WWWによるコミュニケーションを支援する情報集約システム

Supporting web based communication through an information aggregate software.

新井 俊一 [1)]
Shunichi ARAI

1) 新井技術研究所（E-mail: arai@mellowtone.org ）

ABSTRACT. In this project, we developed an information aggregate software. That software extracts new information from any web pages and creates a digest. We are running a web site that provides summarizing service publicly. That web site enables more efficient use of the World Wide Web. Users can read the information from multiple sources at a time and that enables quick response to the information.

1 背景

インターネットの普及が進むにつれ、多量の情報がウェブを通じて提供されるようになった。それらの情報は、検索サイトやディレクトリサイトによって分類され、目的の情報を効率的に探すことができるようになっている。しかしウェブの普及による小規模サイトの増加に伴い、最新情報を定期的に入手するための労力は増大している。新しい情報を得るためには、多数のサイトをブラウザによって閲覧しなければならないからである。

とりわけ個人が運営しているようなサイトや、そこに設置された掲示板などは、数が多く、更新頻度が低いため、手動で閲覧することによって最新情報を入手することは困難を伴う作業となる。

この問題を解決するためのツール類は古くから開発されており、こうしたツールには、更新チェック用クライアントソフトウェア (代表的なものとして WWWC) と、更新状況の一覧を作成するサーバ用ソフトウェア (アンテナと呼ばれている。代表的なものに、なつみかん) などがある。

しかし更新チェック用クライアントソフトウェアにおいては、クライアントにソフトウェアをインストールする労力や、ブックマーク (チェックするサイトのリスト) を作成する労力などが必要となる。そのため、カジュアルユーザ層への普及は得られていない。

アンテナ (更新状況の一覧を作成するサーバ用ソフトウェア) においては、システムを動作させるためにコンピュータの知識や資源を要するため、更新状況の一覧ページを作成することができるユーザは限られている。但し、作成された更新状況の一覧は、一般に公開されるため、多くのユーザがアクセスすることができる。

いずれのソリューションも、更新されたことを検知する機能を備えているが、更新内容を提示する機能は備えていない。そのため、更新が確認された場合、該当するページをブラウザで閲覧し、新しい情報がどのようなものであるかを手動で確認しなくてはならない。

更新日時の取得をするためには、ウェブサーバが提供する Last-Modified 情報や、ページのサイズ情報が利用されることが多い。この場合、Last-Modified 情報を返さない広告付きページなどでは、正しく更新日時が得られない場合もある。これは、系時的に変化する広告情報によってページサイズが変化するためである。

2 目的

前章で述べた現状の問題点について解決するために、我々は、任意のウェブページ群から最新の情報を取得して、一つのページに集約して表示する技術を開発する。そして、システムをウェブ上のサービスとして、誰でも利用できる形で運用を行う。

今までのソリューションと異なり、任意のウェブページから自動的に最新情報だけを取り出すことができ、複雑なエージェントの開発などを必要としない。すなわち、個人の日記ページや、掲示板ページなどの最新情報を一括して閲覧することが可能となる。

またウェブ上のサービスとして公開することで、インストールの手間を省き、だれでもが簡単に利用することができるようになっている。そのため職場や学校、インターネットカフェなどのコンピュータからも利用することが可能であり、ユビキタスコンピューティングの観点からも優れている。

これによって、WWW をより効率的に利用することが可能となる。多くの情報源からの情報を、一元的に閲覧できるようになり、それらの情報への即時的対応によるコミュニケーションの円滑化が期待される。

本プロジェクトでは、ウェブページから最新情報を抽出する要素技術を開発した。また、最新情報の集約と共有を行うシステムを開発し、ウェブ上のサービスとして試験運用を行っている。

本稿では、まず 3 章で、ウェブページから最新情報を抽出する要素技術について解説を行う。その後、4 章で、運用を開始したサービスについて概要を述べる。

3 ウェブページからの最新情報抽出

ウェブページから情報を抽出する手法としては、各ページに個別の特化したエージェントを利用する方法 (一例として Airclub) がある。

こうした従来法では、特定のウェブページを取得するエージェントをプログラミングし、ページから情報を抽出する。利点としては、どのようなページであってもプログラムさえ記述すれば、正確な情報取得が可能なことが挙げ

られる。
欠点として、新しいページに対応するたびにプログラムを記述せねばならず、多くのウェブページに対応することが困難であることが挙げられる。また、各種のエージェントが存在することにより、ユーザ操作が複雑になり使いにくくなるおそれがある。

本稿で、我々が提案する手法は、一般的なウェブページから汎用的に最新の情報を取得するものである。ここで最新情報とは、ウェブページに新規に掲載された情報を指す。これにより、ユーザは、エージェントがページに対応しているかどうかなどを考慮することなく、容易に利用することが可能となる。
最も大きな利点は、任意の個人サイトやその掲示板などの最新情報を自動的に取得できることである。こうした用途は、ページごとにエージェントを用意する方式では対応することが難しい。

(1) 任意のページからの最新情報抽出手法
本手法では、任意のページから最新情報を抽出するために、ウェブページ上のテキストを系時的に比較することを特徴とする。ある一時点の HTML データの構造情報等からでは、意味情報を取り出すことは困難であるが、HTML データを系時的に追跡し、変更箇所を取得することで、最新情報を抽出することができる。以下に最新情報抽出の手順を述べる。

(1) まず、最初にターゲットとするページの情報を取得する。
(2) 次に、HTML データを、文章を分ける意味を持つタグ (テーブルや段落や改行など) によって細かい文字列のリストに分解する。
(3) 細かい文字列毎にハッシュを作成し、ハッシュリスト A を作成する。
(4) それから一定時刻経過するのを待つ。
(5) ターゲットとなるページの情報を再取得し、(2) と (3) の手順を行い、A' を作成する。
(6) ハッシュリスト A と、A' の比較を行う。
(7) 比較の結果、A' にあり、A にない文字列を最新情報として抽出する。
(8) 抽出した文字列のハッシュを A に追加して、(4) へ戻り、繰り返す。

以前の文章データをハッシュとして記憶し、比較することで、消費する記憶領域の低減と、計算速度の向上を実現している。
ターゲットとなるウェブサーバへの負荷や、必要性を考慮し、情報取得の頻度を決めることが望ましい。情報の取得頻度は、過去の更新頻度の平均から算出した値を用いても構わないし、また手動で選択できるようにしても構わない。

4 情報集約と共有のサービス

本プロジェクトは、情報集約と共有のサービスを WWW 経由で提供することを目標としている。このサービスとは、複数のウェブページから最新の情報を取得し、それをまとめて一つのウェブページとして提示するものである。

従来の同種のソフトウェアは、クライアントアプリケーションとして提供されることが多かった。これらのソフトウェアは通信が高コストかつ低速であったころに、効率的に通信を行うためのツールとして利用されていたものが多い。
それにたいして、本プロジェクトでは、情報集約サービスを WWW 経由で提供することで、気軽に利用できる、コミュニケーションを円滑にするためのツールを目標としている。クライアントソフトウェアとウェブサービスの比較を下に挙げる。

クライアントソフトウェア:
・ 低速通信回線でも遅延無く動作する。
・ 高度な GUI による操作が可能である。
・ 株価やニュースなどのリアルタイム情報に適している。

ウェブサービス:
・ ソフトウェアをインストールする手間が無く、気軽に利用可能である。
・更新情報の一覧ページを公開、共有することができる。

(1) 集約された情報の共有
更新情報の一覧ページを公開できるようにすることには大きな意義があると考える。

現在、多くの個人がウェブサイトを作成し、運営している。しかし、それらのなかで長期間に渡って更新が続けられているものは少ない。その理由の一つには、公開するコンテンツのストックが尽きることが考えられるが、もう一つ、大きな要因として反響が少ないことが考えられる。
これは、更新頻度が少ない個人サイトは閲覧者が少なくなり、それにより反響が少なくなる、また反響が少ないことにより更新意欲が減ぜられる、という悪循環によるものと推測される。この悪循環を防ぐために有用なのがコミュニティを形成することであると考える。

ここでコミュニティとは、同じような興味や目的を持つ複数のページ群が、互いにリンクしあっている様子を表す。またページの作者間が、メールや掲示板でコミュニケートしたり、その対話の内容をページに反映させたりするような性質を持つ。
コミュニティが有用に機能するためには、その中心となるサイトが、コミュニティのニュースなどを定期的に掲載したり、メンバーが集まる掲示板を運営したりする必要がある。しかし、コミュニティの参加者全員が集まるような求心力の高いサイトを運営していくことは容易ではない。定期的な更新や、掲示板の管理には、多くの労力が必要となる。

我々が提案する情報集約サービスは、こうしたコミュニティの繋がりをサポートすることを目指している。
情報集約サービスには、コミュニティ関連ページのトップページや、掲示板の URL を登録する。それにより、コミュニティの最新情報が集約され、一目で参照できるページが出来上がることになる。
更新頻度の低い個人ページであっても、複数が集まれば、更新頻度の高いページを構成することができる。それによって閲覧者が増加し、反響も大きくなり、更新の意欲が増大することが期待される。

(2) 情報集約サービス MyPortal
我々は、情報集約サービスを MyPortal と命名し、試験サービスの運用を開始した。試験サービスは、誰でも自由に簡単に利用することができる。

情報集約サービスの出力ページを、ポータルと呼ぶ。ポータルを作成したいユーザは、まず会員登録をし、任意の URLs をターゲットとしてポータルを作成する。作成されたポータルには、それぞれ URL が割り当てられ、その

Table 3.1

CPU	Intel Celeron 400MHz
Main Memory:	312MB
HDD	6.43GB
OS	Linux 2.2.19
Web Server	Apache 1.3.22
Web Application Script	`Perl 5.005_03`
Agent	JDK 1.3.1
Database	Postgres 7.2

URL にアクセスするだけで、誰でもポータルを閲覧することができる。

ポータル URL の例
http://www.moodindigo.org/bin/usr/portal/1/view.html

システムは、フロントエンドに Apache と Perl を、情報収集エージェントに Java を採用している。データベースには Postgres 7.2 を使用し、両者を連携させている。将来的にサービスが拡大した場合、エージェントを複数台のサーバに分散化する必要がある。それに備えて、フロントエンドとエージェントの分離性を高める設計を行っている。Table 3.1 にシステムの諸元を示す。

クロスサイトスクリプティングなどを防ぐため、セキュリティ機能を備えたウェブアプリケーションフレームワークを構築し、その上でフロントエンドを動作させている。

現在、システムを動作させるために、比較的に安価なレンタルサーバを利用しており、ユーザが増加したさいには、サーバ性能がボトルネックになる恐れが高い。
データベースのチューニングや、持続接続などのパフォーマンス対策を行っているが、数千人規模の登録ユーザに対応するためには高性能なサーバ設備が必要となる。現時点では、資金計画が存在しないため、数百人規模にまで漸進的に登録ユーザを増加させ、データを得る計画である。

5 おわりに

本稿では、集約した情報を共有できるサービスの概要につき述べた。本サービスは、ウェブページから自動的に最新情報を抽出する技術を用いて、誰でも容易に利用できるサービスの提供を可能とした。
それにより、コミュニティのつながりを支援し、個人によるウェブサイトなどの情報量の向上が期待されることを示した。
今後の課題としては、資金計画の策定と、パフォーマンス対策による、利用可能ユーザ数の増大が挙げられる。本サービスを、多くのユーザに利用して貰いたいと考えている。

6 参加企業及び機関

なし。

7 参考文献

なし。