データベースシステム　125－13
（２００１．７．１７）

自律的情報収集による超分散Web検索システムPIRCSの設計と試作

小林亜樹†　王宏剛†　樋山大輔†　山岡克式‡　酒井善則†

†東京工業大学 大学院理工学研究科
〒152-8552 目黒区大岡山2-12-1
E-mail: koba@ss.titech.ac.jp　　Tel. 03-5734-2193
‡文部科学省メディア教育開発センター
〒261-0014 千葉市美浜区若葉2-12

インターネット上の情報の多くがWWW上にあり、検索サービスも存在する。しかし、本質的に超分散型アーキテクチャであるWWWの情報を集中型データベースで検索しようとするには限界がある。筆者らは、リンクで結ばれたコンテンツ同士に意味的な共起性が見られる点を利用して、利用者の指定したコミュニティに探索空間を限定した検索機能を提供する枠組みであるPIRCSを提案している。PIRCSでは、利用者の多様な情報要求やコンテンツの種類を扱うために、プラグイン型の検索フィルタを導入し、サーバ、クライアントを現在のWebシステムに対して拡張する。これらの拡張はWebシステムと互換性があり、通信部分は既存のHTTPの拡張となる。システムの構成と試作システムについても説明する。

インターネット, WWW, 情報探索, リンク, PIRCS, Webコミュニティ

Design and Trial System of PIRCS: Super-distributed Web Search system using Autonomous Information Collection

KOBAYASH Aki,†　WANG Honggang,†　HIYAMA Daisuke,†
YAMAOKA Katsunori,‡and SAKAI Yoshinori†

†Tokyo Institute of Technology
2-12-1 O-okayama Meguro-ku Tokyo, Japan, 152-8552
E-mail: koba@ss.titech.ac.jp　　Tel. +81-3-5734-2193
‡National Institute of Multimedia Education
2-12 Wakaba Mihama-ku, Chiba-city, Japan, 261-0014

There are many searching service of WWW and most information is on the internet. However, WWW has super-distributed architecture so that centralized database solution cannot fit to retrieve information from WWW any more. We have proposed PIRCS which provide information retrieval function within limited web community. PIRCS uses collocation between linked contents. PIRCS has induced plug-in retrieval filter to answer various information need and manage every kind of contents. We extend web system and HTTP for our PIRCS system. We describe system framework and trial system.

Internet, WWW, Information search, Link, PIRCS, Web community

1 はじめに

インターネット上で公開されている情報は、その多くが WWW のシステムを通じて入手することができるようになっており、インターネットと言えば、WWW サービスのことを指すこともあるほどである。これらの情報を巨大なデータベースとして利用するために、利用者の多様な要求に応えうる検索サービスが求められている。実際に、Yahoo![1]、Excite[2]、goo[3]、Google[4] など、数多くの検索サイトが実用化されており、Web アクセスの相当数がこれら検索サービスへのアクセスとなっているとも言われており、需要の高さがうかがわれる。

しかし、WWW は本質的に超分散型の情報配信アーキテクチャを持っており、コンテンツの位置や内容、相互の関係などを記述・制御する仕組みを本質的に持っていない。このため、現在の検索サービスでは、検索というサービスを実現するために Web システムとのギャップを埋める必要があり、大きなコストを強いられている。また、インターネットドメイン全体の情報からの検索を目指すサービスでは、必然的に設備等が大規模となるため、利用者個々の多様な情報要求 [5] に応えるという小回りの利いたサービスの提供は困難である。しかも、全情報の収集には成功していない [6]。くわえて、Web コンテンツの実体と検索情報の遊離から、検索情報と実際の情報との間の不整合が問題として挙げられるなど、Web アーキテクチャが検索を考慮していないことによる問題点が多数指摘できる。

そこで本稿では、WWW に柔軟な検索モデルへの適応性を与えるためのアーキテクチャとしての PIRCS(Personal Information Retrieval with Cooperative Servers)[7]–[11] について説明し、その機能性について検討を行う。PIRCS は、Web サービスに対する多様な情報要求を解決するために、原理的にクライアント側で情報検索を実行する。しかしながら、すべての情報を利用者の手元で処理するためには、莫大な HTTP トラフィックが発生することとなり、効率的ではない。そのため、サーバ側で一部の検索機能を実現する。また、スケーラブルな超分散モデルの利点を残したまま、効率的な検索サービスを実現するための分散協調モデルを Web サービスに導入することについて検討する。

その後、PIRCS を既存の WWW との整合性を保ちながら機能を提供するための設計について述べ、試作システムについて説明する。

2 Web 検索モデル

2.1 データベースとしての WWW

WWW は、原理的にはそのコンテンツ（リソース）を識別する URL(URI) と、コンテンツの転送プロトコルである HTTP の組み合わせからなるシステムである。その際、URL とコンテンツの対応は URL からコンテンツへの 1 方向であり、その関連付けについて何らの保証もしていない。このアーキテクチャは超分散的であり、

- URL の位置 (FQDN) 以外の部分は相当に自由に決められる
- URL によって示されるコンテンツの生成・更新・消去が自由である
- コンテンツの数に制限がない

という利点がある反面、特定のリソースを取得することができるだけであり、利用者が特定の意味を持つ情報を見つけ出す検索の仕組みは本質的に盛り込まれていない。古典的なデータベースモデルに則って言えば、WWW は URL（を ID として）、コンテンツの巨大なテーブルであると捉えられる。しかし、このテーブルは URL からコンテンツへの一方通行の取り出し、すなわち、SQL では where 句に URL しかとれない不完全なテーブルである。

このため、全世界に広がる WWW を巨大なデータベースとして活用しようとするときには、検索機能を補完するシステムが必要となる。しかし、where 句にコンテンツをとるようなことをすると、全世界のコンテンツを収集しなければならず、このような古典的な手法が原理的に破綻していることがわかる。

2.2 Web 上の検索モデル

2.2.1 分類型モデル

Web 上の検索サービスとして、最も古くからみられるのは、Yahoo!に代表されるディレクトリサービスである。このサービスは、巨大なリンク集をその意味づけによって階層的に分類することで、利用者の情報要求に応えようとするものである。

人手によって登録を行うため、比較的精度よくサイトの分類が行われるが、登録数が増えるにしたがい、検索時の利便性を増すために分類項目を細分化する必要があり、ジャンル横断的なサイトの存在や利用者の情報要求との項目のミスマッチが問題となる。また、一般に「サイト」と呼ばれる、特定の管理主体が管理する一連のコンテンツ全体での分類を行うため、個々の

コンテンツ（Web ページ）を直接検索することはできない。

2.2.2 全文検索モデル

個々の Web ページに含まれる文章から、直接情報検索を行いたいという要求からうまれたのが、goo、Google などに代表される全文検索モデルである。このモデルによる検索サービスでは、まず、ロボットと呼ばれる自動コンテンツ収集プログラムによって、各ページの情報を収集してインデックスを抽出しデータベースを構築しておく [12]。利用者の問い合わせ時には、このデータベースから検索を行い、結果ページへのポインタ（リンク）を返す。

このモデルでは、広範囲のページデータを事前に収集することで、様々な問い合わせに応えようとする。そのため、特定分野における問い合わせキーワードによる検索を行いたい場合にも、あらゆる種類のページ群から単純に検索が行われてしまい、情報要求に十分応えられない。これは、ページの収集部分を除くと、基本的には従来のテキスト検索技術の延長線上にあり、HTML などのハイパーリンクを検索に有効に利用しているとは言えないことに起因する。また、構築したデータベース中のインデックス情報は、コンテンツ本体から切り離されており、更新の同期は不可能であり、超分散構造である WWW アーキテクチャでの利用には本質的な無理がある。

Google で用いられている PageRank[12] は、HTML 文書など、ハイパーリンクを含む文書（以下、含リンクコンテンツ）について、そのコンテンツへの他の含リンクコンテンツからの参照数などを評価に加味することで WWW の特徴を生かそうとする試みであった。しかし、あくまで単独のコンテンツの適合度評価のパラメータに過ぎず、リンク構造を活かした検索技術としては不十分である。

2.2.3 コミュニティ発見モデル

リンクの参照・被参照関係をより積極的に情報探索に利用しているのが、近年の Web コミュニティ発見型のモデルである。HITS[13] から始まった Cleverproject[14] が代表的で、リンクを多く含む Hub ページと被リンク数の多い Authority ページとに分類し、リンクの重み付き和として表される Authority スコアの高いページをコミュニティを代表するページであるとする手法である。

この方式は、PageRank の考えをさらに推し進めたものと言え、多数の Web ページクリエータが参照する価値があるとした、被リンク数の多いページを発見することができる [16][17]。しかしながら、分野名を問い合わせとするような単純な検索には有効であるものの、複雑な問い合わせにマッチするコンテンツを直接に検索することはできない。

2.2.4 情報の事前収集

分類型モデルと全文検索モデルでは、超分散である WWW のデータを従来のデータベース技術に適合させるために（暗黙、または、明示的に）、事前に情報の収集を行う。このような検索アーキテクチャの下では、原理的に、コンテンツの実体と検索のためのインデックス情報とが分離され、正しい検索結果を保証できない。実際、検索サイトでの結果リストの中には、内容が更新されてしまっていたり、コンテンツ自体が消滅していたりする場合が多々ある。各検索サービスでは、このようなインデックスとコンテンツの不整合をなるべく少なくしようと頻繁に（1 週間〜2 週間に 1 回程度以上）、インデックスのためのアクセスを行っていることを観測しているが、本質的に解決できる手法ではない。また、この情報収集のためのアクセスは、本来ならば不要なものであり、インターネットの帯域を不必要に使用するとみることもできるなど、WWW がその多数のコンテンツをデータベースとして利用する枠組みを持っていないために、無理の生じる設計となっている。

コミュニティ発見モデルの場合、単純な問い合わせにしか適合しないため、事前にページ情報を収集する方式であっても、コミュニティ発見の精度に大きな悪影響は生じないものと考えられる。しかし、リンク情報を利用してコミュニティを発見するためには、大量のコンテンツを収集しなければならず、効率や応答速度の面で問題がある。

2.3 進化的探索モデル

情報探索にあっては、これまでに述べてきたようなモデルばかりではなく、進化的探索に分類されるモデルも存在する。進化的探索（evolving search）とは、利用者の情報要求が検索の過程で新しい情報に出合うことによって動的に変化するようなモデルである [18]。Web システムにおいては、HTML のリンクを辿ることによって、目的の情報を見つけ出す行為は、この進化的探索であるとみることができる。

このような検索を効率的に行うためには、興味の対象となるコミュニティを特定し、その中から問い合わせに適合するコンテンツを発見した後、変化する情報要求に対して適切なリンクを選択できるよう、誘導する仕組みが必要である。このうち、

- コミュニティ指定検索

- ナビゲートアシスト

は従来の手法では提供されない。

現在のところ、一般的な Web 情報検索の方法は、検索サービスに対して、問い合わせ（キーワード）の追加による適合性フィードバックを繰り返し、ある程度コミュニティを絞り込んだ後、結果リスト中の任意のページへジャンプしたあと、さらに HTML のリンクを辿ることで進化的探索を行っていると考えられる。いずれも、適切な情報探索のための代用手段に過ぎず、Web 情報検索のための仕組みを導入する必要がある。

3 PIRCS

3.1 目的

多様化する情報要求に応えるため、WWW の仕組みを拡張して、利用者の目的に柔軟に適合する検索のためのフレームワークを提供することが PIRCS の目的である。

特に、従来の Web 検索モデルが、HTML などの含リンクコンテンツの特性を十分に活かしているとは言えない点に注目し、リンク構造を意識した検索を効率的に行えるよう、Web サーバに検索機能を拡張する。その際、検索測度はコンテンツの種類や利用者の情報要求によって様々であることを踏まえ、サーバ・クライアント双方の協調的動作によって検索を実行するモデルを提案する。

3.2 アーキテクチャ

PIRCS では、コンテンツの検索機能を Web サーバとユーザーエージェントとで分担する。検索システムは一般に、

1. 利用者の（概念的な）問い合わせ
2. システムの問い合わせ文に変換
3. （狭義の）DB 中を検索
4. 結果リストを得る
5. 利用者のために表示

という手順を踏むが、PIRCS では、このうち 3 の検索処理のみをサーバ側で行い、それ以外の機能はクライアント側で実現する。このようなアーキテクチャを採る理由は、

- 柔軟なシステム構成を可能とする
- コンテンツの更新に適切に対応する
- 通信路に不必要な負荷をかけない

ためである。

したがって、サーバ側には従来同様の Web コンテンツ提供機能のほかに、コンテンツの更新にしたがい、適宜構築されるデータベースと、そのデータベースに対する検索機能を拡張する。このような拡張された Web サーバを AWS(Advanced Web Server) と呼ぶ。AWS は、現在の Web サーバのように、要求された URL に対応するコンテンツがあるかどうかをその時点で判断することはない。自サーバ内コンテンツに対しては、検索に対して正しい結果を返す必要性から、コンテンツの更新時に登録処理を行い、検索データベースとの一貫性を維持する。このとき、リンク先コンテンツに対してもコンテンツの取得とインデクシングを行う。

マルチメディアコンテンツの検索処理を実現するためのエンジン部分は、プラグイン形式として、様々なメディアや目的に対応できるようにする。この検索処理を行うルーチン群を検索フィルタと呼ぶ。検索フィルタは、あらかじめ検索インデックスを構築する機能と、検索時にキーとコンテンツとの類似度（距離）を計算する機能とを持つ。

コンテンツの検索は、URL の指定と、利用する検索フィルタ、および、そのフィルタに適合する検索キーの組み合わせを問い合わせとする。この問い合わせは、HTTP/1.1 のヘッダ部分を拡張したフィールドとして送信する。一般に Web サーバは、自身が解釈できないヘッダフィールドは無視するため、AWS 以外の従来の Web サーバに対して、このようなことをおこなっても問題ない。

$$< \text{問い合わせ} > := < \text{start URL} >, \{< \text{フィルタ名} >, < \text{キー} >\} \quad (1)$$

AWS は、この問い合わせを処理し、その URL に対するコンテンツと、そのコンテンツに含まれるリンク先コンテンツの検索フィルタによる類似度をあわせて返す。また、リンク先コンテンツが含リンクコンテンツであるなら、そのリンク先 URL もリストとして返す。これらの情報も HTTP のヘッダとして送信する。

$$< \text{検索結果} > := < \text{フィルタ名 list} >, < \text{結果 list} >, < \text{URL list} > \quad (2)$$

$$< \text{結果} > := < \text{URL} >, < \text{距離 list} > \quad (3)$$

利用者の情報要求に基づく、広域的な検索を担うのはクライアント側である。このような情報探索機能を

持った Web ユーザーエージェントのことを、WBS(Web Browser with Search function) と呼ぶ。WBS は、利用者の要求を PIRCS の規約に従った AWS への問い合わせ要求に変換し、HTTP に載せて問い合わせを行う。その後、AWS からの応答を適当に加工して利用者の提示することで情報探索機能を実現する。

3.3 通信

具体的な HTTP ヘッダの拡張は、RFC2068[19] による HTTP/1.1 の定義に以下の変更を加える形で行う。

＜変更＞

```
Request = Request-Line
          *(general-header
          | request-header-2
          | entity-header )
          CRLF
          [ message-body ]

Response = Response-Line
           *(general-header
           |response-header-2
           |entity-header )
           CRLF
           [ message-body ]
```

＜追加＞

```
request-header-2 = request-header
                 | X-PIRCS-SearchParams
reaponse-header-2 = response-header
                  | X-PIRCS-RetrievedURL
                  | X-PIRCS-Status
                  | X-PIRCS-Filter
                  | X-PIRCS-IncludedURL

X-PIRCS-SearchParams =
        “X-PIRCS-SearchParams” “:”
        pircs-filter-name “|” pircs-key

X-PIRCS-RetrivedURL =
        “X-PIRCS-RetrivedURL” “:”
        http_URL*(“|” (1*3DIGIT | “-1” ))

X-PIRCS-Status =“X-PIRCS-Status” “:”
        (PIRCS_OK | PIRCS_ERROR)

X-PIRCS-Filter =
        “X-PIRCS-Filter” “:” pircs-filter-name
        *(“|”pircs-filter-name)

pircs-filter-name = token
pircs-key = token

X-PIRCS-IncludedURL =
        “X-PIRCS-IncludedURL” “:”
        http_URL*(“|” http_URL )
```

pircs-filter-name はフィルタ名であり、prics-key はそのフィルタに与える検索キーである。

3.4 検索モデル

全文検索型 Web 検索モデルにおける、情報探索空間が広すぎるために起こる問題を解決するため、PIRCS では、その探索空間を利用者が指定したコンテンツ (URL) の周辺に限定する。このとき、指定した含リンクコンテンツをスタートページと呼び、そこからリンクを辿ることで取得できる、一定程度の範囲のコンテンツを探索空間に置く。

このように探索空間を限定するのは、リンクで結ばれたコンテンツ間には、特定の分野であるなど関連性（意味の共起性）がみられることが多く、分野を限定した探索を実現できる可能性が高いと考えられるためである。すなわち、同一の Web コミュニティに属する可能性の高いコンテンツ内で検索を行うことを目指す。

また、1 回の検索結果を基に、スタートページを指定し直すことで、探索空間の移動や拡大を行うことができ、問い合わせのキーの追加や変更以外にも、進化的探索を支援することができる。

4 システム構成

AWS 側の検索インデックス構築のためのインデクシングモジュールと、検索応答モジュール、また、WBS 側の全般的な構成について述べる。

4.1 検索フィルタ

テキストだけではなく、画像や音声、また、それらの複合文書など多様なメディアを検索するには、また、利用者の異なった視点からの検索を実現するためには、そのための検索エンジンが必要である。このとき、コンテンツやキーは、メディアの種類によって形式が異なる。したがって、

- インデクシング：コンテンツ ⇒ インデックス
- 距離関数：distance(< キー >, < インデックス >)

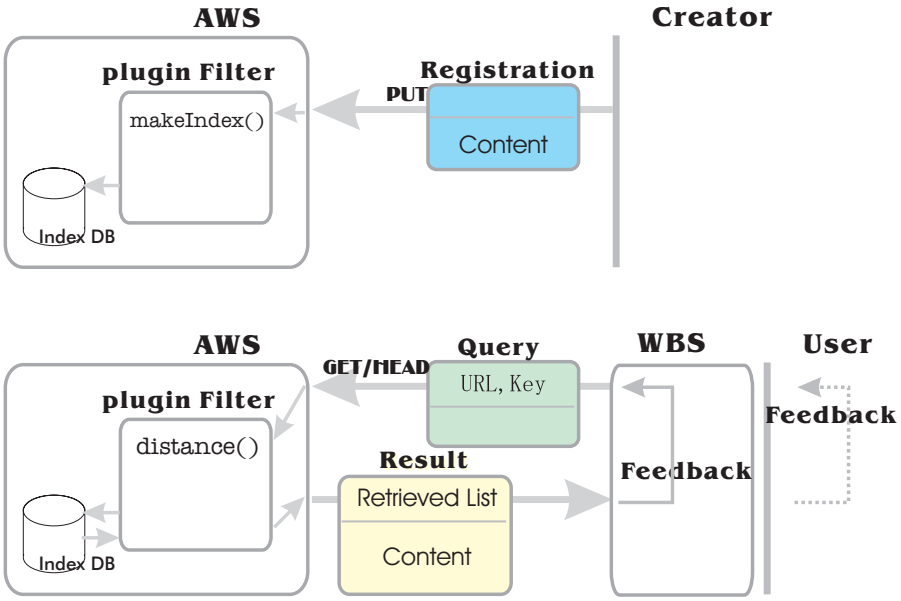

図 1: PIRCS アーキテクチャ

の機能をあわせた、検索フィルタをシステムには含めず、プラグイン形式で利用可能なようにする。

具体的には、一定の規約に乗っ取り、上記の機能性を特定のメソッドで実現する JAVA のクラスとする。

4.2 インデクシング

Web コンテンツのインデクシングは、コンテンツの新規登録・更新時に発生する。また、コンテンツの削除時には該当するインデックスも消去しなければならない。このため、従来 Web サーバとは別個の、FTP などを用いてコンテンツをディスクスペースに置いてきた作業は、すべて Web サーバが行う必要がある。

AWS では、HTTP/1.1 に基づき、PUT および DELETE メソッドを利用する。それぞれ、コンテンツのアップロード、削除を行う HTTP のリクエストメソッドである。インデクシングモジュールは、この 2 つのメソッドに対して起動され、プラグインされた検索フィルタの当該メソッドを呼び出す。

また、当該コンテンツが含リンクコンテンツであるならば、そのリンク先コンテンツを取得し、同様にインデクシングを行う。これによって、検索時の応答速度を高めることができる。さらに、インデクシング発生時には、そのリンク元コンテンツに対して、更新通知を行う。通知された側からみると、リンク先コンテンツの更新を知ることができ、そのインデックスのコンテンツとの一貫性が保証される。

4.3 応答

検索は、HTTP/1.1 の GET および、HEAD メソッド時に行われる。検索応答モジュールはこのとき呼び出され、問い合わせに応答する。問い合わせは、スタートページの URL と、検索に利用したいフィルタ名および、そのフィルタに与える検索キーの組から構成される。フィルタ名は JAVA のクラス名であり、慣習的なパッケージ名の与え方に従えば、不用意に重複することはない。

検索は、start URL に含まれるリンク先コンテンツに対して行われる。このとき、あらかじめ構築しておいたデータベース中のコンテンツに対応するインデックスに対して行うことで、応答速度を確保する。検索結果は、リンク先コンテンツの URL と、各フィルタにおける距離の組のリストになる。また、リンク先コンテンツが含リンクコンテンツであるときには、その URL リストも検索結果に含まれる。これらを HTTP ヘッダに追加し、従来の Web サーバと同様、GET メソッド時には当該コンテンツも送信する。

4.4 検索ユーザエージェント

WBS 側で、どのような検索モデルを実現するか、どのような検索インターフェースを用いるのか、といったことは PIRCS のシステム設計の対象外である。送受信する情報は、AWS の検索応答モジュールに対する説明などで述べた。これらの情報をどのように活かした検索システムを構築するかは、様々な利用モデルが考えられるため、ここでは議論しない。

ただし、典型的には、startURL と利用するフィルタ名、そのキーを入力すると、スタートページからつな

がる、一連のコンテンツ群に対して検索を行った結果が出力される。このとき、コンテンツの含まれるリンクに対しては、その先のコンテンツへの URL リストも同時に提示される。インターフェースは、リンクスポットのクリックや検索キーの書き換えなどによる適合性フィードバックを通じて、進化的探索を支援する。

5 試作システム

5.1 フレームワーク

これまでも述べてきたように、PIRCS は Web 情報探索を行うためのフレームワークであり、実際の検索アルゴリズムはプラグイン形式で別に提供し、広域的な情報探索アルゴリズムはクライアント側に実装される。しかしながら、そのような枠組みが、実際に機能するのかどうかを検証し、試験実装として提供することを考慮に入れて、試作システムを構築した。

PIRCS としての実装は主としてサーバ側にあるが、検索システムとしての動作を確かめるためには適当なクライアントも必要である。そこで、双方を試作した。

5.2 AWS

AWS は、apache の拡張モジュールとして呼び出されるようにした。apache はモジュールの拡張をサポートしており、本システムでは、インデクシング、検索応答それぞれをモジュールとして拡張した。実際には、JAVA として実装されるプラグインの検索フィルタとの適合性を考慮して、処理の大半は JAVA で記述した。起動時のオーバーヘッドを避けるために、これらの JAVA モジュールは常駐プロセスとして機能し、apache の各拡張モジュールからは、これらを呼び出して処理を依頼する形とした。HTTP の PUT、DELETE、GET、HEAD の各メソッドに対応して処理を行う。

DBMS には PostgreSQL を用いた。データベースは、

Filter 検索フィルタに関する情報

URL インデックスを作成する/した URL

Index フィルタ毎に作成したインデックス情報

をそれぞれ作成する。

今回試作した検索フィルタは、

- ファイルサイズ
- ファイルタイプ (拡張子)
- 画像平均色

の 3 つである。

PUT 時は、インデックスの更新処理が必要となる。この処理時間は、使用する検索フィルタやコンテンツによって異なるが、上記のフィルタでは、実際のフィルタリングの処理は 1 秒程度であり、データベースへの格納処理までを含めて 1 コンテンツあたり 3 秒程度であった。

一方、GET、HEAD メソッドに対しては検索処理が必要となる。この場合、指定された URL に含まれるリンク数などによって若干異なるが、処理時間はおおむね 2〜3 秒程度であった。apache 単体での動作では、ほぼ一瞬でコンテンツが送信されるのに対して、かなり長い時間がかかっており、今後の改善が必要である。

5.3 WBS

WBS の実現形態には様々なものが考えられる。ここでは、従来の Web ブラウザを通じて多くの人が PIRCS を用いた検索を行えるようにすることを将来的な目標として、サーブレット形式での実装を試みた。したがって、WBS としての処理は、サーブレットの動作しているマシン上で行われる。このプロセスは、利用者とのセッションを維持し、入力された情報を PIRCS の HTTP ヘッダ形式に変換して、AWS とのやり取りを行う。

試作した 3 つのフィルタのうち、前 2 者は、あらゆる Content-type に適合する。平均色フィルタは、画像ファイルについて、全ピクセルの平均を計算し、その上で距離などを計算するフィルタである。これらによって、動作の確認と画像コンテンツの検索という、従来不可能だった検索の実現性を確かめた。

ユーザインターフェースは、

- 各フィルタのキー
- start URL

の入力部分と、結果の出力として

- 検索結果（上位５件）
- リンク関係（木構造的）
- startURL のコンテンツ

をそれぞれ HTML のフレームなどを用いて表示する。

5.4 検索シーケンス

試作システムにおける検索シーケンスは次のように行われる。

問い合わせの入力 利用者が、使用するフィルタに対するキー、start URL を入力する

問い合わせの変換 WBS が PIRCS 拡張ヘッダを作成し、指定 URL へ GET メソッドを発行する。

問い合わせの応答 AWS が問い合わせの応答を作成し、WBS へ送信する。

コンテンツの表示 WBS は start URL のコンテンツを表示する。

探索空間の追加 WBS は、結果中から適当な URL を選択して、再び問い合わせを行う。これを繰り返す。

HTML の機能を利用して、ブラウザから一定時間毎に WBS の状態を取得して表示するようにして、検索の実行状況を確認した。

6 おわりに

PIRCS は、従来超分散構造であった WWW アーキテクチャに、一定の協調機能を導入して、Web のリンク構造を考慮した検索を提供しようとするフレームワークである。本稿では、WWW 検索モデルとしてどのようなものが考えられるかを整理した上で、PIRCS のアーキテクチャについて説明した。また、試作システムを構築して、その動作を確認したので、試作システムについても報告した。

今後は、更新通知に関して、検索モデルとの関係を理論的に探っていきたい。また、その検索モデルに求められる WBS の機能を明らかにしていく予定である。さらに、多数の AWS を設置して、広域環境での動作を実験的に確認し、PIRCS を利用した検索が有効に機能することを確かめる必要もあろう。

参考文献

[1] http://www.yahoo.co.jp

[2] http://www.excite.com

[3] http://www.goo.ne.jp

[4] http://www.google.com

[5] 徳永健伸, “言語と計算—5 情報検索と言語処理” 東京大学出版会 1999.

[6] 来住伸子，大森貴博，水谷正大，小川貴英，“検索エンジンを利用した日本語 Web ページ数の統計的推計”，Proc. of DBWeb2000，pp.149–156，2000.

[7] 菅原真司, 山岡克式, 酒井善則, “ネットワークにおける画像情報の効率的探索法に関する検討” 信学論 (B-I),vol.J81-B-I,no.8,pp.484-493,1998.

[8] 内藤清一郎, 小林亜樹, 山岡克式, 酒井善則，“超分散サーチエンジンを用いた効率的情報探索について”, 信学技報, **SSE99-200, IN99-163**, pp.123–128, 2000.

[9] 山岡克式, 内藤清一郎, 小林亜樹, 酒井善則，“超分散サーチエンジン”, 信学会総合大会, **B7-120**, 2000.

[10] 小林 亜樹，新名 崇，内藤 清一郎，酒井 善則，山岡 克式，“PIRCS: リンク情報の自律的収集による Web 情報探索”，電子情報通信学会技術報告，**SSE2000-116**， pp.7–12 (2000)

[11] 樋山 大輔, 内藤 清一郎, 小林 亜樹, 山岡 克式, 酒井善則，“超分散型 Web 検索システム PIRCS の試作”, 電子情報通信学会技術報告，**SSE2000-239 IN2000-195**，pp.25–32 2001.

[12] Sergey Brin and Lawrence Page, “The anatomy of a Large-Scale Hypertextual Web Search Engine.” WWW7, Computer Networks 30(1-7), pp.107–117, 1998.

[13] Jon M. Kleinberg, “Authoritative Sources in a Hyperlinked Environment..” Proc. of ACM-SIAM Symposium on Discrete Algorithms, 1998.

[14] IBM Almaden Research Center, http://www.almaden.ibm.com/cs/k53/clever.html

[15] Sergey Brin, Rajeev Motwani, Lawrence Page, Terry Winograd, “What can you do with a Web in your Pocket?” IEEE Computer Society, Bulletin of the Technical Committee on Data Enginnering, Vol.21 No.2, pp.37–47, 1998.

[16] 久我昌崇，中所武司，“Web コミュニティの知識に基づく情報検索手法の評価”，情報処理学会研究報告，**DBS123-6**，pp.37–44，2001.

[17] 豊田正史，“WWW における関連コミュニティ群の発見”，情報処理学会研究報告，**DBS122-40**，pp.307–313，2000.

[18] Bates, M. J. “The design of browsing and berrypicking techniques for the online search interface”, Online Review, **13**, 5, pp.407–424, 1989.

[19] http://www.ietf.org/rfc/rfc2068.txt