について