

インターネットプログラミング言語による 情報組織化機構について

水内 祥晃[†] 篠原 誠[†] 田島 敬史[‡] 田中 克己^{*}

[†]神戸大学大学院自然科学研究科情報知能工学専攻

[‡]神戸大学工学部情報知能工学科

^{*}神戸大学大学院自然科学研究科知能科学専攻

本論文では、インターネットを通じて集められた情報を組織化する機構を提案する。まず第1段階として、電子メールを扱う組織化機構を提案し、実現し試作したシステムについて述べる。この機構は、電子メールをユーザが与えたキーワードやメイリングリストによりフィルタリングし、質問・返答の対へと構造化し、それらをハイパーテキスト化してユーザに呈示する。次に、扱う情報を WWW 上の検索エンジンより得られる結果に拡張したシステム的设计について述べる。この場合、サーバーは検索結果をクライアントに送り、クライアント側でこれを組織化し呈示する。クライアントはさらに、対話的にユーザの要求に応じて、表示形式の変更や再組織化を行うことができる。これらの対話的機能をクライアント側で実現するために本研究では Java 言語を用いている。

An Information Organizing Mechanism by an Internet Programming Language

Yoshiaki Mizuuchi[†]
Keishi Tajima[‡]

Makoto Shinohara[†]
Katsumi Tanaka^{*}

[†]Division of Computer and Systems Engineering,
Graduate School of Science and Technology, Kobe University

[‡]Dept. of Computer and Systems Engineering,
Faculty of Engineering, Kobe University

^{*}Division of Intelligence Science,
Graduate School of Science and Technology, Kobe University

This paper proposes a system that organizes information gathered via Internet. First, we propose and implement a prototype system that automatically organizes a user's e-mails, especially those from mailing lists. This system extracts only needed mails including keywords given by the user, organizes them into pairs of questions and answers, and show the result in the form of hypertexts. Next, we explain a design of a system that deals with results of search engines on WWW. In this case, the server extracts data and sends them to the client. Then the client organizes them and shows them to users. Furthermore, the client interactively reorganizes data or changes the representation of them in accordance with users' requirements. In our system, these interactive facilities of clients are achieved by Java applets.

1 はじめに

インターネットを通じて得られるメール、ニュース、メイリングリストによる討論、検索文書などの情報の特徴として、不特定多数のユーザから発信されるために、量が莫大であるとともに、ユーザの視点に応じた形では配布されないということがある。そのため、その莫大な情報の中から、必要な情報を抽出することは非常に困難な作業である。そこで、これらの未整理なデータをユーザの目的にあわせた形で分類、構造化して表示する機構が必要となる。このような情報の分類、構造化を「情報組織化」と呼ぶ。本稿では主に、メール、メイリングリスト、検索エンジンの出力結果を対象とした情報組織化機構の提案と、設計開発中の試作システムについて述べる。

通じて得る情報の代表的な例として、電子メールが挙げられる。電子メールは不特定多数のユーザから発信され、単純に到着時間順にユーザのメールスプールに格納される。大量の電子メールがスプールにたまった状態でユーザがメールを読みこんだ場合、ユーザに呈示されるのは無秩序に並んだメールの列である。ユーザはこれを順番に読んでいきながら、これらをそのユーザの目的に沿った形で分類し、またどのメールがどのメールに対するリプライなのかという対応関係をたどるという作業を行わなければならない。特に、複数のメイリングリストに参加していて、それらの大量のメールの中から、ある特定の議論のおおまかな経過だけを追いたい場合などは、ユーザが大量のメールを順に読んで行きその議論の流れを捜しさねばならず、ユーザの大きな負担となる。また同様な問題は、保存している過去にきたメールの中から、必要なメールだけを選び出したいという場合にも生じる。

そこで本研究では、このような処理を自動的に行う情報組織化システムを提案する。このシステムでは、ユーザに到着したメールを、ユーザがあらかじめ指定したキーワードに基づいて自動的に分類し、さらにメール間の参照関係に基づいて構造化された HTML 形式のハイパーテキストに変換してユーザに呈示してくれる。また、すでに読み込んで分類済みのメール群に対して、新たにキーワードを与え、分類しなおして表示させる事も可能である。

このような、電子メールの組織化に関するシステ

ムの例としては、Kamiya らによる Grassroots [3] がある。Grassroots は現在のインターネット上の情報収集手段である、WWW、電子メール、ニュースなどのシステムを統合することを目的としている。Grassroots では、各ユーザは公開された複数の folder を持ち、電子メールの送信は、「差出人による、受取人のある folder への情報の書き込み」として表現される。このため、メールは送られる時点で分類されている。この方法は、Grassroots が対象としている狭いグループ内の情報管理には有用である。しかし一般の電子メールを対象とする場合は、自分がどのようなメールボックスを所有しているかを不特定多数のユーザに公開することになり、また各メールをどこに分類するかを送り手側が決定するという点からも問題がある。

議論の流れを整理するためのツールとしては、オンライン上での会議を支援するシステムである gIBIS [1] が挙げられる。gIBIS では会議中に発言を行う際に、「この意見への賛成」、「この意見への反対」などの属性をつけることにし、これらの属性を使って会議の流れをグラフィカルに表示する。われわれのシステムは、このアプローチを電子メールの組織化に適用している。

また、受け取ったメールをハイパーテキストに変換し保存するシステムとして、Msgarch [6] というシステムが提案されている。しかしこのシステムはメールを HTML 化し表示することに重点を置いているため、本稿の意味での組織化は行っていない。

次に、前述の電子メールの組織化のシステムの発展として、インターネット上に分散しているデータを扱う場合を考える。その代表的な例が World Wide Web (WWW) である。WWW 上で欲しい情報のみを検索するツールとしては、既に様々な検索エンジンと呼ばれるツールが提供されている。しかし多くの場合、検索エンジンのキーワード検索の結果は不要な情報をも含んでおり、さらなる絞り込みが必要となる。そのため、現状のように検索結果をただ羅列するだけでは不十分であり、これらの情報を組織化して呈示し、さらにユーザが対話的に絞り込みを行うための機能を提供することが望ましい。また、検索エンジンのように、ネットワーク上の離れた位置にあるデータベースサーバーを使って検索を行う場合、このような絞り込みが、不必要な通信を起こさずにクライアント側

で行える事も重要である。

そこで本研究では、検索エンジンによって得られた結果を、組織化した形でユーザに表示するシステムを開発する。さらに、従来の検索エンジンでは検索結果は単純な HTML のページとして与えられていたが、われわれのシステムでは、検索結果はユーザがさらに絞り込みを行うための機能を持ったアプレットとして与えられる。これにより、ユーザがクライアント側で対話的に検索結果を再組織化したり、編集したりしながら絞り込みを行う事ができる。

このような検索エンジンの結果の絞り込みを支援するシステムとして、西村らによる Mondou [5] がある。Mondou ではキーワード検索を行うと、検索結果とともに、そのキーワードと関連が深いと思われるキーワードもあわせて検索し呈示してくれる。ユーザはこれらの関連の深いキーワードを使って効率良く絞り込みが行える。それに対し、われわれのアプローチでは URL の候補をそれらの間の関連性を使って組織化して表示することでユーザの絞り込みを支援することを狙っており、Mondou のアプローチとは、互いに補完しあえるものと考えられる。

以降、第二節でメールの組織化機構、ついで第三節で検索エンジンの検索結果の組織化機構について述べ、第四節でまとめとする。

2 メールを扱う組織化機構

この章では、本研究の第1段階として、電子メールをシステムが一括して集め、それを組織化して、ユーザに呈示する機構について述べる。

2.1 システムの構成

ここでは電子メール管理に対するユーザ支援システムの構成要素を4つ挙げ、それぞれの機能に関して考察する。

●フィルタリング

1. メールの子プロジェクトや全文から、キーワードによる文字列マッチングを行い、フィルタリングする方法。
2. あるメールに対しての”賛成”・”反対”といったキーワードによってフィルタリングを行う方法。
3. メールの中身を解析し、その情報に基づいてフィルタリングを行う方法。

●組織化

1. あるメールと、そのメールに対する返答群を対にして組織化する方法。これはメールの Subject の行に注目する。Subject の行に”Re” という文字があれば、そのメールはあるメールに対するリプライであるので、その元のメールとリプライのメールを1つのグループとして組織化する。リプライのメールがいくつもあれば、1つのグループは元のメールとそれら複数のメールから構成される。
2. 受けとったメール全体の議論の流れ(例えば誰がどのようなメールに対してどのような返答をしたのか)を判断して組織化する方法。そのメールがどのような種類のメールなのかをシステム側が判断するために、Subject の行に次のようにユーザが予め書いておくことにする。
Subject: Re:A'opinion #supports#
Subject: Re:A'opinion #objects#
このように書くことにより、賛成は賛成意見のグループに、反対は反対意見のグループにまとめられ、その議論の流れを知ることができる。
3. 過去のユーザの履歴から判断してメールを組織化する方法。システム側が、ユーザプロファイルをとることにより、そのユーザが過去どのような組織化を行ったのかを学習し、自動的にユーザの嗜好に合うように組織化を行う。

●呈示

1. メイリングリストなどで、いまどのような議論がなされ、どのような意見が出ているのかなどを gIBIS のようなグラフィカルな形で呈示する。
2. 組織化されたメールをハイパーテキスト化し WWW のページとしてユーザに呈示する。その際、組織化されたグループで1つのページを構成する。

●ユーザ情報管理

ユーザが、どのようなメールに対してどのような対応を取ったかの履歴を管理する。同じようなメールが到達した場合には、その履歴に沿ってエージェントが対応を決定し、それをユーザに伝える。

2.2 システムの実装

本システムでは上に述べた構成方法からフィルタリングと組織化は1と2、呈示は2の方法をとり、次のような機能を持つシステムを構築した。このシステムでは図1に示すように、まず(1)ユーザがシステムにキーワードを送り、(2)フィルタリング部がメールをフィルタリングする。(3)フィルタリングされたメールが組織化部に送られ組織化される。(4)最後に呈示部に送られ、メールがハイパーテキスト化され、WWWブラウザにより呈示される。また図2にシステムの初期動作画面を示す。な

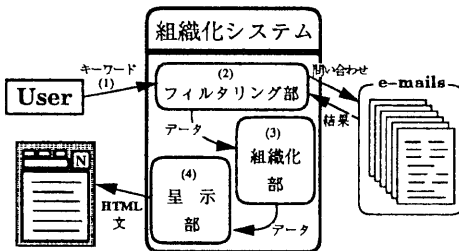


図1: 電子メール組織化機構

お本システムはJava言語[4]により実装されている。Java言語で実装されたアプリケーションはプラットフォームホームから独立しており、Javaインタプリタさえあればそのマシン上でも動作することができる。

● フィルタリング

図2のテキストエリアにキーワードを入力することによって、システムが全てのメールの全文に対してフィルタリングをかける。そしてキーワードにマッチングしたメールの番号はシステムにより保持される。また、1度打ち込まれたキーワードもシステムにより保持される。したがって、ユーザは何度も同じキーワードを打つ必要はない。

● 組織化

2.1節で述べた組織化のうち1と2について実装を行っている。まず、質問メールと、それに対する返信メールを1つのグループとして組織化する。そしてそのグループのなかでさらに「賛成」「反対」といったキーワードを基に組織化を行う。また質問メールに対して返信メールがない場合には、「返信がない」というメールのグループに入れられる。そのメー

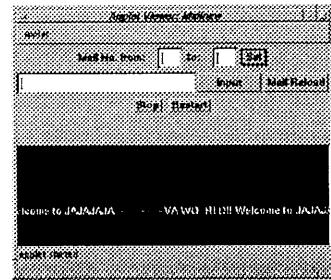


図2: 初期動作画面

ルに対する返答が来た場合には、システムによりメールの再組織化が行われ、先程”返答がない”メールのグループに入っていたメールは新しいグループを作る事になる。この再組織化については以下で述べる。

● 再フィルタリング、再組織化

このシステムは初期動作時に、指定されたディレクトリにあるメールは全て読み込むが、それだけではシステム作動後に受理されたメールはフィルタリングや組織化が行なわれない。したがって本システムでは、1分毎(デフォルトとして指定、変更可能)にディレクトリにメールの数の変化がないかどうか確認している。システムは初期作動時にその時のメールの数を認識するのでもし作動後にメールの数に変化があれば、その変化を認識し、再フィルタリング、再組織化を行う。この時それ以前に打ち込まれたキーワードに基づいて動作を行う。このような機能を備えているので、このシステムをデーモンとして使用することも可能である。

● 呈示

本システムでは組織化されたメールをハイパーテキスト化しWWWのページとして各々表示させている。実際にユーザに呈示されたページを図3に示している。まず第1段階としてユーザにより打ち込まれたキーワードを表示するページを作成する。そして第2段階のページには組織化されたグループの名前が表示されている。その名前をクリックすることにより、それらのメールが書かれたページへと飛ぶ事が出来る。

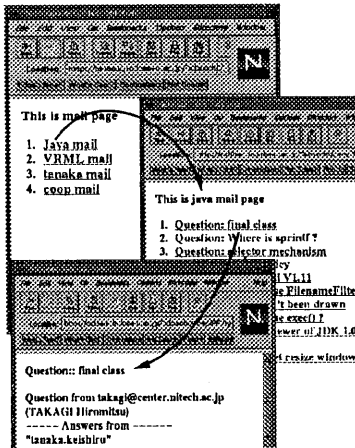


図 3: HTML 化されたメールを表示するページ

3 インターネット上の情報に関する組織化機構

次に、扱う情報をインターネット上の情報に拡張して、それらを組織化する機構を提案する。前節に述べた組織化の機構は、電子メールを集めそれらを組織化し、HTML 化することによりユーザにとって見易い形で呈示するというものであり、あくまでもスタンドアロンなアプリケーションであった。

それに対し、インターネット上の情報に拡張する場合は、サーバー、クライアント間で通信を行い、クライアントからの組織化の要求に答えなければならない。従来の CGI を使用するような手法では、クライアント、サーバー間で何度も通信を行うことになるのでネットワークに対して負荷をかける。またサーバー側で全ての要求に答えなければならないので、インターネットのように同時に不特定多数の人々がアクセスするような環境ではサーバーに対しても負荷をかけることになる。

そこでこの節では、例としてインターネット上の検索エンジンである Alta Vista[2] の検索結果を組織化の対象として、組織化機構を Java 言語によるアプレットで実現することを目指す。Java 言語で作成されたアプレットならば、クライアント上でユーザが対話的に情報を組織化することが可能である。またサーバー側にあるデータベースとの通信も 1 回セッションを張ることによりそれが保たれるため、アクセスの度にデータベースのオープン・クローズといった動作は不要となる。

3.1 問題点

Alta Vista の検索は、ユーザがキーワードを打ち込むだけでよい。そして打ち込んだ結果、図 4 に示すように 1 ページあたり 10 個の候補を表示しその表示内容は、ページのタイトル、ページの最初の数行、URL である。

現在ではユーザはタイトルの下に書かれた少量のコメントを基に、自分に必要となる情報がどれかを推定してリンクをたどっていく。もし最初に表示されたページに欲しいリンク情報がなければ、次のリンク情報のページへと移るのである。

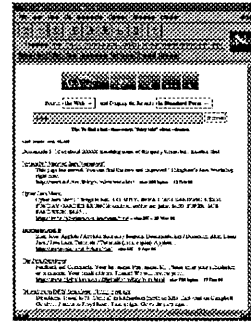


図 4: Alta Vista

しかし、このように大量に呈示されてもユーザが見ようとするのは多くても 2~3 ページである。もしもそれ以降に本当に必要とするページが挙げられていたとしても、見つけることはほとんど不可能である。またあるトップページからたどれるようなページがヒットしたとしても、それらはまとまりのある単位として示されるのではなく、ばらばらに示されるの大変非効率である。

3.2 組織化

検索された結果を、一旦データベースに格納し、それをよりユーザに見やすい形で呈示できるように組織化することにする。その組織化の方法としては次のようなものが挙げられる。

1. 検索結果から得られる URL の情報を用いることによる組織化。
 - ある共通のディレクトリから下のページは同じグループに属するものとしてまとめあげる。例えば URL が、“http://xxx.ac.jp/~yyy/zzz.html” と “http://xxx.ac.jp/~yyy/www.html” の場合は、同

じサイトでかつトップディレクトリが同じであるので、この2つは同じグループであると見なす。また、グループの1番先頭にあたるものをグループの代表とし、表示する際にはその代表を表示し、他のものは隠蔽する。

- 同じサイト上にあるものは、関連性のあるものとして近い位置に表示する。例えば、“http://xxx.ac.jp/~aaa/”と“http://xxx.ac.jp/~bbb/”は関連性があるとみなす。

2. 検索結果のうち、削除してもよいものと興味のあるものをいくつかユーザにマークしてもらうことによって、そのユーザのプロファイルを取り、それを基に、マークした以外のものについてシステムが自動的に削除するもの、見るべきものを判断し、再組織化を行う。

これらの組織化システムはJava言語を用いて実装を行うため、1度データがクライアントに送られると、再組織化やユーザプロファイルの獲得を全てクライアント側で行う事ができる。また、クライアント側で次のようにビューを変化させることも可能となる。

- 初め、検索結果は上に記したように同じグループに属するものはすべて表示されるのではなく、代表1つだけが表示され、他のものは隠蔽される。もしユーザが代表以下のものも見たいければ、その要求に応じて表示を切り替え代表以外のものも見ることができる。
- ページのタイトルや更新日時により、再フィルタリング、並べ替えをすることができる。

3.3 システム構成

このシステムは以下に挙げるものを構成要素とする。システム構成図を図5に示す。

1. クライアント側のWWWブラウザ上で動作するアプレット
2. クライアント、サーバ間の通信を行うツール
3. サーバ側で動作するアプリケーション
4. サーバ側のデータベース

4 あとがき

本研究では、まず電子メールを扱うスタンドアローンな情報組織化機構についてその設計法を検討し、

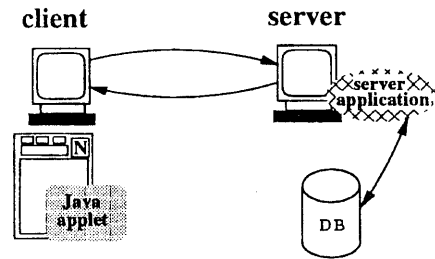


図5: システム構成

実際にその開発を行った。次に扱う情報をインターネット上の情報に拡張してそれらを組織化する機構を提案した。今後の課題としては以下の事項が考えられる。

- 実際に Alta Vista 等の検索エンジンと、統合できるようなシステムの実装
- ユーザの嗜好や履歴を反映できるようなアプレットの実現
- よりインタラクティブなユーザインタフェースの開発
- 組織化され示された情報に対する問い合わせ (query) 機能の実現

謝辞

本研究は、一部、文部省科学研究費重点領域研究 (課題番号 08244103) による。

参考文献

- [1] Conklin, J. and Begeman, L. gIBIS: A Hypertext Tool for Exploratory Policy Discussion. *ACM TOIS*, Vol. 6, No. 4, pp. 303-331, October 1988.
- [2] Digital Equipment Corporation. AltaVista: Main Page. <http://www.altavista.digital.com/>.
- [3] Kamiyama, K. and Martin R. Grassroots: A System Providing a Uniform Framework for Communicating, Structuring, Sharing Information, and Organizing People. *In Proc. of Fifth International World Wide Web Conference*, May 1996. http://www5conf.inria.fr/frch_html/papersessions.html.
- [4] Inc. Sun Microsystems. Java language environment white paper. <http://java.sun.com/doc/>.
- [5] 西村英樹, 伊藤耕一郎, 河野浩之, 長谷川利治. 重み付き相関ルール導出アルゴリズムによる www データ資源の発見, 電子情報通信学会, 第7回データ工学ワークショップ論文集, pp. 79-84, March 1996.
- [6] 今津英世. Web サーバーの技法. *Internetworking*, Vol. 2, No. 7, pp. 31-48, July 1996.