Vol.2009-DPS-141 No.32
Vol.2009-GN-73 No.32
Vol.2009-EIP-46 No.32
2009/11/27

Webサービスとのマッシュアップを支援するメーリングリスト機構の提案

藤原啓輔[†1]　乃村能成[†1]　谷口秀夫[†1]

メーリングリストは，同好の不特定多数を結ぶツールとして発展してきた．このため，特定の話題についての議論や情報収集は，メーリングリストによって行われることが多かった．しかし，現在では，このような議論や情報収集は，Wiki や掲示板で行われるのが一般的となっている．一方で，メーリングリストは，小さな部署や特定のプロジェクトにおけるコミュニケーション手段や文書の共有スペースとして現在もさかんに利用されている．つまり，ML が大規模で不特定多数向けの利用から，小規模で特定のグループ向けの利用へと変化している．

にもかかわらず，現在一般に普及しているメーリングリスト管理システムは，旧来の大規模不特定多数向けメーリングリストの運用支援にとどまっているのが現状である．

そこで，本稿では，Web サービスとのマッシュアップを支援するメーリングリスト機構を提案する．これにより，小規模特定グループ向けメーリングリストと組織内の Web サービスの連携を可能とする．

Mailing List System Mash-up with Web Services

KEISUKE FUJIWARA,[†1] YOSHINARI NOMURA[†1]
and HIDEO TANIGUCHI[†1]

Mailing list have been a popular system to help people those who have the same predilection, such as Linux user group or academic SIGs. Therefore, joining a mailing list was a common route to discuss or gather information about specific topics. However, recent days, these type of mailing lists are shrinking and giving way to the Wiki or other web services.

Nevertheless, mailing lists are still frequently used by different groups in different ways; by small groups of specific solid projects for collaboration or storing documents. Mailing list management systems should change their style to meet the requirements of those small groups.

In this paper, we have proposed a new mailing list management system that helps small solid groups to connect their in-house web services with mailing lists by using mash-up technique.

1. はじめに

メーリングリスト (以下，ML) は，特定の話題に関心を持つグループなどで議論や情報交換を行うツールとして発展してきた．例えば，オープンソースの開発や使いこなしに関する ML がその代表例である．具体的には，GNU/Linux の開発や利用者向けの ML がある．参加者は，多い ML では，数千人を越える．

しかしながら，近年では，このような大規模 ML は減少し，既存の ML も流量が減少しつつある．その代わりに Web の掲示板や Wiki の設置が増加し，これらに議論の場が移されつつある．例えば，FreeBSD の開発プロジェクトは，2008 年 11 月 16 日付けのアナウンスで，Web ベースのディスカッションフォーラムを告知している[1)]．アナウンス中では，このフォーラムは，あくまでも ML の補完であるとしつつも，これまでにない動きである．また，Mandriva Linux 開発者向け ML(cooker ML[2)]) の流量は，2007 年まで年間 4-5 万通あったものが，2008 年 12 月の時点で 1 万 4000 通弱となっている[3)]．また，GNU/Linux の一般ユーザ向けの ML は，流量，加入者共に減少傾向にある．これには，様々な理由の説明があるものの，ML というツールそのものからの利用者離れであると指摘する声もある[4)]．

一方で，現在 ML は，小さな部署や特定のプロジェクトにおいてコミュニケーション手段や文書の共有スペースとしてさかんに利用されている．つまり，ML が大規模で不特定多数向けの利用から，小規模で特定のグループ向けの利用へと変化している．しかし，ML の機構自体は変化しておらず，こうした利用を支援するには機能が不足している．

そこで，本稿では，Web サービスとのマッシュアップを支援するメーリングリスト機構を提案し，小規模，特定のグループにおける ML の利用を支援する．

2. 小規模特定グループの ML 利用における要求と問題

2.1 ML の利用形態

本稿で取り扱う，小規模特定のグループによる ML の利用には，以下の 4 つ特徴がある．

(1) 参加者は少数で，メンバは互いに既知
ML に参加しているメンバは数人から数十人程度である．また，メンバはお互いを認

†1 岡山大学 大学院自然科学研究科
Graduate School of Natural Science and Technology, Okayama University

© 2009 Information Processing Society of Japan

Vol.2009-DPS-141 No.32
Vol.2009-GN-73 No.32
Vol.2009-EIP-46 No.32
2009/11/27

識しているおり，メールアドレスから個人を特定することもできる．

(2) 文書の共有スペースとして利用
議事録や会議資料をメールで送信することで，文書の共有を行う．

(3) コミュニケーションは ML 以外にも発生
情報の交換は ML 以外でも行われる．例えば，メンバ同士は直接会って会話や会議を行う，あるいは，SNS やグループウェアといった Web サービスを利用する．

(4) ML 利用頻度における大きな個人差
特徴 (3) で，コミュニケーションは ML 以外の Web サービスでも発生すると述べた．そうしたサービスを有効活用するメンバがいる一方で，Web サービスの利用を嫌い，連絡や議論，日程調整，文書共有などの全てを ML 利用だけで済ませようとするユーザがいる．こうしたメール偏重ユーザによって組織全体での Web サービスの利用が進まないことがあり，ML と Web サービスの両者の利点を活かせない一因となっている．

2.2 ML への要求

2.1 節で小規模特定のグループにおける ML 利用に関する特徴と，メール偏重ユーザによる組織内 Web サービスの利用停滞に関する問題をあげた．そこから，これらグループのための ML が備えるべき要素として，次の 3 つが考えられる．

(1) アーカイブを個人毎にカスタマイズし，機能強化を行いたい
旧来の ML 管理ツールは，個々のユーザの違いを意識したサービスではないため，提供するアーカイブも一律で，必ずしも組織や個人の利用形態に合致したものではなかった．ここで想定する ML では，メンバが小数で，特定可能なため，個人毎にカスタマイズ可能なアーカイブが実現可能である．また，文書管理スペースとしての利用を支援するため，添付ファイルを一括で管理したり，組織内の文書管理システムとの連携を図りたい．

(2) 他の Web サービスと連携を行う仕組みが欲しい
(1) でメールアーカイブから文書管理システムへの連携を行う例を挙げた．この他にも Web サービスからアーカイブ内のメール情報を利用する連携が考えられる．このため，アーカイブと Web サービスが双方向に連携を行うための API を機構内に持つ必要がある．

(3) メールを Web サービスへの入口としたい
組織内外に設置された各種 Web サービスは，本来の目的に特化されたツールであるため，必要に応じてメールではなく個々の機能を利用してもらうことが組織全体の仕事の効率を上げる．したがって，メールを入口としてメール偏重ユーザを Web サービスへと誘導し，組織全体での Web サービスの利用を支援することが重要である．

これら 3 つの要求に対処した新しい ML 機構が必要となる．

2.3 従来の ML 機構による対処

従来の ML 機構によって，2.2 節で挙げた要求への対処を考える．しかし，従来の ML 機構は，不特定多数の利用における支援を目的として成立した．このため，要求への対処には，以下の 2 つの問題がある．

(1) アーカイブの支援機能が不十分である
従来のメールアーカイブは，不特定多数での利用を想定しているため，個人に合わせたアーカイブのカスタマイズ機能は，提供していない．また，文書の共有スペースとしての利用を想定していない．このため，添付ファイルの管理を行えても，こうした利用の支援は行えない．

(2) 他の Web サービスとの連携が困難である
ML 単体での運用を支援しているため，Wiki や掲示板，スケジューラといった Web サービスとの連携は考えられていない．

こうした点から従来の ML 機構では，2.2 節で挙げた要求にまったく対処できていない．このため，小規模特定のグループにおける ML の利用支援を行うのは難しい．

2.4 他の機構による要求への対処

従来の ML 機構以外で要求への対処方式として，メーラ側で同様の機能を実現する方式が考えられる．他サービスと連携を行うメーラの例として，Microsoft の Outlook[5] がある．Outlook は，メーラ機能以外に，住所録機能やスケジュール管理機能を提供している．また，Outlook が管理するスケジュールは，グループウェアを通じて他の Outlook ユーザと共有できる．こうしたメーラによって提案機構を構成した場合，Web サービスとの連携を含めて，各処理をユーザの手元にあるメーラ自身で実現する．このため，Outlook 以外のメーラを利用しているユーザは，利用するメーラを変更する必要がある．しかし，ユーザは一般に自身の使用しているメーラに愛着がある．このため，システムの移行に伴うメーラの変更は困難である．特にメール偏重ユーザはこうした傾向が強い．

あるいは Web メーラの例として，G-mail[6] や Yahoo!メール[7] がある．Web メーラは，各 Web サービスと連携が取りやすい．しかし，Outlook と同様の理由に加えて，メールそのものを外部の組織に預けることを問題とする場合もある．

© 2009 Information Processing Society of Japan

Vol.2009-DPS-141 No.32
Vol.2009-GN-73 No.32
Vol.2009-EIP-46 No.32
2009/11/27

他の機構によって，2.2 節で挙げた要求への対処を行う場合，要求の (1) と (2) を満たすことは可能である．しかし，要求の (3)，特にメール偏重ユーザの Web サービスの移行に関して対処できていない．

3. 提 案 機 構

3.1 提案機構の提供機能

2.3，2.4 節で述べた方式では，小規模特定のグループ向けの ML 利用要求を完全に満たすことが出来ない．そこで我々は，Web サービスとのマッシュアップを支援するメーリングリスト機構を新たに提案する．提案機構は，以下の 3 つの機能によって，2.2 節で挙げた 3 つの要求に対して，それぞれ対処を行っている．

(1) アーカイブのカスタマイズ機能
アーカイブされているメール情報を加工することによる他 Web サービスとの連携を考える．また，こうした加工機能をメンバが自由に追加，削除することでアーカイブのカスタマイズが可能となる．

(2) 他 Web サービスからの利用支援機能
メンバ管理やアーカイブの各機能に対して，それらを外部から利用するための API を提供する．これにより従来 ML 管理ツールに閉じていた情報と他サービスとの連携が可能となる．これは，ML からの一方的なやり取りではなく，双方向での情報交換を実現する．

(3) 他 Web サービスの利用支援機能
メール偏重ユーザを Web サービスへスムーズに誘導するために，メール本文にそのメールと関連する Web サービスの情報を挿入する．例えば，Web サービスへのリンクをメール本文末尾に付加する．このとき，ユーザはリンクを辿るだけで Web サービスを利用できるため，ユーザを負荷無く Web サービスへの誘導できると考えられる．また，こうした情報は，メール本文に挿入されているため，ユーザは利用しているメーラを変更する必要がない．

3.2 提案機構のシステム構成

3.2.1 システムの全体構成

3.1 節で提案機構が実現すべき機能を明らかにした．これに基づいて設計した提案機構の構成を図 1 に示し，以下に説明する．この構成は大きく分けて，以下の 3 つの群に分かれる．

(1) 主要機能群

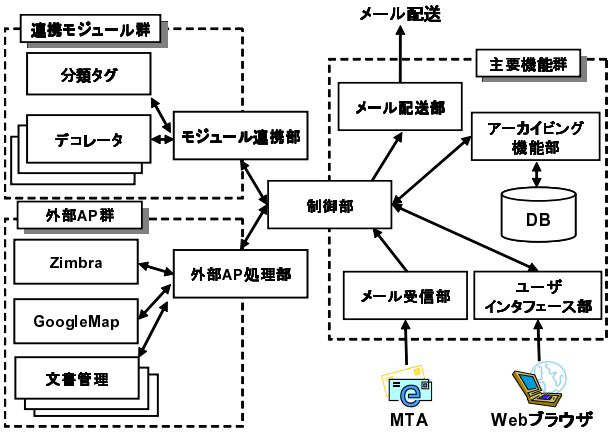


図 1 システム構成

主要機能群は，メール受信部，制御部，アーカイビング機能部，メール配送部，ユーザインタフェース部，DB(データベース) の 6 つの要素から構成され，メールの配送やアーカイビングを行う．また，通常の ML の運営に必要な新規 ML の作成や，メンバ情報の編集，アーカイブ閲覧用 Web ページの提供を行う．

(2) 連携モジュール群
連携モジュール群は，配送されるメールやアーカイブされているメールにタグ付けやリンクの挿入を行う複数のモジュールと，これらのモジュール群を管理するモジュール連携部によって構成される．

(3) 外部 AP 群
外部 AP 群は，メール情報をもとにマッシュアップを行う外部の AP とこれらの AP と情報のやり取りを行う外部 AP 処理部から構成される．

3 つの群はそれぞれ，制御部，モジュール連携部，外部 AP 処理部の 3 つの要素を通じて連携を取っている．これら 3 つについて以降の項で説明する．

3.2.2 制 御 部

制御部では，メールの配送時やアーカイブページの閲覧に合わせてモジュール連携部を利用してメールを加工し，それぞれ配送すべきメールや閲覧可能な Web ページを構成する．例えば，メール配送時において，制御部は，以下のように振舞う．(1) 配送すべきメー

© 2009 Information Processing Society of Japan

Vol.2009-DPS-141 No.32
Vol.2009-GN-73 No.32
Vol.2009-EIP-46 No.32
2009/11/27

ルをメール受信部から受け取る．(2) メールをアーカイビング機能部とモジュール連携部にそれぞれ転送する．(3) アーカイビング機能部は，メールをそのまま DB に保存する．(4) モジュール連携部は，メールにいくつかのデコレータ (後述) を適用してメール本文に情報を付加させる．こうしたメールに対する付加情報をここでは，アノテーションと呼ぶ．アノテーションの例として自身の保存先のアーカイブページの URL がある．(5) アノテーションを追加されたメールは，モジュール連携部から再び制御部に戻され，制御部からメール配送部を通じて各メンバに配送される．

また，アーカイブページの閲覧時においては，(1) ユーザインタフェース部からユーザの閲覧要求を受け取る．(2) 該当メール情報をアーカイビング機能部から取り出す．(3) 要求ユーザに合わせて必要なデコレータを選択し，その選択情報をメール情報と共にモジュール連携部に転送する．(4) モジュール連携部は，受け取ったメール情報を加工し，再び制御部に返す．(5) 制御部は，加工されたメール情報をユーザインタフェース部を通じてユーザに提示する．

その他，外部 AP 処理部と連携を行う場合，外部からの要求に合わせて，ML メンバの情報やアーカイブされたメールといった情報を外部 AP 処理部に渡す．

3.2.3 モジュール連携部

モジュール連携部は，各デコレータを管理する．デコレータとは，メールおよびメールのアーカイブページに，加工 (デコレーション) を行うモジュールの集まりである．例えば，配送するメールの本文にアーカイブページの URL を追記するデコレータがある．この他，アーカイブされているメールにデコレーションを行うデコレータがある．これらのデコレータには，特定の文字列に反応し，タグ付けやリンクの挿入を行うものがある．

3.2.4 外部 AP 処理部

外部 AP 処理部は，外部 AP と制御部との情報交換，あるいは制御部から外部 AP へのアクセスを行う際の窓口となる．外部 AP が取得する情報は，メンバの情報，アーカイブされているメール，および添付ファイルである．外部 AP 処理部は，外部に REST(REpresentational State Transfer) インタフェースを提供する．これは，アーカイブページといったユーザインタフェースは Web ベースである．このため，HTTP を用いて簡単に情報を授受できる REST と相性が良いと考えられるからである．

4. プロトタイプの実装

3.2 節で述べた提案機構の構成をもとにプロトタイプを実装した．プロトタイプは，我々の研究室内で開発し運用している Web サービス「LastNote」内の一機能として実装した．Postfix[8] をメールサーバとし，メールの受信と連動して配送メールへのアノテーションの付加やアーカイビングを行う．実装は，LastNote 本体と同じ Ruby on Rails[9] により行った．本プロトタイプでは，基本的な ML 管理に必要な新規 ML の作成やメンバ管理機能に加え，提案機構の特徴を具体化したいくつかの機能を実装した．以下の (a)〜(f) の 6 機能がそれに当たり，それぞれ (1)〜(3) の特徴に分類した．

(1) 「続きは Web で」を支援する機構
  (a) メール本文にアーカイブへのリンク挿入
  (b) 行事の出欠をメールから登録

(2) アーカイブの有効活用
  (c) プロジェクト管理ツール Redmine[10] との連携
  (d) Google マップ[11] との連携
  (e) LastNote 内文書管理システムとの連携

(3) 他 Web サービスとの連携用 API
  (f) Zimbra Collaboration Suite[12] との連携

以降で，各機能の詳細について説明する．

4.1 「続きは Web で」を支援する機構

メーラが受け取ったメールを基に，ユーザを Web サービスに誘導する機能の例として
  (a) メール本文にアーカイブへのリンク挿入
  (b) 行事の出欠をメールから登録

の 2 つを紹介する．これら 2 つの機能により，利用者は，メール本文に挿入されるリンクを辿るだけで Web サービスを利用できる．これにより，ユーザは，メール本体を Web サービスの一部であるかのようにみなすことができるため，Web サービスをスムーズに利用できる．

4.1.1 メール本文にアーカイブへのリンク挿入

まず，メール本文に挿入するアーカイブへのリンクについて，図 2，図 3 を用いて説明する．ML の各メンバに配送されたメールには，自動的にアーカイブ閲覧 Web ページの当該メールに対するリンクが挿入される．リンク先を辿ることでアーカイブページに移動できる．アーカイブページを通して閲覧するメールには，アーカイブが提供する各機能へのリンクが張られており各 Web サービスとの連携の窓口になっている．

例えば，図 3 は，議事録として送信されたメールをアーカイブページから文書管理システ

© 2009 Information Processing Society of Japan

Vol.2009-DPS-141 No.32
Vol.2009-GN-73 No.32
Vol.2009-EIP-46 No.32
2009/11/27

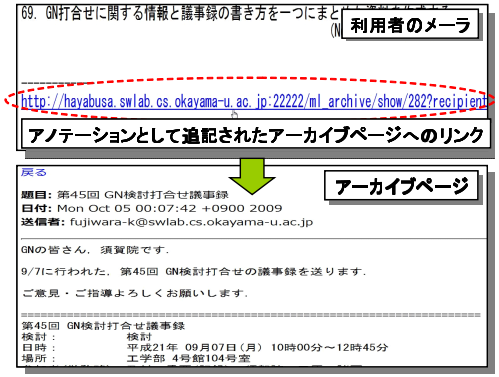


図 2　アーカイブページへのリンク

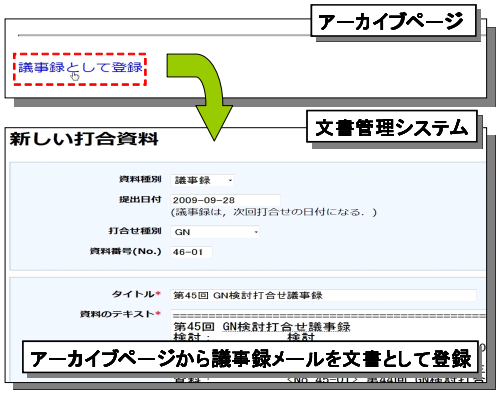


図 3　議事録の登録

ムへ登録する機能の例である．このようにメーラからアーカイブページ，アーカイブページから他の Web サービスへとユーザを誘導することできる．

4.1.2　行事の出欠をメールから登録

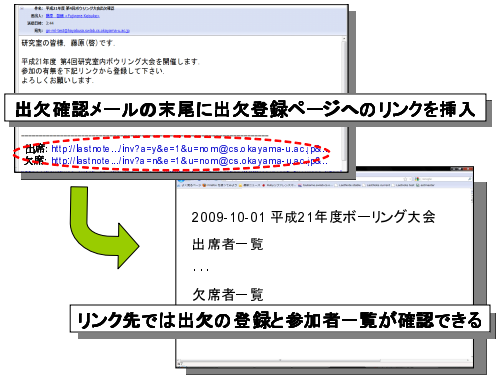


図 4　出欠登録用 Web ページへのリンク

デコレータは，リンク (アノテーション) を挿入する際にメンバ毎の異なる文字列を返すことができる．そのため，利用者がリンクを辿ってアーカイブページを閲覧する際に，制御部は，利用者が読んでいたメールや利用者のアカウント情報を得ることができる．この情報を利用することで，様々な機能が提供できる．

例えば，行事ごとの出欠確認を容易に行うことが考えられる．その様子を図 4 に示す．この出欠確認メールの末尾には，自動的にリンクが挿入されている．行事の参加者は，メールの出席か欠席のどちらかのリンク先をクリックする．これでリンク先の出欠管理システムに出欠情報が登録される．

4.2　アーカイブの有効活用とカスタマイズ

3.2.3 項で，デコレータについて説明した．本項では，そうしたデコレータの中から特定の文字列に反応し，リンクを挿入するデコレータと Web サービスとの連携例を紹介する．紹介する連携の例は，以下の 3 つである．

(c) プロジェクト管理ツールとの連携
(d) GoogleMap との連携
(e) 文書管理システムとの連携

© 2009 Information Processing Society of Japan

Vol.2009-DPS-141 No.32
Vol.2009-GN-73 No.32
Vol.2009-EIP-46 No.32
2009/11/27

4.2.1 プロジェクト管理ツールとの連携

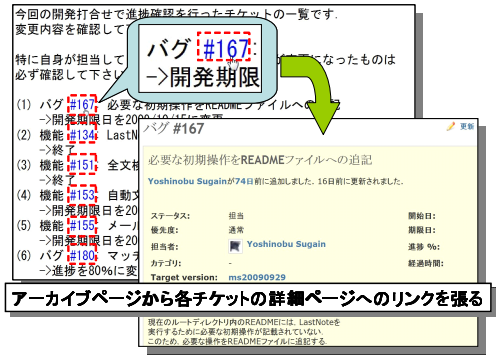


図 5 アーカイブページから Redmine のチケット詳細ページへのリンク

Redmine との連携機能を例に，デコレータの追加方法を示す．Redmine とは，オープンソースのプロジェクト管理ソフトウェアである．プロジェクト毎のタスク管理やアクセス管理を Web ベースで行っている．Redmine は，タスク管理の際，仕事をチケットと呼ばれる単位で分割している．チケットには，タイトル，仕事の内容，担当者，終了期限，および進捗情報といった情報が含まれている．また，チケット作成時に個別の数字が割振られる (例 : 10，150)．Redmine では，チケット固有の数字と他の数字を区別するため，チケット固有の数字を「# + 数字」で表記している (上記の例では，#10，#150 となる)．このため，Redmine によってプロジェクト管理を行う場合，グループ内でのメールにチケットに関する内容が記述されることが多い．その際，チケットは，Redmine 内の表記に則って，「#数字 チケットのタイトル」あるいは,「#数字」で記述される．また，こうしたチケットの詳細は Redmine 内のチケット詳細ページから確認でき，進捗変更や担当者の変更といった操作もこのチケットの詳細ページから行う．このため，メールアーカイブから「#数字」で表記されている文字列に各チケットの詳細ページへのリンクが張ってあれば非常に便利である．チケットの詳細ページの URL は，

`http://redmine.example.com/issues/show/`数字

となる．このため追加するデコレータは，以下の処理を行う．

(1) #数字 で記述された文字列を検出する

(2) (1) で検出した文字列にチケット詳細ページのリンクを挿入する．
挿入するリンクは，
http://redmine.example.com/issues/show/xxx
xxx は，検出した文字列から#を除いた数字となる．

これにより図 5 に示すように，アーカイブページに Redmine のチケットへのリンクが張られる．

本方式は，http:// から始まる文字列にリンクを張ること以外にメーラでの特別な処理を要求しない．これは，ほぼ全てのメーラが受け入れられる妥当な要求であるといえる．こうした連携を行う場合，メールそのものに何らかの構造を持たせ，メーラがそれに対応するソフトウェアモジュールを持つことで，受け取ったメールを直接処理することも考えられる．しかし，ユーザに専用のメーラ利用を迫るため，2.4 節で述べた問題がある．また，上記のようにメーラからアーカイブ，アーカイブページから目的のページへ，と遷移を行うだけなら，ユーザの操作は，1 つ余分にリンクを辿る程度で済むので，利便性を損っていない．

4.2.2 Google マップとの連携

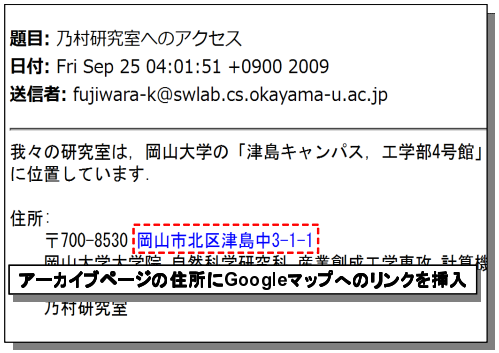


図 6 Google マップとの連携

Google マップとの連携を行う場合も，反応すべき文字列 (正規表現) が複雑になるだけで，Redmine の例と変わらない．デコレータは，日本語で書かれた住所に反応し，該当住所に対するリンクを挿入すればよい．

4.2.3 文書管理システムとの連携

文書管理システムとの連携について説明する．Redmine の例と同様にメール中の資料番

© 2009 Information Processing Society of Japan

Vol.2009-DPS-141 No.32
Vol.2009-GN-73 No.32
Vol.2009-EIP-46 No.32
2009/11/27

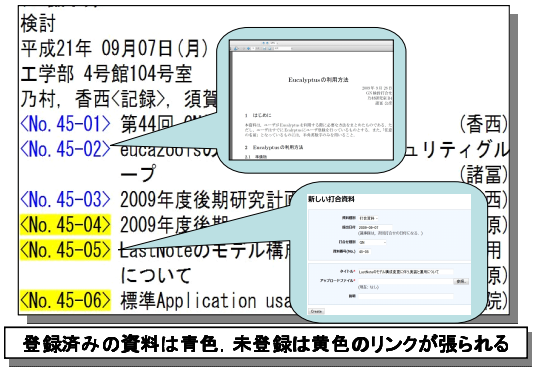


図 7　アーカイブから文書管理へのリンク

号を示す文字列にリンクを張ることは同様である．ここでは，更なる連携の例として，文書管理システムが持つ情報までリンクに反映させるところが異なる．

ここで連携する文書管理システムは，資料を登録する際に登録者の所属プロジェクトや資料番号の情報を要求する．資料番号は，No.xx-yy (xx と yy はそれぞれ数字) の形式をしており，この番号から，あるプロジェクト内の資料を一意に特定できる．また，会議の議事録中では，この資料番号は，⟨⟩ を補って，⟨No.xx-yy⟩ と表記される．

図 7 は，議事録メールに書かれている資料番号にリンクを張るデコレータの例である．リンク先は，登録された資料の閲覧ページとなっているが，議事録で言及があるにもかかわらず資料が未登録の場合，資料番号へのリンクが黄色で示される．この黄色のリンクを辿ることで文書管理システムの文書登録ページへと誘導する．この時，メール内容から登録に必要な情報を抽出し，文書登録画面に移動する．ユーザは登録する文書をアップロードするだけでよい．

この例では，デコレータは，資料番号に該当する文字列を取り出し，ML の所属するプロジェクトの情報と合わせて，文書管理システムに該当の資料があるか問い合わせる．問い合わせた結果に基づいて，異なる動作をするリンクを生成する．このようにデコレータに反応する文字列，抽出した文字列に対する処理を変えることで，様々な連携を行える．

4.3 Zimbra Collaboration Suite との連携

他の Web サービスとの連携用 API と外部 AP の連携の例として，Zimbra Collaboration Suite 中の Web メーラ (以下，Zimbra) から提案機構の持つ ML メンバ情報を取得する機

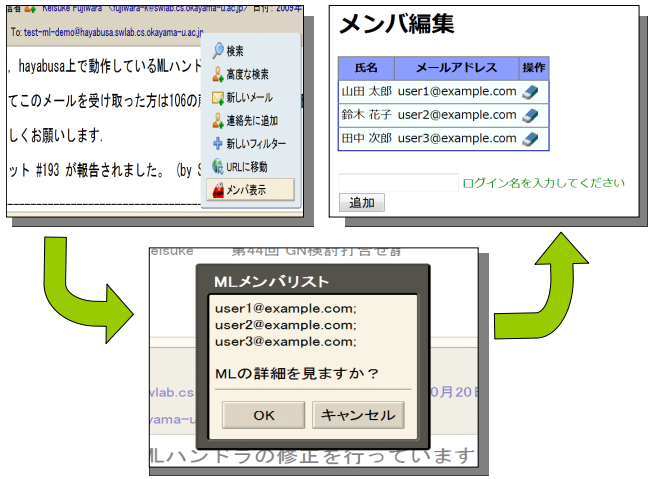


図 8　メンバ管理と Zimbra との連携

能について説明する．Zimbra は，JavaScript を追加することで容易に自身を拡張する機構を備えており，この JavaScript を Zimlet と呼んでいる．我々は，提案機構の API を利用した Zimlet を記述し，ML のメンバリストを Zimbra 上で直接確認できるようにした．本 Zimlet によるメンバ情報確認の様子を図 8 に示す．

この Zimlet はまず，メンバ情報を取得するメールアドレスの文字列を JSON 形式で提案機構に送信する．次に提案機構側で，受け取ったメールアドレスがシステムに登録されている ML のアドレスと一致するかどうか判定する．ML として登録されたアドレスならば，DB に登録されている ML の ID 情報と所属しているメンバのメールアドレスの一覧を JSON 形式で Zimlet に返す．こうして返信された情報を元に Zimlet は，取得したメンバのメールアドレスの一覧を表示する．また，詳細情報の閲覧を助けるために，提案機構のメンバ編集画面へと誘導する．

このように，提案機構は，連携用 API を通じて，これまで ML 内だけで閉じていたメンバの情報やアーカイブの情報を外部の AP に提供できる．これにより例えば，メール文中にあるスケジュール情報を抽出し，外部のスケジューラに登録するといった連携が可能となる．

© 2009 Information Processing Society of Japan

Vol.2009-DPS-141 No.32
Vol.2009-GN-73 No.32
Vol.2009-EIP-46 No.32
2009/11/27

5. 関 連 研 究

提案機構以外にも ML，および ML に流れるメール情報を効果的に利用する研究が行われている．例えば，いくつかの研究では，メールに含まれる仕事情報をもとにタスク支援を行っている．グループ内で ML を利用する場合，メールが持つ情報に個人やグループごとの仕事情報が含まれることが多くなる．KASIMIR[13] は，メールアドレスや件名，および本文中の特定の文字列をタグとして，半自動的にタスクの登録する機能とそのタスクを管理する機能を提供している．

また，ML アーカイブの利用において，特定の議論内容に関する経過を追跡したい場合がある．この場合，いわゆるスレッド表示がよく利用されるが，参照関係の把握において漏れも多い．msgcab[14] は，従来のスレッドを用いた方式に加え，メールの全文検索結果を組み合わせることで，議論の追跡可能性の向上を目指している．

また，qwikWeb[15] は，ML と Wiki との連携を目指した ML 管理システムである．qwikWeb は，ML に投稿されたメールを Wiki 上に蓄積する．蓄積された Wiki ページは，もとになるメールの件名をページ名として持ち，同じ件名のメールは，1 つのページにまとめられる．Wiki ページは，メンバによって直接編集できるため，メールによらない不定形な情報を柔軟に融合できる．グループに新しく加わったメンバは，Wiki ページを参照することで，議論の経緯を概観できる．こうした連携は，アーカイブの拡張機能として非常に有益であると考えられる．

提案機構では，これらの研究成果を外部連携する対象ととらえることができるため，連携 API を調整することで，メンバ支援をさらに強化できると考えられる．

6. まとめと今後の課題

本稿では，小規模，特定のグループによる ML の利用を支援するため，Web サービスとのマッシュアップを支援するメーリングリスト機構の提案を行い，そのシステム構成，およびプロトタイプの実装について述べた．提案機構は，アーカイブされているメールと Web サービスとの連携を促進する．連携手法の 1 つとして，Web サービスからアーカイブ内のメール情報を利用する API を提供している．また，組織や個人の利用形態に適したアーカイブ機能を提供するため，自由に追加，削除可能なモジュール群を提供している．今後は，連携可能な Web サービスの拡大を目指した提案機構の拡張性向上を目指す．また，提案機構の有用性を示す評価を行う．

謝辞 本研究の一部は，それぞれ科学研究費補助金・若手研究 (B)(課題番号: 21700139) による研究費，及び日本電信電話株式会社 NTT サービスインテグレーション基盤研究所と国立情報学研究所の提供する研究設備，回線を活用した．ここに記して謝意を示す．

参 考 文 献

1) The FreeBSD Forums Admin Team, “Official FreeBSD Forums,” FreeBSD-announce ML, http://docs.FreeBSD.org/cgi/mid.cgi?20081116160403.GC79046 , Nov. 2008.
2) “Mandriva Linux 開発者向け ML(cooker ML),” http://lists.mandriva.com/cooker/
3) Vincent Danen, “Sad metrics on community decline,” linsec.ca blog, http://linsec.ca/blog/2008/12/05/sad-metrics-on-community-decline/ , Dec. 2008.
4) 八田真行, “メーリングリストの死,” SourceForge.JP Magazine, http://sourceforge.jp/magazine/08/12/08/0850251 , Dec. 2008.
5) Microsoft Corporation, “Outlook,” http://office.microsoft.com/ja-jp/outlook/FX100487751041.aspx
6) Google Inc.,“G-mail,” http://desktop.google.com/ja/
7) Yahoo! Inc.,“Yahoo!メール,” http://mail.yahoo.co.jp/
8) GWietse Venema, “Postfix,” http://www.postfix.org/
9) David Heinemeier Hansson, “Ruby on Rails,” http://rubyonrails.org/
10) Redmine.JP, “Redmine,” http://redmine.jp/
11) Google Inc.,“Google マップ,” http://maps.google.co.jp/
12) Zimbra Inc., “Zimbra Collaboration Suite,” http://www.zimbra.com/
13) Olaf Grebner, Ernie Ong, and Uwe Riss, “KASIMIR -Work process embedded task management leveraging the Semantic Desktop,” Proc MKWI’08, pp.715-726, 2008.
14) 上野 乃毅，田中 哲，“手軽で柔軟な ML アーカイバ msgcab の設計と実装,” Japan Linux Conference 抄録集 第 4 巻, 2006.
15) 江渡 浩一郎，高林 哲，増井 俊之，“qwikWeb: メーリングリストと Wiki を統合したコミュニケーション・システム,” 情報処理学会研究報告. HI, ヒューマンインタフェース研究会報告，pp.5-11, 2004.

© 2009 Information Processing Society of Japan