データベースシステム 108－8
（1996. 5. 23）

WWWサーバと連携したアクティブデータベースからの画像検索

保木大典　小西 修
高知大学理学部情報科学科
780 高知県高知市曙町 2-5-1
Tel:0888-44-8342
E-Mail:{ daisukeh,konishi }@is.kochi-u.ac.jp

WWW(World Wide Web)の非構造化された世界、すなわち、インターネット上の非公式に結合された情報源であるWebには、ますます多くの情報が適応されている。これは、従来の構造化されたデータベースの世界にも影響を与えている。我々は、Webは、構造化されたデータベースに対しても、オープンで使いやすいインターフェースを提供するという視点からのアプローチを行なった。本論文では、時間的、空間的に関連した複数の画像データベース（気象衛星画像）を対象に、アクティブデータベースとWWWサーバを結合させた場合の効果的な利用法について述べる。また、アクティブルールを使った仮想ビューと実体化ビューの適用について考察する。

Retrieval from Active Database connected with WWW Server of Images

Daisuke HOKI　Osamu KONISHI
Dept.of Imformation Science,Faculty of Science,Kochi Univercity
2-5-1 Akebono-cho Kochi 780 Japan
Tel:0888-44-8342
E-Mail:{ daisukeh,konishi }@is.kochi-u.ac.jp

In the World-Wide Web(WWW), increasingly more information will be available on the Web, a collection of informally connected resources on the Internet. The facts affect and impact the well structured world of the databases. The World Wide Web may provide open and easy-to-use interface against structured databases. Selecting from a large, expanding collection of images such as weather satellite images requires carefully chosen search criteria. Here is an approach that integrates an active database system with a WWW server. We consider to implementing the virtual views and materialized views using active rules.

1 はじめに

近年、データベースは、その対象として画像を管理することが求められてきた。例えば、気象画像や、絵画など、データベースとして保管しておきたいものに対して、画像データベースと呼ばれるものが作られている。

更にこれらのデータベースは、多くのユーザが容易に利用できるインターネット上で、オープンであるべきである。

しかし、実際上、現在主流である情報検索システムのWWW[6][7]は、その管理方法が静的である。従って、日々刻々と変化するデータを取り扱う、気象画像などをうまく管理する画像データベース管理システムがなかった。

そこで、今回は以上のような問題を解決するために、リレーショナルデータベース拡張機能を用いて、WWWと連携した画像データベース管理システムを提唱し、更に、ユーザに対し、動的に働き掛けるルール機能を用いアクティブデータベースを用いた。

2章で、アクティブデータベース、WWWなどの、基本概念。3章で、今回構築したシステムの説明。4章で実際例を見つつ、検証をする。

2 設計概念

従来、データーベースを利用するには、様々なデータベースを操作するための知識が必要であった。これは、利用者を著しく限定することであった。これが、様々な利用者が存在する、自然科学分野の画像データーベースなどであれば、その欠点は致命的である。また、その情報を管理している場所が、ローカルなものであれば、そのデータベースの存在価値そのものが、希薄なものになる。そこで、今回は、インターネット上で、もっとも普及している、情報検索システムである、WWWを利用することにより、世界中のあらゆるユーザから、利用可能な環境を構築する。

更に、先程述べた、データベースの利用の簡略化であるが、これは、WWWサーバ内に設定されるCGI(common Gateway Interface)スクリプトを利用することによって、解決される。これは、WWWクライアントと、WWWサーバと連携しているデータベースとの間にあって、両者との間を取り持つゲートで、クライアントは、データベース側の操作を知らなくてもよく、また、データベース側も、ユーザに対しての提示の仕方は、考えなくてもよい。CGIが、それらの作業を取り持ってくれるのである。これによって、インターネット上で、あらゆるユーザが、容易にデータベースを利用する環境が、構築される。

ところで、基本的なデータベースの利用方法は、必要な情報が複数存在する場合は、その情報が含まれるテーブルを、一つ一つ選択していかないといけない。これは、アクセス時間の増大などを引き起こす。現在のWWWの利用の仕方などが、それに近いものであろう。そこで、考え出されたものが、Viewである。これは、必要とする情報を組み合わせて、新たなテーブルを作成し、表示するものである。これによってユーザは、必要な情報のみを検索することが可能となる。しかしながら、このViewは、あくまで、ユーザからの操作によって、作成される静的なものである。そこで、ルール機能を有するアクティブデータベースを利用し、その上でのVirtual View機能[?]によって、ユーザからの依頼にたいし、データベース側から動的にViewを作成し、提示するようなシステムを、設計した。

以上のような機能によって、データベースの知識のない一般的なユーザであっても、インターネット上で、WWWを利用することによって、容易にデータベース上の情報を検索できる環境が、構築されることとなる。

3 システム設計

本研究のために構築したシステム[1]を、図1に示し、以下に、各要素の説明を行なう。

3.1 拡張RDBMSとWWWサーバ、CGI

まず、今回利用した拡張リレーショナルデータベース管理システム（以下拡張RDBMS）[5]であるが、その元となるリレーショナルデータベース

[1] http://cygnus.is.kochi-u.ac.jp/weather_query.html に公開してあります。

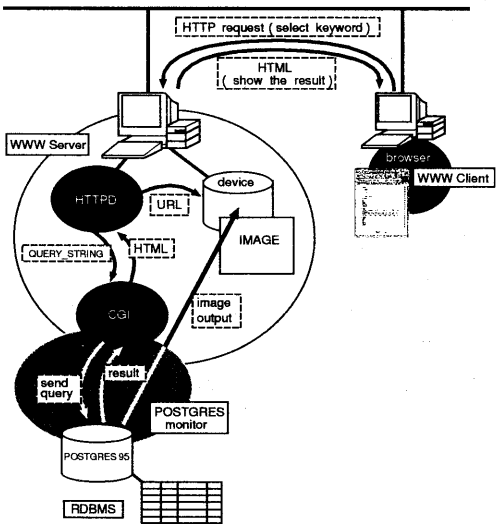

図 1: 拡張 RDBMS に基づく WWW サーバを用いた画像データーベースの構成

(RDB) というものは、文字、数値、書誌情報、画像などが挿入されている。従来の RDBMS では、文字、数値、ある程度の書誌情報は、取り扱うことが出来たが、画像などの大容量データなどは取り扱えなかった。そこで、拡張 RDBMS が登場した。拡張された機能として、大容量データを取り扱え、データのインテグリティ (整合性) を管理する機能である、参照整合性機能。参照関係のある複数テーブル間で、データ整合性を管理し、かつ、業務処理の中核部分 (手続き) を、RDBMS に埋め込み、RDBMS 自身の判断によりそれを実行する機能である、トリガー機能。トリガー機能の一種で、あらかじめ SQL 文を実行可能な形で RDBMS に置いておく機能の、ストアド・プロシジャー機能などがある。

今回使用した拡張 RDBMS は、学術研究用に開発された POSTGRES95[3] で、これは、大容量からなるテキストの、データ管理をサポートしている **Large Object 機能** [4]、そして、トリガー機能として、**ルール機能** [1] を持ち、これによって異種画像間の関連作業を行なう [2]。

また、WWW サーバについてであるが、WWW とは、ネットワーク上で利用可能なサービス全般を一つの巨大なハイパーテキストと考え、それを利用することを目的とした、分散ハイパーメディアシステムである。この、WWW サーバの問題点として、ファイル管理問題がある。これは、WWW サーバ上の情報は、静的なファイルとして管理されているので、自分の必要な情報がどこに存在するのかを知らない場合に、そこへ辿り着くのは困難である。そこで、この問題は WWW サーバのバックエンドに DBMS を配置し、CGI を用いて、検索要求を DBMS に渡し、データベースから出力された検索結果は、sed(script editor) というスクリプトにより、内容が解析され、結果を HTML 文へ受け渡すことにより解決できる。

ここで、CGI スクリプトであるが、HTTPD が必要に応じて、C 言語やシェル、Tcl 言語、Perl などで書かれたコマンド、プログラムを実行する際に使用するスクリプトで、コマンドへの入力引数は、環境変数を通して渡され、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 気象画像と Large Object 機能

先に述べたように、今回使用する POSTGRES95 は、Large Object 機能を有している。POSTGRES95 上では、通常のタップルは、その大きさが 8K バイトに制限されている。そこで、8K バイトを越える属性を扱う場合は、Large Object 機能を用いることにする。画像データの場合、ピクセル値が 100×100 の GIF フォーマットであっても、10K バイト近くなり、今回取り扱う画像である、気象画像などは、一つのタップルに挿入することが出来ない。そこで画像データを Large Object 機能によって管理する。ここで、POSTGRES95 上での、Large Object の取り扱い方法であるが、Large Object を塊に分解し、データベース内のタップルに保存する方法を取っている。Large Object は、オブジェクト識別子 (OID) と、インデックス (索引) とテーブルの連携で管理しておくことができる (図 2)。この機能を利用して、今回は気象画像の画像データベースを構築した。構築したテーブルは、以下の 3 テーブルである。

```
polar(year,month,day,time,image)
asia(year,month,day,time,image)
globe(year,month,day,time,image)
```

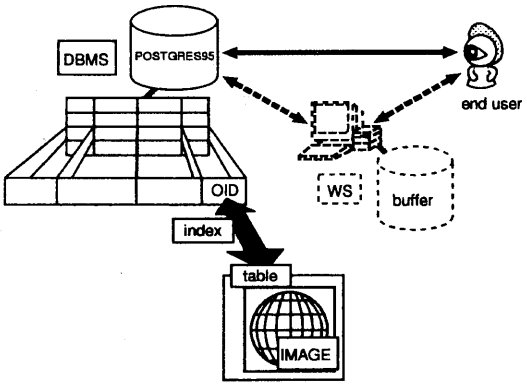

図 2: Large Object 機能による画像管理方法

それぞれ、日本近辺、アジア近辺、東半球の画像であり、テーブル内の image に OID が格納されている。

3.3 仮想ビューと実体化ビュー

最初に、ビューをサポートしているルール機能の説明をしておく。ルール機能とは、テーブル（ビューテーブルを含む）の、ある項目の選択、更新、挿入、削除を行なった際に、関連するテーブルについて、あらかじめ定められ、RDBMS に格納された手続きを、RDBMS が自動的に実行する機能である。ルール機能の書式フォーマットは、ECA(Event Condition Action) ルールという概念に基づいて構成されている。ECA ルールとは、イベントが起きた際に、そのイベントに対するコンディションに従って、アクション部分を実行するものである。このことによって、従来のデータベースでは、ユーザが問い合わせをしてはじめて処理を行なうのに対して、アクティブデータベースでは、一般に、あるイベントが起こると、それに反応して、別のアクションが起動される。

また、他の DBMS の同様な機能を持ったトリガー機能では、ユーザやアプリケーションに関係なく、トリガー文が発行されると自動的に実行されていたが、ルール機能では、ユーザ定義ができ、特定ユーザのみにルール機能を施行させることが可能である。

ルール機能のシンタックス、イベント、コンディション、アクションの対応は、以下のようになる。

```
CREATE RULE < rule_name > AS
 ON [SELECT|UPDATE|DELETE|INSERT] ->Evt
 TO object  ----------------------->Evt
 [WHERE  qualification ]  --------->Cnd
 DO [INSTEAD][action|nothing| ----->Act
                    [action..]]; -->Act
```

イベントには、どの Object に、select、update、delete、insert のいずれかが起こったかを記述でき、WHERE 以降で示されるコンディションは、イベントに対してのものであって、SQL 文に対応している。アクションもまた SQL コマンドの集まりである。またアクション部の INSTEAD 句の有無によって、アクションをイベントと同時に実行するか、イベントの代わりに実行するかを指定できる。即ち、DO [action] だと、イベントとアクションが同時に起こり、DO INSTEAD [action] では、イベントの代わりとして、アクションが実行される。

以上のようなルール機能が、ビュー・メカニズムをサポートしている。ビューとは、SQL 文の select 文を発行する際に、任意に複数テーブルを結合させ、検索結果を出力したり、あらかじめ任意のテーブルを結合させておき、データーベース内に論理的なテーブルを作成しておくことを指す。

ここで、実際に仮想ビュー (Virtual View) と、実体化ビュー (Materialized View) について説明をする。例として、以下の 3 つのテーブルを用意した [1]。

```
Order(orderNumber,Client,Status,Amount)
Notice(OrderNumber,Bank,State,Date)
Account(Client,Bank)
```

ここで、CriticalOrder というビューを考える。このビューは、銀行口座 (Bank account) を持たず、その社会的地位 (Status) は不確かなもので、貸付金 (Amount) の量が 50 を越えている依頼人 (Client) のリストを提供するものである。このビューは、以下のように定義される。

```
CREATE VIEW CriticalOrder AS
      SELECT Order
        FROM Order
       WHERE Order,not Account,
             Status != approved,
             Amount > 50;
```

これを、仮想ビューとして実装するために、select イベントによって、CriticalOrder テーブルを作成するアクティブルールを、以下のように定める。

```
CREATE RULE R1 AS
        ON select TO Order
     WHERE true
        DO select OrderNumber,Client,Amount
             from Order
            where Order,not Account,
                  Status != approved,
                  Amount > 50;
```

これによって、Order テーブルを select すると同時に、CriticalOrder テーブルが作成され、Order テーブルと共に提供される。即ち、ユーザが命令を出さなくても、自動的に DBMS 側の方で、CriticalOrder テーブルを提示し、ユーザに有効な情報を与えるのである。また、ルール部分でユーザの識別を行なわせることが出来る。このことを利用して、ユーザに対して、より動的に働き掛けるビューを提供できる。上述した CriticalOrder と、3つの基本テーブルを使って、例を上げてみると、

```
CREATE RULE R11 AS
        ON select TO Order
     WHERE true,user()='A'
        DO nothing;

CREATE RULE R12 AS
        ON select TO Order
     WHERE true,user()='B'
        DO action 1;
```

ただし、ここでルール R12 の action 1 とは、ルール R1 に出てきた、アクション部分と同一であり、user() は HTML の INPUT TYPE="hidden" などで情報を得るものとしておく。この2つのルールによって、ユーザが A であれば、Order テーブルのみが提示され、ユーザが B であれば、CriticalOrder ビューも、Order テーブルと共に提示される。このように、仮想ビューを利用することによって、ユーザに対して、よりアクティブに対応することが可能である。

では、ここでもう一つのビューである実体化ビューについての説明をする。実体化ビューとは、仮想ビューがユーザが起こした、select イベント発生時に作成されるのに対して、あらかじめ作成されて保存されているビューである。したがって、実体化ビューの利点として、処理速度が、仮想ビューと比べると高速であることが挙げられる。

また、実体化ビューに対するアクティブルールの働きであるが、実体化ビューの基になる基本テーブルに対して、select や、delete が行なわれた場合、必要があれば自動的に実体化ビューの方も、その select、delete に対しての変更が行なわれなくてはいけない。そこで、これを自動的に実行するために、ルール機能が利用される。例として、先程、仮想ビューで使用した、CriticalOrder ビューの基になるテーブルに対しての insert が起こった場合に使用されるルールを考えてみると、

```
CREATE RULE R2 AS
        ON insert TO Order
     WHERE inserted[order],
           not Account,
           Status = approved,
           Amount > 50
        DO insert into CriticalOrder;
```

というルールが、考えられる。ただし、WHERE 部の、inserted とは、すでに、insert イベントが完了している、という事を示している。このルールは、基本テーブルの1つである Order テーブルに、あるタップルが挿入された場合に自動的に起動され、Account を持ってなくて、Status が、approved で、Amount が 50 を越えるタップルならば、CriticalOrder ビューに挿入しなさい、というものである。

また、実体化ビューにおいて、基テーブルのデータに変化が起こった際のビューの変化を計算するアルゴリズムに、incremental view maintenance というものがあり、これは実体化ビューを更新するために、ビューの変化した部分のみを計算するというものである。

こうしたルールによって、実体化ビューは維持される。

3.4 アクティブルールを用いたビューの実装

それでは、今回実装したビューを示す。

```
CREATE VIEW area(year,month,day,time,
              asia.image,globe.image) AS
     SELECT polor(year,month,day,time,
              image)
       FROM asia(year,month,day,time,
              image),
            globe(year,month,day,time,
              image)
      WHERE asia.year=polor.year,
            asia.month=polor.year,
            asia.day=polor.day,
            asia.time=polor.time,
            globe.year=polor.year,
            globe.month=polor.year,
            globe.day=polor.day,
            globe.time=polor.time;
```

で表わされるビューとなり、次のルールによって、サポートされる。

```
CREATE RULE r1 AS
    ON select asia(year,month,day,
                 time,image)
    DO select year,month,day,time,
              asia.image,globe.image
       from asia(year,month,day,
              time,image),
            globe(year,month,day,
              time,image)
      where asia.year=polor.year,
            asia.month=polor.year,
            asia.day=polor.day,
            asia.time=polor.time,
            globe.year=polor.year,
            globe.month=polor.year,
            globe.day=polor.day,
            globe.time=polor.time;
```

このルールによって、ユーザが polor のテーブルを select した場合に、asia と globe のテーブルとを結合させたビューも、polor テーブルと共に表示される。

4 検索の実現

4.1 検索例

実際例として、1996 年 1 月 15 日の画像を選択して、要求する気象画像を得るまでを述べる。最初に、気象画像データベースの初期設定ページで、初期入力値として、年月日を選択し、間違いがなければ、SUBMIT ボタンをクリックする。その結果 1996 年 1 月 15 日の日本近辺の全気象画像が表示される (図 3)。その中に要求する画像が存在したら、画像をクリックする。この例では、1996 年 1 月 15 日 6 時の画像を選択した (図 4、図 5)。ここで、先ほど説明した仮想ビューの働きで、1996 年 1 月 15 日の日本近辺の画像だけでなく、同時刻における他の気象画像 (アジア近辺、東半球) も同時に表示される。この時に、もし他の画像が検索されず、画像が出力できないときは、その地域の画像は表示されない。後は見たい画像、各々をクリックするだけで、各地域の元画像を見ることができる (図 6、図 7、図 8)。

4.2 結果と考察

今回、拡張 RDBMS の機能である Large Object 機能を用いて気象衛星画像データベースシステムを構築し、アクティブデータベースの一概念であるルール機能を用いて、異種画像間の関連操作を行ない、また、従来の WWW サーバの問題であった、静的なデータの扱いをデーターベース管理システムを用いることで、動的に扱うことに成功した。このことにより、本研究の目的を実現できた。

今回は、仮想ビューを用いた画像検索を行なって一応の成功が見られたが、その問題点も発見できた。Large Object を用いた画像出力であるが、monitor などの、POSTGRES95 に付属しているアプリケーションは Large Object 機能を用いて、一つ一つの画像を出力するのに一回一回検索要求をしなければならず、それに伴う通信量の増加によるスループットの減退があり、画像出力に対する遅延問題がある。ただ、この問題は、現在使用しているハードウェアに依存しているので、これ以上の改善はしようがない。ただし、前者の問題は、monitor のソースプログラムや、POSTGRES95 の DBMS 内の Large Object 機能の部分などを理

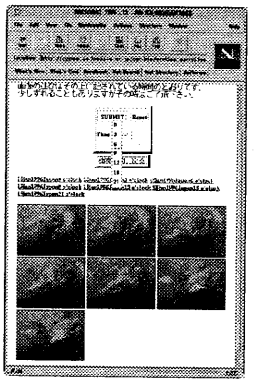

図 3: 1996 年 1 月 15 日における日本近辺の全気象画像

図 6: 1996 年 1 月 15 日 6 時の日本近辺の元画像

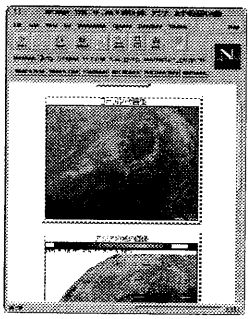

図 4: 1996 年 1 月 15 日 6 時の 3 地域の複数画像、上部

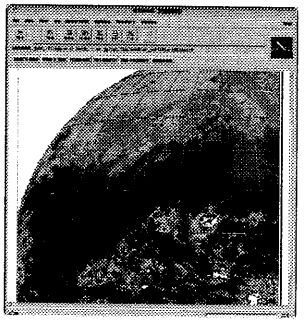

図 7: 1996 年 1 月 15 日 6 時のアジア地域の元画像

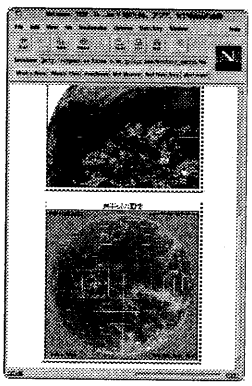

図 5: 1996 年 1 月 15 日 6 時の 3 地域の複数画像、下部

図 8: 1996 年 1 月 15 日 6 時の東半球地域の元画像

解し、手直しすることで解決できると思われる。また、実際の検索方法での問題点として、以下のような問題がある。

アクションによって、ビューを作成する場合に、テーブル asia と globe のどちらかの画像が存在しない場合に、ビューの同年月日、同時刻タップル間で、不都合が生じて結合条件がうまく働かずに、画像が検索されない。

この問題に関しては、テーブル内にヌルデータを格納しておくことで対処できる。

また、実体化ビューを用いることにより、更に高速処理をすることが可能になる。仮想ビューよりも記憶領域を必要とするが、ユーザのイベントに対してダイレクトに反応できるからである。従って、実体化ビューを実装することが、今後の課題の一つといえる。

こうして、仮想ビューを用いた検索によって、検索発行回数を減らすことができ、通信量現象に伴うスループットの向上が期待できる。したがって、ルール機能の利用による拡張性が将来的に見て、有用性があると考えられる。

5 おわりに

近年、リモートセンシング画像のような、日々刻々と増加し続ける大量の自然科学分野の画像データが注目されているが、これを効果的に利用し、管理する技術が求められている。また、WWW サーバは、情報検索をグローバルな環境で行なうシステムとして開発されてきたが、そのファイル管理方法が静的であったので、拡張性を見いだせずにいた。従って、これらをうまく管理するための DBMS が求められており、本研究では数々の拡張機能を備えた拡張 RDBMS である POSTGRES95 と、画像データに気象画像を用いて、RDB の拡張機能によるアクティブデータベースと連携した WWW サーバ・画像データベースを実現した。

今後の課題として、仮想ビューを作成する際に、ユーザによって作成するビューを変えて、よりアクティブなデータベースを構築すること、さらに実体化ビューを実装し、高速処理を可能にすることなどが挙げられる。

このような課題を実現することで、データーベースシステム全体の奥行きを増すようにしていきたい。

謝辞

本研究を行なうにあたって、気象画像の提供に力を貸してくださった菊地時夫助教授に感謝します。なお、本論文は、立松浩一氏（現在 NTT 勤務）の修士論文を発展させたものである。

参考文献

[1] Jennifer Widom , Stefano Ceri , "ACTIVE DATABASE SYSTEMS" , Morgan Kaufmann Publishers , Inc.

[2] Michael Stonebraker , Anant Jhingran et.al , "ON RULES , PROCEDURES , CACHING AND VIEWS IN DATA BASE SYSTEMS" , SIGMOD Conf.90

[3] Andrew Yu and Jolly Chen , "The POSTGRES95 User Manual (Sep 5.1995) " , Computer Science Div.,Dept.of EECS , Univercity of California at Berkeley

[4] Michael Stonebraker , Michael Olson , Large Object Support in POSTGRES , Dept.of Electrical Engineering and Computer Science Univercity of California at Barkeley

[5] 立松浩一 , 小西修 , 分散環境下における拡張リレーショナルデータベースの設計・構築 , Memoirs of Faculty of Science Kochi Univercity (Series F Information Science) Vol.17 1996

[6] 益岡竜介、木庭袋圭祐、World-Wide Web (WWW) , 情報処理学会誌 Vol.36 , No.12 , pp.1155-1165 , Dec 1995

[7] 久保田和己 , 石川博 , 富士通研究所 , データー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 , 情報処理学会データーベースシステム研究会 , 104-34 , 7.20.1995