グループウェアと
ネットワークサービス 47－8
（２００３．３．２０）

コミュニティを支援するメッセージ集約機構とその応用

坪井 創吾、原口 琢磨、後藤 和之、梅木 秀雄
sougo.tsuboi@toshiba.co.jp

概要 コミュニティの活動において、メーリングリストや掲示板などの電子的なグループコミュニケーションの手段は今や不可欠であるが、情報の共有や活用という点では効率的であるとは言えない。我々は、コミュニケーションの情報から、コミュニティ内の重要な情報を用途別に抽出してまとめ、更新などの管理を支援するメッセージ集約システムを開発した。本稿では、抽出プロセスを中心に本システムの機構を説明し、また効果的な利用方法や適用分野についても論じる。

A Collaborative Message Consolidation Mechanism and Its Application

Sougo Tsuboi, Takuma Haraguchi, Kazuyuki Goto, and Hideo Umeki

Abstract Group communication systems, such as mailing lists and electronic bulletin boards, have now become indispensable to the support of community activities. These systems alone, however, are inefficient in terms of community information sharing and leveraging. We have developed a message consolidation system that provides functions to extract the key information matching certain rules, reorganize them into auto-updated documents, and support the management of the resulted documents in communities. This paper describes how the system works with a focus on its extraction process, and discusses the effective usage and application.

1 はじめに

コミュニティの活動において、メーリングリストや掲示板、チャットなどの電子的なグループコミュニケーションの手段は今や不可欠である。企業内活動や顧客との関係においても、こうした電子的なコミュニケーションを介した情報収集、対応、意思決定を行う機会が増えてきている。こうした状況において、人同士のコミュニケーションシステムに必要な機能とは、もはや単純な通知や連絡といった「伝える」ことに留まらず、やりとりされた情報の蓄積（コミュニケーションログ）を様々な形で活用し、情報共有や意思決定を支援するものでなくてはならない。

ところが、コミュニケーションでやりとりされる情報は、断片的な内容が、急速に蓄積されるため、重要な情報が散在しがちで、結論や現状を把握することが難しい。つまり、このようなフロー型情報 (flow-type information) は、即時性や低い作成コストというメリットがある反面、静的な Web ページやデータベースなどのように単独で文書として扱うことのできるストック型情報 (stock-type information) に比べて、情報共有や再利用という面での効率性は低くなりがちである [1]。たとえば、ある会議通知の情報を含んだメールを見つけても、その後日程に変更がないかどうかを確かめるには、以後のメールを調べる必要がある。また、コミュニティ内でしばしば質問される事柄をメンバの誰かがまとめて一覧を作っても、メールで配信するのでは再び埋もれてしまい、共有可能な領域に文書のような形で載せたとしても、その内容の新鮮さを保つコストは高い。このため、コミュニケーション情報の共有や活用には、情報の鮮度と内容のまとまりという面において相補的な役割を持つフローとストックの情報をより密接に連携させることが必要であるといえる [2]。

これまで、我々のグループでは、ある議論と関係する文書（「まとめ」とよぶ）の作成を支援し、両者の対応関係を積極的に活用することで、効率の良い議論と文書の作成・洗練を促進させるシステム CIKLE [1] を開発し、社内での利用実験を行ってきた。その結果、「まとめ」がほぼ手動に頼ったものであるにもかかわらず、コミュニケーションの話題に対応した議事録の作成や、コミュニティ内で共有したい情報を「まとめ」として作成するなどの利用が行われた。しかし、まとめの新鮮さを維持するコストが依然とし

株式会社東芝 研究開発センター
Corporate R&D Center, Toshiba Corporation

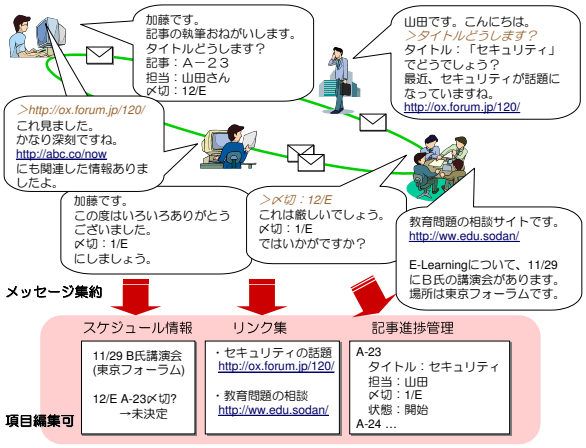

図 1: メッセージ情報集約機構の利用イメージ

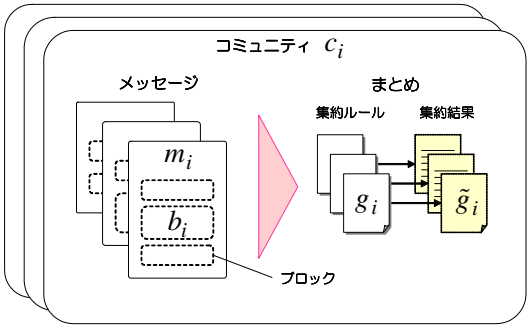

図 2: コミュニティシステムにおけるメッセージ情報集約機構の構成要素

て高く、メールとまとめに同じ情報が重複するなど、フロー型情報をストック型情報へ生かす仕組みの強化が望まれていた [2]。

従来より、フロー型情報から何らかの情報抽出を行う研究は行われてきている。典型的なところでは、フロー型情報の内容を少しでもまとまったわかりやすい情報として提示するもの [3,4] や、専門家が自然言語処理やパターンマッチ技術を駆使し、フロー型情報から特定の目的に特化したストック型情報を生成するアプローチ [5–7] などがある。

しかし、メッセージからどのようなまとめを作りたいかは、ユーザの目的により様々で、定型的なものだけでは不十分である。また、メッセージを文書として扱う単純な検索機能では、検索結果をコミュニケーションの経緯を含めた、まとまった情報として活用することは難しい。そこで我々は、コミュニケーションの情報から、ユーザの要求に合った情報をピックアップし、その情報を共有・維持管理するのに適した形式で提示するメッセージ情報集約機構を提案し、この機構を搭載したグループコミュニケーション支援システム GroupScribe を開発した [9]。本稿では、抽出プロセスを中心に具体的な例を挙げて説明する。

2　メッセージ情報集約機構の概要

メッセージ情報集約機構 (Message Consolidation Mechanism) とは、ユーザがメッセージ中から目的に合った情報を抽出してまとめる（集約する）と共に、その集約ルールと集約結果をコミュニティ内で共有し、編集するプロセスを支援する仕組みである。図 1 にこの機構の利用イメージを示す。このように、様々なタスクや情報が混在した一連のメッセージから、スケジュールや進捗情報といったユーザがまとめておきたい単位で情報を集約することができる。

図 2 に、メッセージ情報集約機構の構成要素を示す。コミュニティシステムにおいて、各コミュニティは、その中でやりとりされるメッセージの集合と、メッセージの集約結果としてコミュニティ内で共有される「まとめ」の集合をもつ。

各「まとめ」は、それを生成するための集約ルールのスクリプト (GroupScript と呼ぶ) が 1 対 1 で対応している。集約ルールには、「どの範囲のメッセージから、どういう情報を最終的にどのような形式で取得したいのか」を記述する。集約ルールをユーザレベルで記述できることが本システムの特徴の一つであり、メッセージの構造やメッセージ同士のリレーション情報などを用いて、抽出条件を柔軟に設定することができる。また、すでに作成されている集約ルールをカスタマイズして再利用することもできる。情報抽出プロセスは、メッセージが追加されるたびに行われ、「まとめ」は常に最新の情報を取り込んだ状態で管理される。

一方、抽出結果を編集するプロセスでは、たとえば、各まとめの抽出単位ごとに編集、削除、追加をコミュニティ内で行うことができる。また、その編集記録と関連づけて、編集理由や編集内容をメッセージとしてコミュニティ内に周知させることもできる。

3 情報抽出プロセス

3.1 集約ルール

集約ルールは、情報抽出と表現の方法を記述したもので、ユーザによって自由に編集できる。集約ルールの中は、大きく分けて以下の三つの要素から構成される。

- 抽出対象
- 抽出条件
- 表示レイアウト

抽出対象指定では、抽出処理の対象となるメッセージの範囲を指定する。指定可能な要素は、以下の通りである。

- コミュニティ（通常はそのコミュニティ内のみ）
- メッセージスレッド
- 送信日時（期間）

これらは AND もしくは OR の組み合わせで指定することができる。

ある条件にマッチする抽出結果は、エントリ (entry) と呼ぶ抽出単位ごとにまとめられ、各エントリはスロット (slot) と呼ばれる具体的な抽出項目を保持している。例えば、スケジュールに関する情報を抽出する際のエントリには、日時 (when)、イベント内容 (what)、場所 (where) などのスロットが含まれている。ここで使用できる情報は、後述するブロック情報、言語表現情報、リレーション情報である。エントリは通常、より細かい抽出条件を記述した抽出モジュールによって管理されており、ユーザは、スケジュール情報やリンク集、Q＆A集などの抽出モジュールと、必要に応じて、具体的なメッセージ中に現れる文字列を抽出条件に指定する。現在の実装システムでは、抽出された内容はデータベースに格納され、XML 形式でその内容を取得することができる。

表示レイアウトの指定では、抽出したエントリをどのように表示するかを指定する。情報は内容によって望ましい表示形式が異なる。リスト、表、カレンダーなどの表示形式から、表示件数、ソート方法に至るまでを XSLT スタイルシートを用いて管理している。ユーザはデフォルトで用意されているリストや表形式、カレンダー等のスタイルシートを選択するか、これらを import してカスタマイズを行った別シートを指定することができる。これにより、簡単かつ柔軟な拡張が可能である。

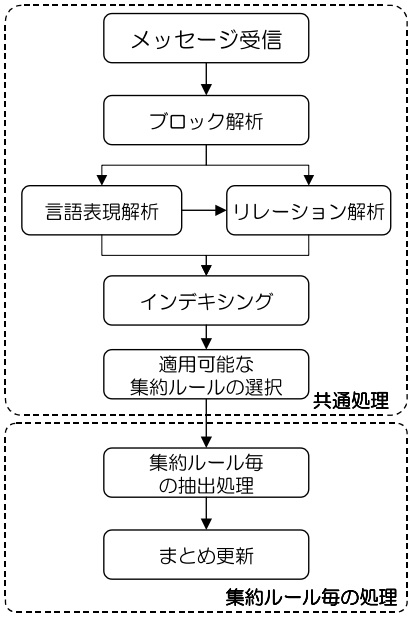

図 3: メッセージ集約の処理の流れ

3.2 情報抽出処理

ここでは、メッセージが投稿されてから、集約ルールに基づいて必要な情報を抽出するまでの過程を説明する。全体の処理の概略を図 3 に示す。処理は集約ルールに依存しない共通処理と集約ルール毎の抽出処理に分けられる。前者はメッセージからさまざまな特徴を抽出し、後者の抽出処理で使える「道具」を用意することが目的である。解析によって得られた情報は、インデキシングされ、リレーションや正規表現マッチングなどの柔軟なアクセスを提供する。

ブロック解析

メッセージの文章は基本的に自由記述であるが、ある程度慣習的な書き方が存在する。そこで、投稿されたメッセージを箇条書きや空行、タブなどのテキスト表示上のスタイルに基づいて、ブロックという単位に分割する。表 1 は、ブロックとして認識する種類を列挙している。ブロックは、抽出結果としてどのように扱われるべきかを示す抽出属性をもつ。抽出属性は、大きく分けて、メッセージのメタ情報 (M)、引用情報 (C)、引用以外の省略可能な情報 (O)、本文情報 (B) の４種類ある。ブロックは通常、空行によって分割されるが、空行のないところでも表 1 にあげた異なるブロックの種類が接していると判断した箇

ブロックの種類	(*name*)	**抽出属性**	内容
サブジェクト	(subject)	**M-s**	メッセージのサブジェクト
ヘッダ	(message-header)	**M-h**	メッセージのヘッダ
添付ファイル	(attachment)	**M-a**	メッセージの添付ファイルに関する情報 (ファイル毎)
末尾引用	(ending-citation)	**C-e**	末尾に引用されたメッセージ部分
挿入引用	(inserted-citation)	**C-i**	文中引用されたメッセージ部分
前書き、挨拶	(foreword)	**O-f**	例：「(名前) です」「こんにちわ」「各位」など
結びの言葉	(afterword)	**O-a**	例：「以上」「よろしくお願いします」など
引用情報	(citation-info)	**O-ci**	例：「Taro Toshiba wrote:」「○○が書きました：」など
署名	(signature)	**O-sg**	末尾の投稿者に関する情報
免責文	(disclaimer)	**O-d**	免責に関する定型文
区切り	(separator)	**O-sp**	「-----8<----8<----」「　　記　　」など
コマンド	(command)	**O-cm**	システムへの指示を行うための定義済み文字列
通常文	(plain)	**B-p**	その他の通常の文章
見出し	(heading)	**B-h**	例：「1.　はじめに」など
テーブル	(table)	**B-t**	マス目状に文字列を配置した段落
記号付き箇条書き	(itemization)	**B-i**	・, ○,- などの記号で始まる箇条書き
見出し付き箇条書き	(description)	**B-d**	例：「場所：A会議室」など
番号付き箇条書き	(enumeration)	**B-e**	例：「1. 従業員は 8:15 に出社...」など

表 1: ブロックの種類と抽出属性。M:メタ情報、C:引用、O:省略可能、B:本文。

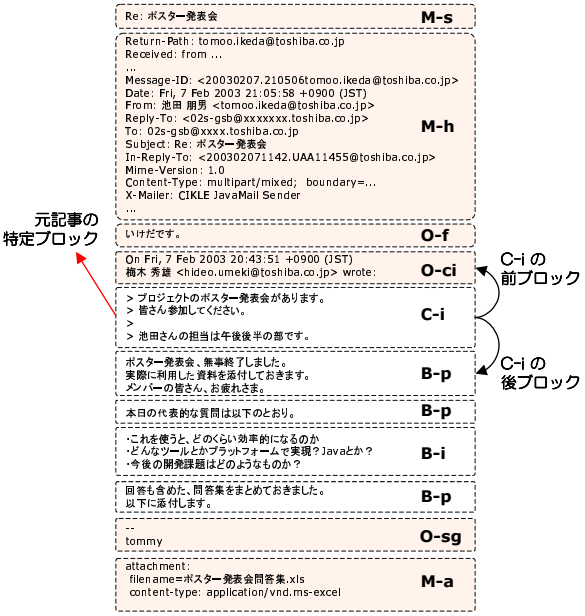

図 4: ブロック解析例

所で、二つのブロックに分割する。ブロック解析の結果例を図 4 に示した。波線で囲まれた単位がブロックである。ブロック右の記号は各ブロックの種類に対応する抽出属性を表している。

ブロック解析のアルゴリズムはヒューリスティックの占める割合が多く、本稿では詳しく述べない。

言語表現解析

情報抽出プロセスにおいて、ユーザが抽出条件として指定する場合、ユーザ指定のキーワードが含まれているかどうかは容易に指定できるが、「日時に関する情報が書いてあるメッセージ」を検索したいなど、単純なキーワードでは指定できない条件を記述したいことがしばしばある。例えば、「日時表現」「URL 表現」「場所の名前」「人の名前」などである。これらは複雑なパターンマッチングや何らかの辞書が必要となる。例えば、イベントなどの日程情報を抽出する場合、日時に対応する文字列には「2002 年 10 月 9 日」「10/9 (水)」「9 日」など、さまざまなパターンがある。言語表現解析部では、このような特殊な表現を「日時表現」といった言葉で指定できるるように管理している。

現在のシステムでは、日時表現および URL 表現についての表現パターンを実装し、該当ブロックと結びつけて記録している。日時表現判定に関しては、[8] を始め、様々なヒューリスティックが存在している。また、URL 表現の判定は RFC1738 の仕様に従った。図 5 に言語表現解析の実行結果例を挙げた。日時表現はある時点もしくは期間を取得することができ、URL は実際のその URL が生きているかどうかの情報と共に、ブロックと関連づけて記憶する。

図 5: 言語表現解析例

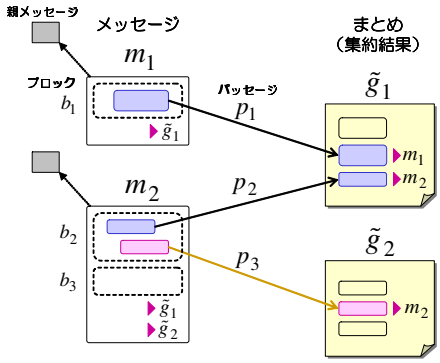

図 6: メッセージと集約結果（まとめ）間の相互参照関係

リレーション解析

リレーション解析では、メッセージのヘッダ情報やメッセージ本文での記事引用部分と他のメッセージなどと比較することにより、ブロック間の関係を関係づける。図 4 に示すように、引用ブロックは、他のメッセージ情の一つ以上のブロックと引用関係を持つ。引用ブロックと、その前後のブロックを何らかの関係で意味づけるという考え方もあるが、本システムでは文字列パターン上明らかなことに限定し、複雑な関係づけは行っていない。

集約ルールの抽出内容

メッセージが集約ルールの対象指定にマッチした場合、その集約ルールに記述された抽出処理が行われる。抽出処理の出力の単位はエントリと呼ばれる構造体であり、「まとめ」は集約ルールと、複数のエントリをもつ集約結果によって構成される。図 6 に示すように、各エントリには、メッセージ中の抽出条件にマッチした部分 (パッセージ; passage) が格納される。図中の各集約結果には複数のエントリがあり、各エントリには抽出元のメッセージの ID が格納される。また、メッセージに対しても抽出先の集約結果の ID が割り付けられ、相互に参照できるようになっている。

集約ルールの抽出内容には、エントリの中にどのようなスロットを用意するのか、そのスロットの中に何を格納するかを記述する。スロットにはテキスト、メッセージ、ブロックの集合が格納できる。この内容に従って抽出処理が実行され、まとめにエントリの内容が書き込まれる。集約ルールの記述に使える道具は、ブロック情報、用語情報、リレーション情報と、メッセージの返信関係による親、子、孫、祖先関係である。スロットには必須もしくは任意の属性が存在し、必須スロットに格納する情報が取得できたとき、１つのエントリとして成立する。ここで、メッセージからスケジュール情報を抽出する場合と、Q&A 形式の議論をまとめる場合について、各まとめで設定されるスロットと抽出条件について具体例を挙げる。

●スケジュール情報一覧の場合のスロット：

Date(必須) 本文属性 **B** のブロック内に一つの日時パターン情報 (DT-pattern) をもつ **B-p** ブロックまたは **B-d** ブロックの中の、DT-pattern の内容（テキスト）

Event(必須) Date スロットの取得ブロックより前の **B-p** ブロックの中に含まれる「(行います｜開催します｜会)を含む 1 文。それが存在しなければ **M-s** ブロック

Place(オプション) Date スロットの取得ブロックから距離が 2 以内の **B-d** ブロックの、見出しに「場所」を持つ内容（テキスト）

● Q&A 一覧のまとめの場合のスロット：

Question(必須) **M-s** ブロックの中に「(質問｜か$｜?)」という言葉が含まれるメッセージの冒頭の **B** ブロック

Answer(オプション) Question スロットのブロック、もしくは既存の Answer スロット内のブロックと引用関係にある **C-e** ブロックの後の連続した **B** ブロック。もしくは子の関係にあるメッセージの第一 **B** ブロック。

以上のような記述により、エントリの定義を行うことができる。これらはシンプルな内容だが、やり取りしている議論の形にそぐわない場合は、内容をカスタマイズすることで対応していけばよい。

実装システムでは、このような典型的な集約ルールをテンプレートとして用意した。テンプレートの内容をカスタマイズしていくことで、集約ルールもコミュニティの資産として活用できることが期待でき

る。また、表示形式としては、エントリの内容をリスト形式/表形式/カレンダー形式として表示する XSLT スタイルシートを用意している。

3.3 編集プロセスの支援

コミュニティの財産としてまとめを維持するためには、手動による編集作業を手軽かつ効果的に行えることが重要である。本システムは、まとめに対して、エントリの追加、既存エントリの削除および内容編集インターフェースを備える。

エントリの追加は、集約ルールで定義されたスロットの内容を手動で与えることができ、システムが抽出した情報と変わらない枠組みで、まとめに情報を追加することができる。これにより、集約ルールでは拾いきれなかった情報や、メッセージを介する必要のない情報を加える行為を支援する。

エントリの削除は、システムによる抽出／ユーザによる追加に関わりなく、エントリを削除する機能である。誤抽出した情報や内容が古くなった情報を削除し、まとめの内容を常に意味がある状態で保つことを支援する。

内容編集は抽出内容が冗長だったり、いくつかのスロットが抽出に失敗している場合等のフォローに用いられる。内容が編集された後、新規メッセージの投稿によってそのエントリ内のスロットに変化があった場合、システムはエントリの編集内容を確認した方がよい旨を提示する機能を持つ。これにより、編集内容が最新の情報を反映しているのかを確認することができる。

編集作業は作業履歴として記録され、コミュニティメンバは何がいつ修正されたかを確認することができる。また、必要な場合には編集内容や編集理由をメッセージとしてまとめや編集エントリに関わるメッセージと関係づけられて送信できる。

4 コミュニティ支援へ

このような情報集約機構を備えたコミュニティウェアを利用することによって、従来、連絡はメールで、結論やまとまったものは Web ページや他の文書管理システムなどと分けて管理を行うことなく、コミュニケーションを中心とした情報共有から管理までの統一的なインタフェースを提供することが可能となる。

また、CIKLE での課題として、「他人がコストをかけて作ったまとめは編集しにくい」というものがあったが、本機構では作成コストが非常に低いことや、情報の編集単位がエントリという小さなものなので、そのような意識が緩和されることを期待できる。

さらに、メッセージ集約の結果をコミュニケーションに活用することで、立場の違いによるメッセージ量のコントロールに利用できる。通常のメーリングリストで行っているダイジェストサービスの高機能化として、ある内容に関するまとめの更新が行われたときに限ってメールを受け取るなどの処理が可能となる。

メッセージ情報集約機構を組み込んだコミュニティウェア GroupScribe [9] は、４月から数千人規模での社内公開実験を行う予定である。この実験を通してシステムの有用性評価を行うと共に、集約ルールとメッセージ解析の洗練化を行い、概念辞書との連携や、スロットにセットされた内容へのテキスト自動要約技術の適用や、大規模辞書の利用を検討する。

参考文献

[1] 梅木秀雄, 笹氣光一, 福井美佳, 他. コミュニティベース知識協創プラットフォーム CIKLE. 第 62 回情報処理学会全国大会特別トラック (1) 講演論文集, pp.159-162 (2001).

[2] 梅木秀雄. コミュニケーションに埋もれた知識を活用するコミュニティウェア. 情報処理 2002, vol.43, No.10, pp1085-1092 (2002).

[3] 村上明子, 長尾 確. 引用に基づくオンラインディスカッションの構造化.「知識発見のための自然言語処理」シンポジウム (1999).

[4] 山見太郎, 村越 広享, 島津 明, 他. ICEMail++: 討議構造参照機能を有するメールクライアント. インタラクション 2001 pp39-40 (2001).

[5] 佐藤 円, 佐藤 理史. 電子ニュースのダイジェスト自動生成. 情報処理学会論文誌, Vol.36, No.10, pp.2371-2379 (1995)

[6] 乃村 能成, 花田 泰紀, 牛島和夫. MHC - Message Harmonized Calendaring System の設計と実装. 情報処理学会論文誌, Vol.42, No.10, pp.2518-2525 (2001)

[7] 長谷川 隆明, 高木 伸一郎. 文書構造の認識と言語の特徴の利用に基づく電子メールからのスケジュールと ToDo の抽出. 情報処理学会論文誌, Vol.40, No.10, pp.3694-3705 (1999)

[8] 長谷川 隆明, 高木 伸一郎. 電子コミュニケーションにおけるスケジュール情報の抽出. 自然言語処理シンポジウム「実用的な自然言語処理に向けて」 (1997)

[9] 原口琢磨, 梅木秀雄, 横田健彦. メッセージ集約型コミュニティウェア GroupScribe. インタラクション 2003, in press (2003)