

データベースとインターネットアプリケーションの 連携に関する試み

久保田和己 石川博
富士通研究所

インターネット上の情報検索ツールとしての WWW (World Wide Web) が注目されている。WWWはネットワークで利用可能なサービス全体を単一の巨大なハイパーテキストと考え、様々なアプリケーションを統一された操作性で利用できる分散ハイパーテキストシステムである。しかしながら WWW サーバはクライアントからの要求に対して保存されているデータを渡すだけの単純なものであり、それに認証機能が追加された経緯がある。

すなわち現在の WWW システムのサーバやクライアントではデータ管理機能の不足から、一貫性の維持や、資源の名前空間の管理が困難であることなどのさまざまな問題点が生じている。

我々はデータベースの研究の立場からマルチメディア分散情報システムとして WWW システムをとらえ、データベースシステムとWWWシステムの連携によりさらに強力なデータ管理機能を提供できるようにしていく必要があると考える。

本稿では、データベースシステムの連携によるWWWサーバ機能の支援、データベースシステムの情報管理機能を利用した WWW サーバ構築支援システム、ネットワーク上の資源の所在をデータベースによって管理するディレクトリサーバの構築の3つのアプローチを中心に述べる。

On Integration of Database Systems and Internet Applications

Kazumi KUBOTA and Hiroshi ISHIKAWA
FUJITSU LABORATORIES LTD.

WWW (World Wide Web) has been proposed in order to enable shared access to the information resources created by multiple autonomous applications in the INTERNET. WWW provides World Wide Hypertext to integrate the distributed resources and information navigational tools. Currently WWW servers have been developed simply to reply the files requested by users. However, as the files tend to be large scale, WWW servers do not satisfy data management requirement sufficiently.

Database systems we thought to be helpful as a data manager for the WWW system in several ways. In this paper, we propose various approaches to the integration of WWW and database systems, which include gateway systems, support systems for contents creation, and information directory server systems.

1. はじめに

1.1 WWW とは

インターネット上の情報検索ツールとしての WWW (World Wide Web) [1] が注目されている。

WWWはもともとスイスの CERN において、物理学者たちの研究活動を支援する目的で開発された分散ハイパーメディアシステムであり、ネットワークで利用可能なサービス全体を単一の巨大なハイパーテキストと考え、様々なアプリケーションを統一された操作性で利用できるという画期的なものであった。当初 CERN で開発されたクライアントソフトは文字ベースのものであったためにあまり普及しなかったが、NCSA Mosaic の登場以降 GUI ベースのマルチメディア化されたクライアントソフトがあらわれて、インターネットの世界で熱狂的に受け入れられ、短期間で急速に普及した。

WWWを構成する中心的概念はHTML[5], URL, HTTP[4] の3つである。

HTML: ハイパーテキストを記述するための言語
URL: インターネット上の資源を指定する記述法
HTTP: WWWのサーバとクライアントでハイパーテキストをやりとりするためのプロトコル

前述したように Mosaic は WWWシステムのクライアントアプリケーションの1つで HTML 文のブラウザである。本稿ではWWWという用語をHTTPサーバ (httpd) とクライアントおよび、そのあいだのプロトコルであるHTTPを包含したシステムをさすものとして用いる。

1.2 現在の WWW システムの問題点

WWW関連の分野で現在問題になっている点や今後の方向性について述べる。現在WWWシステムのサーバやクライアントには、データ管理機能の不足から次の問題がある。

○ビジネス利用とセキュリティ

WWWシステムをビジネスで利用するためにはユーザの認証が最低限必要である。現在のHTTPプロトコルでは認証のための情報を交換できる仕組みが備わっているが安全性は低い。

○クライアントの機能アップ

クライアント側で複数のアプリケーションを外部ビューワーとして起動する仕掛けは備わっているがそれらを組み合わせた環境で切れ目のない操作環境は現時点では提供されていない。ただしクライアントのOLE化 (サーバからWindowsアプリケーション用のデータを渡し、それを直

接利用できる) などが提案されている。

○サーバの機能

もともと WWW サーバはクライアントからの要求に対して保存されているデータを渡すだけの単純なものであり、それに認証機能が追加された経緯がある。現時点ではサーバに格納されているデータの管理機能は不十分である。

○リソースの名前付け

WWWサーバが提供するHTML文、画像、音声などのマルチメディアリソースはすべてURLを用いて位置に依存した記述で行われる。URLはファイルシステムのディレクトリ階層に基づいている。従ってリソースの位置が変わるとそれを参照しているURLをすべて変更しなければならず、一貫性を維持するのが困難である。またURLの一貫性も保証されていない。例えば、あるURLを参照しているハイパーリンクがHTML文中に記述されていたとしても、その参照先のリソースが実際に存在するかどうかは保証されていない。

○ディレクトリサービス

ハイパーテキストの問題点として、特定の経路をたどるようなナビゲーションは可能であるが、自分が必要としている情報がどこにあるかを知らなければ、その情報にたどり着くことが困難であるという問題がある。

そのためにインターネット上のどこにどのようなリソースがあるかという情報を提供するディレクトリサービスが必要である。

現在はCERN, NTTなどの特定のWWWサーバの管理者にリソースの所在をEmail等で連絡することで登録される運用形態がとられている。

あるリソースに更新が起こったことを伝達する有効な方法はまだ提案されていない。例えば、ユーザにとって有益な情報が追加されたり更新されたらいい、そのことを知ることは難しい。

○ハイパーテキスト (HTML文) の作成

ハイパーテキストはリンクのタグを文章中に埋め込むことで作成される。一般に HTML 文を記述することは難しいとされている。また文法を覚えなければならないといった問題以上に、リソースの名前付けの問題と関連して一貫性を保つように記述することは困難な作業である。

現時点では通常のプレーンテキストをHTML文に変換することやTeXや SGML、ワープロの格納形式などの制御文字を用いて構造化された文書を変換したり、HTML用のテキストエディタなどが提案されている。

2. WWWとデータベースとの連携

2.1 DB との連携による問題点の解決

データベース管理システムと WWW サーバを連携することで前述の問題点の解決を考える。

○データの管理とセキュリティ

WWWサーバの機能を拡張してデータ管理機能やセキュリティ機能を充実することがもちろん必要であるが、WWWサーバのバックエンドにデータベース管理システムを置くことで、データベース管理システムの提供するデータ管理機能やセキュリティ機能をそのまま利用できる。

○リソースの名前管理

データベース管理システムを用いてマルチメディアリソースを管理し、HTML文を動的に生成することでリソースの名前管理の問題は大幅に改善することができる。

リソースの指定を URLで行っていた部分は、その1つ上の層で位置に独立な識別名をつけて両者の対応をデータベースを用いて管理することも可能である。また、キーワード等をマルチメディアリソースに付与することで、必要なリソースを検索し提供することも可能になる。

HTML文を動的に生成することで名前の参照の一貫性を保証することも可能である。

○ディレクトリサービス

インターネット上のどこにどのようなリソースがあるかをデータベースで管理し、WWWシステムを通してこのデータベースにアクセスできる仕組みをつくることでディレクトリサービスを実現することができる。

○ハイパーテキスト作成の支援

単にハイパーテキストの作成を支援するだけでなく、WWWサーバで提供する様々なマルチメディアリソースをデータベースに格納し、必要なリソースを管理することが可能である。またこれらのリソースを元にしてHTML文の自動生成を行ったり、データベースへの問い合わせのための (httpd から実行される) プログラムを生成することもできる。

以上のように WWW システムと データベース管理システムの連携は WWW システムがもっている多くの問題点を解消するための手段として有効である。また、データベースシステムから見た利点として、WWW のクライアントをユーザインタフェースとしてデータベースへのアクセスをユーザに提供することが可能になる。

インターネットというとMosaicと言われるほど

にWWWシステムの普及はめざましいものである。このなかで、我々はデータベースの観点からマルチメディア分散情報システムとして WWW システムをとらえ、データベースシステムとWWWシステムの連携によりさらに強力な機能を提供できるようにしていく必要があると考える。

2.2 連携のためのアプローチ

前述のように WWW のデータ管理の機構として、データベースシステムを適用することは有効である。そこで、WWW と データベースシステムとの連携に関して行っている我々のアプローチを紹介する。データベースシステムとして、オブジェクト指向データベースシステム ODB II [2]を用いている。

○WWW 連携マルチメディア情報システム

WWWサーバのバックエンドに ODB II を置き、マルチメディアリソースを ODB II のデータベース上にオブジェクトとして格納することで WWWサーバのデータ管理機能を支援する。WWWサーバと ODB II をつなぐゲートウェイを提供し、ODB II のデータベースを WWW のインタフェースを通してのアクセスを可能にする。

○WWWサーバ構築支援システム

WWWサーバを構築する際に必要となる HTML 文の作成環境を含めて、マルチメディアリソースを データベースで管理し、ハイパーテキストのページを記述する際に必要となるマルチメディア部品 (テキスト、画像、音声など) の名前の管理やデータベースへのアクセスインタフェースの自動生成を行う。

○ディレクトリサービス構築システム

ディレクトリサービスを実現するための手法をディレクトリサーバ構築支援、ディレクトリサービスの利用方法、ディレクトリ情報の配送方式など広い視野で検討する。

これら3つのアプローチについてそれぞれくわしく述べる。

3. WWW連携マルチメディア情報システム

3.1 概要

このアプローチの目的は、マルチメディアリソースを ODB II のデータベース上にオブジェクトとして格納することで WWWサーバのデータ管理機能を支援することである。具体的には、WWWサーバのバックエンドにオブジェクト指向データベース管理システム ODB IIをおき、

Mosaic等のクライアントをユーザインタフェースとしてデータベースへのアクセスを可能にする。これを実現するために、WWWサーバとODB IIをつなぐゲートウェイを作り、ODB IIのデータベースをWWWのインタフェースを通してアクセスする実験を行った。システムの基本構成を図1に示す。

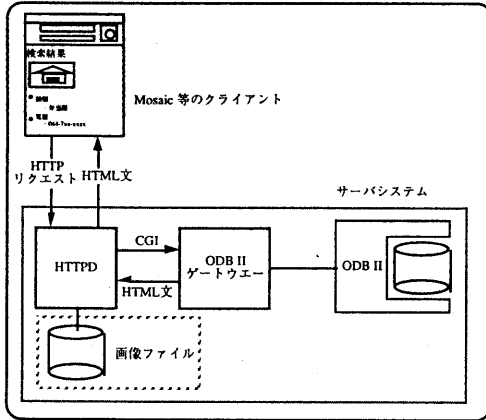


図1 システム基本構成図

システムの構成は、次のコンポーネントからなる。

○サーバシステム側

- ・ ODB II
- ・ ODB II ゲートウェイ
- ・ HTTPD
- ・ 画像やHTML文を格納するファイルシステム

○クライアント側

- ・ WWW クライアントアプリケーション

各コンポーネントについて述べる。

ODB II ゲートウェイ

HTTPD から CGI (Common Gateway Interface) を用いて呼ばれる ODB II データベースアクセスのためのコマンドの集合である。各コマンドは ODB II のアクセス言語である NEL の動的実行モードを使用し、DBアクセスを行う。

ユーザからの問い合わせがあるごとに上記のコマンドが起動され、コマンドの実行結果は HTML文として標準に出力される。

CGI インタフェース

httpd は必要に応じてC でかかれたコマンドを実行することができる。このときに使用されるインタフェースがCGIインタフェースである。コマ

ンドへの入力引き数は環境変数を通してわたされる。呼び出されたコマンドは実行結果として HTML文を標準出力に出して停止する。

3.2 実験システム

実験システムとして次のシステムを試作した。

川崎工場周辺オレンジページ

川崎工場周辺の地理案内システム

NCSA httpd 1.3

ODB II 1.0 Solaris 版

ODB II に川崎工場周辺の建物に関するデータを格納し、ユーザの要求に応じて地理案内を行うシステムである。画面のイメージを図2に示す。

動作

建物の名前や位置、種類などをフォーム形式でユーザに入力させ、該当する建物の名前の一覧を表示する。

建物の名前がキーになっているので、名前の一覧のなかから1つを選択して再度問い合わせを行うと、その建物に関する詳細情報(建物の写真、名前、住所、電話番号、レストランのメニュー、商店で扱っている品物、関連するビデオ映像、地図上での位置など)を表示する。

クラス階層

このシステムのために ODB II に定義したクラス階層は次の通りである。

```

class 建物 {
    string 名前 unique
    int X座標
    int Y座標
    int エリア番号
    string 写真ファイル名
    text 住所
    text 電話
    text コメント
    string ビデオファイル名
}
  
```

```

class 商店 superclass 建物 {
    string 種類
    set of string 商品
}
  
```

```

class レストラン superclass 建物 {
    string 種類
    string 和洋中
    string 価格帯
    set of string メニュー
}
  
```

```

}
class 公共の建物 superclass 建物 {
    string 種類
}

class 病院 superclass 建物 {
    string 種類
}

class 会社 superclass 建物 {
    string 種類
}

```

3.3 評価

WWWサーバー側の提供するインタフェースである CGI を用いれば、簡単なゲートウェイコマンドを作成することで、ODB II 自身や httpd に改造を加えることなく ODB II に格納されているオブジェクトを WWW クライアントのインタフェースを用いてユーザに提供することが可能であることがわかった。

このゲートウェイコマンドはクライアントから送られてきた検索条件にもとづいてデータベースにアクセスするためのデータベース文を生成し動的に実行しているだけなので、処理自体は単純であり、テンプレートさえ与えればコマンド自体を自動生成することも可能であると考えられる。

現在のシステムではデータベースのスキーマに依存したアクセス用コマンドを使用していて、データベースのスキーマが変わることにあたりコマンドを作成しなければならない。もちろんスキーマに依存しないような形式でゲートウェイをつくることも可能であるが、実装の容易さからこの方式を用いた。

ユーザが調べたい建物の詳細情報を得るためには少なくとも2回の ODB II アクセスが必要となり、このことがレスポンスを悪くすることになるかと思われたが、実際にはかかる時間の内訳は大半が結果データの転送時間であり、WWW と ODB II の連携によって発生するオーバーヘッドは、この事例ではそれほど大きなものではなかった。

しかしながら、今回の実験で用いた方式ではコマンドが呼び出されるごとに ODB II のセッションが起動されるために、ユーザーからのアクセスが集中した場合動作が不安定になることは明らかであり、あまり実用的ではない。同時に開かれるセッション数の制限などの機能を付加したゲートウェイを導入する必要がある。

ここで行ったような、検索条件にもとづいてデータベースにアクセスするためのデータベース文を生成する方式とは別に、データベースシステムのエンジンに手を入れてデータベース中

のデータ (オブジェクト) を URL を用いて識別する方式を提供するアプローチも必要であろう。

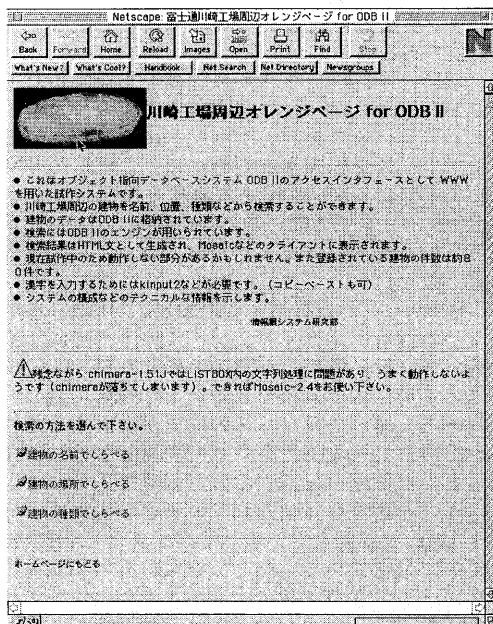


図2 実験システムの画面イメージ

4. WWW サーバ構築支援システム

4.1 概要

このアプローチの目的は、データベース用いて WWWサーバーを構築する際の作業を支援することである。

このためにつぎの3つの機能を提供するHTML文構築環境を提供する。

- ・支援システムの基盤となるHTMLエディタ
- ・HTTPサーバからデータベースを呼び出すためのアクセスインタフェースの自動生成
- ・データベースのオブジェクトスキーマをWWWクライアントに見せるためのHTML文の自動生成

1章で述べたように、WWWサーバーのコンテンツであるハイパーテキストは、リンクのタグを文章中に埋め込むことで作成されるため、リソースの名前付けの問題と関連して一貫性を保つように記述することは困難な作業である。

現時点で通常のプレーンテキストをHTML文に変換することやTeXや SGML、ワープロの格納形式などの制御文字を用いて構造化された文書を変換したり、HTML用のテキストエディタなどが提案されているが、これらはテキスト中

の章だてを修飾するために文字のフォントやポイント数を変える程度の機能を提供しているにすぎない。本質的な問題はハイパーテキスト間のリンク関係の記述をいかに支援するかという問題である。

単にハイパーテキストの作成を支援するだけでなく、WWWサーバで提供する様々なマルチメディアリソースをデータベースに格納し、必要なリソースを管理するためには、これらのリソースを元にしてHTML文の自動生成を行ったり、データベースへの問い合わせのための (httpd から実行される) プログラムを生成することも必要である。

WWWサーバが提供するHTML文、画像、音声などのマルチメディアリソースはすべてURLを用いて位置に依存した記述で行われが、URLはファイルシステムのディレクトリ階層に基づいている。従ってリソースの位置が変わるとそれを参照しているURLをすべて変更しなければならず、一貫性を維持するのが困難である。またURLの一意性も保証されていない。例えば、あるURLを参照しているハイパーリンクがHTML文中に記述されていたとしても、その参照先のリソースが実際に存在するかどうかは保証されていない。しかし、データベース管理システムを用いてマルチメディアリソースを管理し、HTML文を動的に生成することでリソースの名前管理の問題は大幅に改善することができる。

4.2 システムの構成

システムの構成を図3に示す。システムを構成する各コンポーネントについて説明する。

○HTMLエディタ

WWWサーバ構築支援システムの中心となるコンポーネントである。ページエディタと部品エディタからなる。

○部品エディタ

ページエディタで使用する画像、動画などのマルチメディア部品をデータベースに登録するために使用する。これらの部品は論理的な識別名やキーワードをつけてデータベースで管理する。また、ファイルの形で格納されている部品についてはファイル名やディレクトリ上での位置と論理的な識別名との対応を管理することも可能である。

○マルチメディア部品

部品エディタで登録する画像、動画などの資源をマルチメディア部品と呼ぶ。マルチメディア部品は論理的な識別名やキーワードをつけてデータベースで管理したり、ファイルの形でフ

ァイル名やディレクトリ上での位置と論理的な識別名との対応を管理する。

○ページエディタ

HTML文をページ単位で記述するエディタである。部品エディタを用いてデータベースやファイルシステム中に登録されているマルチメディア部品やテキストエディタを用いて作成した文書をレイアウトしたり、HTML文で記述可能なボタンやフィールド、リストボックスなどのフォームをページ上に定義するのに用いる。

○ページテンプレート

各ページはタイトル、テキスト、フォーム、その他のマルチメディア部品の列とみなすことができる。これらの部品の並び方をページテンプレートとして定義することもできる。

○ODB II インタフェース

マルチメディア部品をデータベース (ODB II) に格納したりキーワードや識別名で検索したりするためのアクセスインタフェースを提供する。

○HTMLファイル

HTMLエディタを用いて作成したHTML文をHTMLファイルに格納する。HTMLファイルはHTTPDからクライアントの要求により取り出される。これとは別に、ODB II に格納されているオブジェクトを組み合わせることで動的にHTML文を生成することも可能である。後者の方式はODB II GWコマンドを通して行われる。

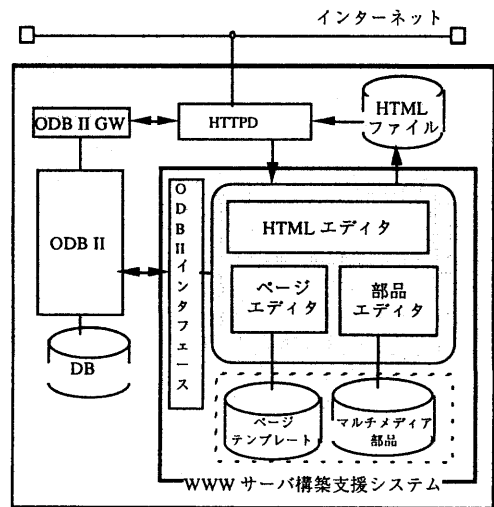


図3 WWWサーバ構築支援システム構成図

4.3 ページの定義とページテンプレート

各ページはページエディタを用いて定義する。

例えば図4のページを定義する場合を考える。このページを記述するためのHTML文は次の通りである。

```
<H1>川崎工場オレンジページ</H1><P>
<IMG SRC="/parts/images/fujitsu_red.gif"><P>
情報網システム研の WWW サーバによるこそ
<P>
<A HREF="q_by_n.html">名前で調べる</A><P>
<A HREF="q_by_p.html">場所で調べる</A><P>
<A HREF="q_by_k.html">種類で調べる</A><P>
<HR>
御質問は kubotak@flab.fujitsu.co.jp まで<P>
情報網システム研究部
```

これを見るとわかるように、このページは、
タイトル 川崎工場オレンジページ
ロゴイメージ 富士通のロゴのGIFイメージ
テキスト 「情報網システム研の WWW サーバ
によるこそ」

他ページへのリンク 名前で調べる
他ページへのリンク 場所で調べる
他ページへのリンク 種類で調べる

横線

テキスト 御質問はkubotak@flab.fujitsu.co.jpま
で

テキスト 情報網システム研究部

という部品の並びで構成されている。ページエディタでは各ページを上記のような部品の並びを定義することでページを記述する。この部品の並びをテンプレートと呼ぶ。

例えば、標準的なホームページ（そのWWWサーバにアクセスしたときに最初に表示されるページ）のレイアウトを考えてみると、

サーバーの名前

ロゴマーク

管理者からの挨拶（お知らせ）

サーバーの最新情報をまとめたページへのリンク

サブページへのリンク

問い合わせ先

という形式が一般的に使われていることが多い。そこでこのような場合、ページエディタを使って、

タイトル

イメージ

テキスト

リンク

リンク

...

テキスト

という部品の並びをホームページのテンプレートとして定義しておくことで、サーバを構築する際には、このテンプレートに実際のタイトルやイメージ、お知らせなどをいれるだけでホームページを作ることができる。

同様に、What's NEW ページのテンプレートなど、よく使われるページのテンプレートをライブラリ化して提供する事も可能である。

もし、ユーザが記述したいページのテンプレートがライブラリの中になければ、ユーザはページエディタを使用して部品の並びから定義することで、自分の希望するページを作成することが可能であり、また、ユーザが作成したページをテンプレートとして定義すればレイアウトの再利用も可能である。

4.4 評価

テンプレートとして部品の並びを登録しておくことで、一般的なページのレイアウトをテンプレートのライブラリとして提供し、ページの記述を効率的に支援することが可能である。また、このテンプレート自体も1つの部品としてデータベース中に格納することも有効である。

実際のファイル名と論理的な名前の対応をデータベースが管理することで、HTML文、画像や音声等のマルチメディア部のファイル名をユーザから隠蔽する事が可能となった。

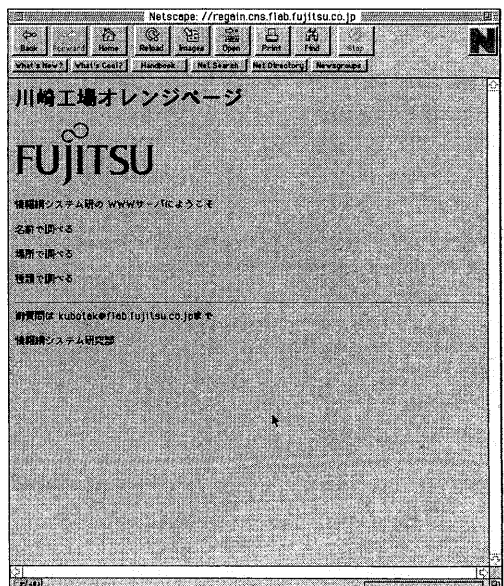


図4 ページの例

5. ディレクトリサービス 連携システム

5.1 概要

このシステムのねらいは、情報提供システムのディレクトリサービスを実現する際の情報収集作業を自動化することで情報収集と検索の効率を向上させることにある。情報収集を自動化するための方法として、サーバ間を渡り歩きながら情報を収集し資源のリストを作成する方式と、情報の提供者がそれぞれ自分が公開したい情報をディレクトリサーバに伝達する方法がある。ここではNews の配送に基づいてディレクトリ情報を限定された範囲でブロードキャストし、後者の方式を用いて情報収集を行うディレクトリサービスについて述べる。

5.2 システムの構成

システムの構成を図5に示す。

本システムでは、(1)WWWサーバ上の情報へのポインタ (URL) をニュースの転送プロトコルを用いて配送する。(2) この記事を受信したサイトではその情報をデータベース (ODB II) に格納する。記事中には情報へのポインタ (URL) とその内容のキーワード、有効期限が記入されている。配送されてきた記事はデータベースに格納され、必要な情報へのポインタをキーワードで検索することが可能となる。(3,4)前述のデータベースとWWWサーバの連携機能を利用して情報へのポインタの一覧を HTMLとして見ることが可能となる。(5)最終的に得られた URL にもとづいて情報源のサーバに到達することが可能となる。

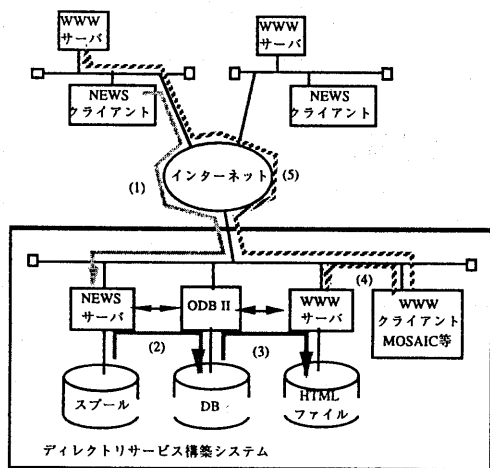


図5 ディレクトリサービスシステム

5.4 評価

この方式の利点は、通常のディレクトリサー

ビスでは情報の提供者が特定のディレクトリサーバに意識的に登録依頼を出さなければならないが、ここで述べたようにニュースの配送システムを利用して制限された範囲の中でリソースの所在情報ブロードキャストすることで、すべてのディレクトリサーバに効率的に情報を伝搬できる点である。

またロボットを用いて他のサーバの内容をサーチするよりも情報に重複が無いために効率的に情報が獲得可能である。

しかしながら、リソースの情報をニュースに投稿してもらうための運用上の問題は残されている。また、配送に遅延が生じる可能性のあるシステムをしているために、遅延によって生じる問題について十分検討する必要がある。

6. おわりに

本稿ではWWWシステムとデータベースシステムの連携によるWWWサーバ機能の支援、データベースシステムの情報管理機能を利用したWWWサーバ構築支援システム、ネットワーク上の資源の所在をデータベースによって管理するディレクトリサーバの構築の3つのアプローチについて述べた。WWWのデータ管理の機構として、データベースシステムを適用することはWWWシステムの持つ一貫性の維持や、資源の名前空間の管理などのさまざまな問題点を解決するのに有効である。

今後はディレクトリサービスシステムの実験等を通して、データベースシステムとWWWなどのインターネットアプリケーションとの連携の方式についてさらに新しい手法を取り入れていくつもりである。

文献

- [1] Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H., et al. (1994). The World-Wide Web. Communications of the ACM, 37(8), pp.76-82.
- [2] Takashi Kanda et al. Object Database Management System : ODB II, FUJITSU, 44 (6), pp.514-520.
- [3] Martin Sjolín, A WWW Front End to an OODBMS, Proc. 2nd International WWW Conf, (1994), <http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Databases/sjolin/sjolin.html>.
- [4] Tim Berners-Lee. Hypertext Transfer Protocol (HTTP). CERN, Geneva, Switzerland, 2.0 edition, 5 November 1993. <http://info.cern.ch/hypertext/WWW/Protocols/HTTP/HTTP2.html>.
- [5] Tim Berners-Lee and Daniel Connolly. Hypertext Markup Language (HTML). CERN, Geneva, Switzerland, 13 July 1993. <http://info.cern.ch/hypertext/WWW/Markup/HTML.html>.