The University of Osaka
Institutional Knowledge Archive

Title	協同編集機能を持つメールアーカイブシステムの試作
Author(s)	チョイ, コー イー; 松下, 誠; 井上, 克郎
Citation	電子情報通信学会技術研究報告. SS, ソフトウェアサイエンス. 2006, 106(121), p. 25-30
Version Type	VoR
URL	https://hdl.handle.net/11094/26663
rights	Copyright © 2006 IEICE
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

Institute of Electronics, Information, and Communication Engineers

社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
SS2006-24 (2006-06)

協同編集機能を持つメールアーカイブシステムの試作

チョイコーイー† 松下 誠† 井上 克郎†

† 大阪大学 大学院情報科学研究科
〒560-8531 大阪府豊中市待兼山町 1-3
E-mail: †k-choy@ist.osaka-u.ac.jp

あらまし 既存のメールアーカイブシステムは単純な検索手法だけを提供しているため，開発者はアーカイブから必要な情報を効率よく取得できない．我々はメールアーカイブシステムに協同編集機能を導入し，開発者にメールの品質管理や適切な分類などの作業を支援する機能を提供することによって，その問題を解決できると考えた．そこで，本研究では，協同編集機能を持つメールアーカイブシステムの試作を行う．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ルアーカイブを用いて試作したシステムの評価を行った結果，開発者が試作したメールアーカイブシステムを利用することで，必要な情報をより効率的に取得することができることを確認した．

キーワード オープンソースソフトウェア開発，メールアーカイブ

Implementation of Mail Archive System with Collaborative Editing

Kho Yee CHOY†, Makoto MATSUSHITA†, and Katsuro INOUE†

† Graduate School of Information Science and Technology, Osaka University
1-3 Machikaneyama-cho, Toyonaka, Osaka 560-8531, Japan
E-mail: †k-choy@ist.osaka-u.ac.jp

Abstract Existing mail archive system provides only basic search functions. Therefore, developers cannot retrieve useful information from the archive efficiently. In this research, we implemented a mail archive system with collaborative editing functions, which let developers manage the quality of mails stored in the archive and group them more appropriately. We hope that the precision of the search results can be improved. Besides, a mail archive which is actually used in an open source software development project was used in an experiment to evaluate the proposed methods and we obtained satisfying results from the experiment.

Key words open source software development, mailing list archive

1. まえがき

オープンソースソフトウェア開発では，開発者間の意思疎通のためにメーリングリストが利用される．バグの報告や機能の実装などに関する情報がメーリングリストに送信され，メールアーカイブに保存される．これらのバグ報告や実装の詳細情報は開発者にとって有用である．例えば，ある開発者がソースコードを見て挙動に問題がある箇所を発見したが，実際にバグであるか確信を持てない場合に，その部分の実装に関する議論を参照することで，理解を深め判断することができる．従って，開発者がこのアーカイブから必要な情報を効率よく取得できれば，開発効率を向上させることができる．

ところが，メールアーカイブに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．メール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，開発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，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，内容の間違ったメールや実際の開発作業に参考にならないメールも存在しているため，上記の選別作業は更に困難になる．

一方，最近では，インターネットなどを通して利用者が協同に編集，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 [3]．協同編集作業では，ネットワーク上のどこからでも，いつでも，誰でもサーバ上にある文章を編集し保存することができるという特徴を持つ．この特徴のため，誤った情報を素早く発見したり，古くなった情報を容易に更新したりすることができる．このような協同編集機能をメールアーカイブシステムに導入すれば，メールアーカイブ内のメールの内容を更新したり，メールの検索精度を上げたりす

NII-Electronic Library Service

Institute of Electronics, Information, and Communication Engineers

るができると考えた．

そこで，本研究では，メールアーカイブから開発者に必要な情報を効率よく届けるために有用と思われる協同編集機能を提案し，提案した機能を持つメールアーカイブシステムの設計と試作を行った．開発者はシステムが提供する協同編集機能を利用して，メールアーカイブにあるメールを編集したり，評価したり，分類したりすることによって，メールの品質と検索精度を向上させる．

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ルアーカイブを用いて，試作したシステムの評価を行った．その結果，本システムを用いることで，開発者がメールアーカイブから必要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．

以降，2. 節では，オープンソースソフトウェア開発と，メーリングリストとメールアーカイブ及び既存の問題点について説明する．3. 節では，協同編集システムについて説明し，提起した問題点を解決するために提案した協同編集機能について述べる．そして，試作したシステムを利用する際の操作の流れを説明し，実際の利用例を挙げる．4. 節では，試作したシステムについて説明する．そして，5. 節では，システムの評価結果を報告する．最後に，6. 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開発環境

本節では，研究の背景となるオープンソースソフトウェア開発とメールアーカイブ，メールアーカイブシステム，そして協同編集システムについて説明し，既存のシステムが持つ問題点について説明する．

2.1 オープンソースソフトウェア開発の概要

オープンソースソフトウェア開発とは，ソースコードやドキュメント等をインターネットに公開し，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開発作業を行う開発手法である [1]．インターネットを用いて開発者はいつでも自由にオープンソースソフトウェア開発作業に参加することができる．近年では，この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ため注目を集めている．ただし，オープンソースソフトウェア開発の成功には，開発者間の意思疎通が重要な課題となる [2]．

2.2 メールアーカイブとメールアーカイブシステム

オープンソースソフトウェア開発では，世界中に分散した開発者間の意思疎通のためにメールが利用されている．メールの配信と管理を簡単にするために，メーリングリストが利用される．メーリングリスト宛に送られたメールは参加者全員に配信されると同時に，アーカイブに保存される．本稿ではこのアーカイブをメールアーカイブと呼ぶ．

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場合がある．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．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．その例を図 1 に示す．

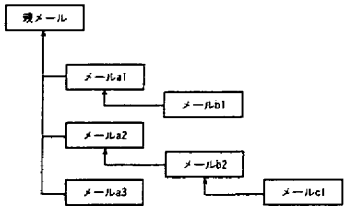

図 1　スレッド構造

開発者は検索システムを利用してメールアーカイブ中のメールを検索し，開発に必要な情報を取得することが可能である．このようにメールアーカイブを対象に検索などの機能を提供するシステムを本稿ではメールアーカイブシステムと呼ぶことにする．

2.3 協同編集システム

協同編集システムとは，インターネットなどを利用して利用者が協同して編集，レビューを行うことでシステム内の情報の品質を上げることを支援するシステムである．協同編集システムの代表的なものとして，Wiki [3] が挙げられる．Wiki は，様々な機能を提供するシステムの部分と，その中に管理されている文章から構成される．以降，そのシステムを Wiki システムと書く，中にある文章を Wiki ページと呼ぶ．

Wiki システムの実装方法によって実際に提供される機能は異なるが，その共通点としては以下のような特徴がある．

- ネットワーク上のどこからでも，いつでも，誰でも Wiki ページを書き換えて保存することができる．
- Web ブラウザだけを用いて Wiki ページを書き換えることができる．
- 簡単に Wiki ページ間にリンクを張ることができるため，高度に関連しあった Wiki ページ群を容易に作成できる．

オープンソースソフトウェア開発では，Wiki システムはドキュメンテーションの作成に利用されることが多い．

2.4 問　題　点

本節では，既存のシステムが持つ問題点について説明する．

メールアーカイブには，品質の低いメールが存在するため，開発者がメールアーカイブから情報を求める際，これらのメールを見分けて品質の高いメールを選出しなければならない．ここで情報の品質が低いというのは，そのメールの内容が実際の問題解決の参考にならないことを指す．例えば，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たメールが挙げられる．

また，メールアーカイブ内の各メールは 1 つの話題にしか対応を付けることができないため，1 つのメールの内容が複数の話題にまたがる場合，そのメールはどの話題に対応付けられるかは明白ではない．現状では，メールアーカイブの管理者は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，メーリングリストの参加者は自らの判断で適切だと思われる

NII-Electronic Library Service

メーリングリストに送信する．しかし，複数の話題についての情報を持つメールを探す際，開発者はどのメーリングリストから探せば良いのかはわからないため，効率的にメールを検索することができない．

さらに，開発者はメールアーカイブのメールから必要な情報を抽出する際，メールの内容だけが頼りになる．メールの内容が不正確であれば，開発者は間違った判断をしてしまいかねない．

3. 提案システム

本節では，2.4 節で説明した既存の問題点を解消するためにメールアーカイブシステムに導入する協同編集機能を説明し，提案システムの検索機能について説明する．その後，提案システムを利用する際の操作の流れを説明し，実際の利用例を挙げる．

3.1 協同編集機能

本節では，2.4 節で提起した問題点を解決するために，メールアーカイブシステムに協同編集機能を導入することを提案する．具体的には，メールの品質改善のためにメール編集機能を，メールの分類のためにタグ付け機能を，メールに本文以外の情報を提供するためにランク付け機能とコメント追加機能，新規 Wiki ページの作成機能を提案する．以下，それぞれの機能について説明する．

メール編集機能

開発者がメールを編集することで，メールの内容の品質を改善することができる．この機能で，例えば無効になったリンクなどの古い情報や不正確な情報を直すことができる．また，編集履歴を保存し，参照可能にすることで過去に行われた編集を確認することができる．

ランク付け

開発者がメールにランクを付けることによって，メールの内容の品質を評価する．開発者は閲覧しているメールは他の開発者にどのように評価されているかを知ることによって，その情報を信じるか信じないかを決めることができ，誤った判断を下すことを避けられる．また，メールのランクは検索精度の向上にも利用される．

タグ付け

タグとは開発者がメールを分類するために任意に付けることのできる文字列である．1 つのメールに対して複数のタグを付けることによって，1 つのメールを複数の話題に対応させることが可能になる．また，メールに付けられたタグは検索精度の向上にも利用される．

コメント追加

開発者はメールにコメントを追加することによって，メールの本文以外の有用な情報を他の開発者と共有することが可能になる．

新規 Wiki ページの作成

メールを閲覧していると，わからない用語に遭遇する場合はよくある．これらの用語の定義を新規の Wiki ページに書いておき，メールの本文にリンクを張ると，閲覧中のメールからすぐに用語の定義のページへ辿り着くことができる．これでその用語についてシステムを離れて別のシステムを利用し検索しなくてもよくなるので，余計な手間を省くことができる．

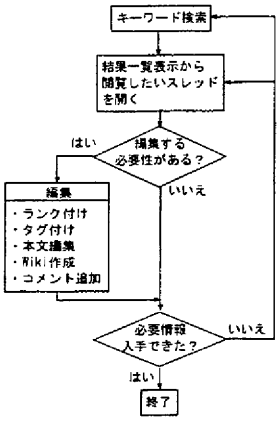

図 2 操作の流れ

3.2 検索機能

本節で本システムの検索機能について述べる．

本システムでは，メールアーカイブに蓄積されたメール・Wiki の内容に対して，入力したキーワードを用いて全文検索を行うことができる．メールアーカイブ内の全メールではなく，ある期間中に送信，作成されたメール・Wiki のみを検索することもできる．また，タイトルのみから検索を行うことも可能である．メール・Wiki が編集可能であるためメール・Wiki には多くのバージョンが存在しうるが，全文検索を行う際には，最初のバージョンと最新のバージョンだけを検索の対象にする．検索に該当する結果が複数存在した場合には，それぞれの結果のスコアを計算し，その値の大きい順に表示する．なお，本稿で検索結果というのは，検索に該当するメール・Wiki のスレッドのことを指す．メーリングリストでは，議論の流れはスレッド単位にまとまっているため，検索結果もスレッド単位にまとめるようにした．即ち，スレッド内の任意のメール・Wiki が検索に該当すれば，スレッド全体が検索の結果となる．

3.3 利用時の操作の流れ

本節では，上記の協同編集機能を導入したメールアーカイブシステムを利用する際の操作の流れを説明する．図 2 に操作の流れを示す．

まず，開発者はキーワードを用いてメールアーカイブを検索する．次に，検索結果の一覧から，閲覧したいスレッドを選択して開く．ここで，メール・Wiki の内容を見たり，協同編集機能を利用してメールの内容を編集したり，ランクやタグを付けたり，Wiki ページを作成したりする．

閲覧中のメールに必要な情報を取得でき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，最初から検索し直すことになる．

3.4 利用例

本節では，従来のメールアーカイブシステムを利用した場合の作業効率と，本システムを用いた場合の効率の差を例を用いて説明する．

NII-Electronic Library Service

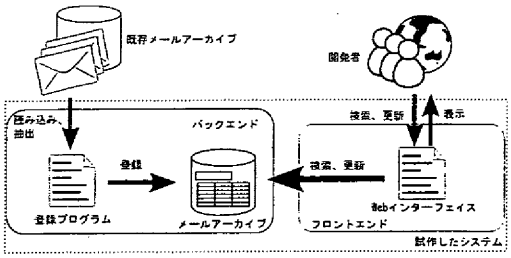

図 3　システムの概略図

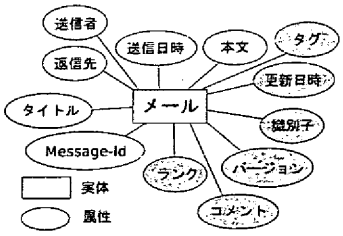

図 4　メールの属性

まず，従来のシステムを利用した場合を考える．開発者 A はある機能の実装について疑問を持ち，バグが存在していると疑いを持ったとする．従って，開発者 A はまず実装の詳細を理解するために開発者向けのメールアーカイブを検索してから，一般ユーザからのバグ報告を集めるために一般ユーザ向けのメールアーカイブを検索した．様々なキーワードの組合せで検索を行った結果，必要な情報を取得することができたが，検索の結果の中に，実装の詳細ではなく手法の有効性に関する議論が多かったため，かなりの手間がかかった．開発者 A は取得した情報から疑問を持った実装にバグはなかったことがわかった．

後に，開発者 B は開発者 A と同じ疑問を持った．開発者 B は開発者 A が行ったことを繰り返し，同じほどの手間をかけて，開発者 A と同じ結論に至った．

ここでわかったことは，同じ疑問を持った開発者は同じ操作を繰り返すため，必要な情報を効率よく取得できないことである．

次に，本システムを用いた場合，作業の効率がどのように改善されるかを考える．本システムを用いた場合にも，開発者 A は従来と同じように検索を行わなければならないが，開発者 A はその疑問を解決するために有用である情報を持つメールにタグを付け，メールのランクを高くすることができる．このようにしておけば，開発者 B，またはその後に同じ疑問を持った開発者はそれほどの手間をかけずに必要な情報を取得することができるようになる．また，メールにコメントを追加することにより，他の開発者に自分の出した結論を共有することもできる．

上記のように，提案した協同編集機能を利用することで，開発者はメールアーカイブから必要な情報をより効率よく取得することができるようになる．

4.　システムの試作

本節で，3. 節で提案した協同編集機能を持つメールアーカイブシステムの試作について説明する．

4.1　システムの構成

システムの構成を図 3 に示す．本システムは既存のメールアーカイブからメールを抽出して新たに編集可能なメールアーカイブを構築するバックエンドと，Web ブラウザで利用できるフロントエンドの二つの部分に分かれている．

バックエンドでは，登録プログラムが既存メールアーカイブに蓄積されたメールを抽出し，メールアーカイブに蓄える．本システムを利用するには，メール抽出の段階以外に既存のメールアーカイブを必要としない．

フロントエンドでは，開発者は用意した 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 3.1 節で説明した各編集機能を利用してメールアーカイブ内の情報を更新する．

以下で，バックエンドを構成する登録プログラムとメールアーカイブについて説明してから，フロントエンドの機能である協同編集機能と検索機能及びその Web インターフェイスについて説明する．

4.2　登録プログラムとメールの抽出

既存のメールアーカイブからメールを抽出する際，メールの性質を考慮しなければならない．ここでメールの仕様について簡単に説明する．

メールの仕様は主に RFC 2822 によって定められている．これらの RFC によると，メールはテキストファイルであり，ヘッダと本文の 2 つの部分に分かれている．ヘッダには，本文とは関係しない情報が「フィールド名:値」という形式で書かれている．メールはヘッダの Message-Id というフィールドの値によって一意的に識別することができる．

本研究で特に重要なメールの属性を図 4 に示す．ただし，影付けされたものは本システムで追加された属性であり，メールの本来の属性ではないため，メールの抽出段階では考慮しない．

登録プログラムは，図 4 に示した属性を抽出し，メールアーカイブに登録する．登録プログラムを定期的に実行することによって，既存のメールアーカイブからメールとその属性を本システムのメールアーカイブ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．

登録プログラムは以下の手順でメールの属性を抽出する．まず，Message-Id とタイトル，送信日時，送信者に関する情報は全てヘッダの各対応フィールドから取得する．なお，既に読み込んだメールを二度と読み込まないようにするために，Message-Id を利用してメールの識別を行う．次に，メールの返信関係を表すスレッド構造の再現に必要な情報を，メールのヘッダにある In-Reply-To フィールドから得る．例えば，メール 2 の In-Reply-To フィールドにメール 1 の Message-Id が記述された場合，メール 2 はメール 1 に対する返信であると判断する．

4.3　メールアーカイブ

メールアーカイブの実装にはデータベース管理システム MySQL を用いた．メールの情報を蓄えたり，メールの編集や

NII-Electronic Library Service

Institute of Electronics, Information, and Communication Engineers

それに関する情報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．これらのテーブルで図4に示したメールの属性を記録する．

4.4 協同編集機能

本節では本メールアーカイブシステムの協同編集機能ついて説明する．

本システムではメールの編集にあたって2.3節で紹介したWikiシステムを導入し，メールとWikiページを同じように扱うことにした．そのために，メールの持つ属性とWikiページの持つ属性を統一した．ここで，メールの送信をWikiページの作成として考える．すると，図4のメールをそのままWikiページに置き換えても構わない．また，メールには返信の構造を表すスレッド構造があり，Wikiページにはそれに対応する概念がないが，Wikiページを返信先も返信となるメールもない単独なメールと考えれば，この違いはシステムの実装に全く影響しない．

便宜上，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．

メール編集機能

メール・Wikiの編集は，システムのWebインターフェイスを通して開発者が行う．各メール・Wikiはバージョン番号を持つ．編集されるたびにバージョン番号が上がり，このバージョン番号で同じメール・Wikiの異なるバージョンを識別する．

ランク付け

内容の良さによって開発者はメール・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のランクは，開発者が付けたランクの平均値である．

タグ付け

メール・Wikiには英数字からなる任意の文字列をタグとして付けることができる．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．

コメント追加

メール・Wikiに追加情報を他の開発者と共有したい場合，コメントを追加することができる．コメントを追加する際ユーザ名とメールアドレスを入力することができるが，強制ではない．

新規Wikiの作成

前に述べたように，本システムでWikiページはメールと同じように扱われる．その主な違いは，メールは既存のメールアーカイブから抽出されるのに対し，Wikiページは開発者が本システムを利用して新規に作成する点にある．また，メールはMessage-Idによって識別されるのに対し，Wikiページはページのタイトルによって識別される．

Wikiページを作成するために，まず既存のメール・Wikiの本文に，追加しようとするWikiページのタイトルをキャメルケース(Camel Case)で入力する．このタイトルを持つWikiページがメールアーカイブに存在しない場合，キャメルケースで記述したタイトルが新規Wikiページ作成画面へのリンクと変わる．新規Wikiページ作成画面でWikiページの内容を記入して登録すれば，キャメルケースで書いたタイトルがWikiページへのリンクに変わる．

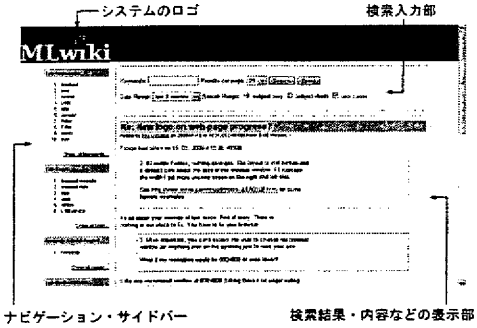

図5 本システムのメインインターフェイスの構成

4.5 検索機能

本システムでは，全文検索のエンジンとしてMySQLの全文検索機能を利用した．これから，検索結果のスコア計算について説明する．メール・Wikiに対するスコア，Xは式(1)を用いて算出する．

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

式(1)のS_{my}は，MySQLが内部に検索結果に付けたスコアである．Rは，開発者がメールに付けたランクである．N_{title}は，メール・Wikiのタイトルに含まれるキーワードの数であり，N_{tag}は，メール・Wikiに付けられ，キーワードにも含まれるタグの数である．

本システムでは，検索結果はスレッド単位で表示されるため，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．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，$\cdots$，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．

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

4.6 Webインターフェイス

WebインターフェイスはPHP言語で実装され，Webブラウザで利用する．メインインターフェイスを図5に示す．

検索入力部に検索キーワードを入力して，メールアーカイブを検索する．検索結果はスレッド単位でまとめられ，スレッドのトップだけが表示される．なお，検索結果は4.5節で説明したスコアの順に表示される．

検索結果の一覧から閲覧したいスレッドをクリックすると，メール・Wikiの単体表示画面に移る．ここでは，メール・Wikiの内容の他に，属するスレッドのスレッド構造と編集部，コメント，編集履歴が表示される．

図6に示した編集部では，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．編集画面とコメントの追加画面へのボタンも表示される．既存のメール・Wikiの編集と新

NII-Electronic Library Service

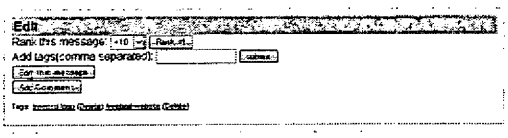

図 6　編集部の表示

表 1　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

規 Wiki ページの作成，コメントの追加は，それぞれの編集画面から行う．

5. 評　　価

本システムの検索精度を評価するために，実際のオープンソースソフトウェア開発で用いられた FreeBSD のメールアーカイブを対象に，検索結果について再現率，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 の実際の開発者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．

5.1 再現率，適合率及び f 値の測定

この評価では試作したシステムの検索精度を検証する．再現率，適合率及び f 値のいずれも高いほど優れた検索システムであるとされる．

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，本システムの両方で検索を行い，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，適合率を求め，f 値を計算した．

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．既存の全文検索システムには，Namazu を利用した．

実験結果は表 1 に示す．実験の結果から，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．

5.2 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 の開発者にオンラインアンケートを通して評価してもらった．

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 という一般利用者向けのメーリングリストの参加者に 3 週間にわたって公開された．回収できたアンケートは 14 件であった．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．本システムの提供する機能の中，実際に使おうとする機能を選んでもらった．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者がその半分を超えたが，「メール編集」と「Wiki 作成」を使おうとする者は少なかった．Wiki の作成には知識が必要なため，使用するのが難しかったと考えられる．一方，メール編集をしようと思う者はそれよりも少なかった．アンケートに書かれていた理由を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，下記の解決方法がアンケートに提案された．

- 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利用者登録などの仕組みで登録した者のみが編集できるようにすべき．

アンケートの結果から，メール編集機能に戸惑う者が多かったが，その他の機能は実用的であることがわかった．

6. ま と め

本稿では，オープンソースソフトウェア開発で用いられるメールアーカイブから必要な情報を効率よく入手することを支援することを目的とした協同編集機能を持つメールアーカイブシステムの設計と実装について述べた．

試作したシステムの実用性を評価するために，既存全文検索システムとの再現率，適合率と f 値の比較を行った．その結果，試作したシステムは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 の開発者に公開し，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感を持つ者が多いが，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可能にしたり，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．

文　　献

[1] Eric S. Raymond, "The Cathedral & Bazaar", O'REILLY, 2001.
[2] Sagers, G. W., McLure Wasko, M., Dickey, M. H., "Coordinating efforts in virtual communities: Examining network governance in open source", Proceedings of the Tenth Americas Conference on Information Systems, New York, 2004.
[3] Wikipedia, the free encyclopedia, http://www.wikipedia.org/.

NII-Electronic Library Service