

MineS(Multimedia Information Network Environment Services)における ユーザインタフェースの実現方式

南野謙一 布川 博士 宮崎 正俊

岩手県立大学 ソフトウェア情報学部

〒 020-0193 岩手県滝沢村滝沢字巢子 152-52

e-mail:{minamino, nunokawa, miyazaki}@soft.iwate-pu.ac.jp

あらまし コンピュータネットワークの進展に伴い、ネットワーク上に存在する様々なデータベースにアクセスすることが可能になってきた。このようなデータベースを統合利用することができれば、多彩なデータを管理する仮想的な巨大データベースとして活用することができる。しかしながら、スキーマ設計の異なる独立なデータベースを統合利用する統一的方法は確立していない。本稿では、問い合わせの結果、複数のデータベースから得られるデータを表示するユーザインタフェースについての分析を行い、アーキテクチャを示す。そしてそれに基づいて実装した MineS(Multimedia Information Network Environment Services)のユーザインタフェースについて述べる。

キーワード ユーザインタフェース, 情報環境, メディア, マルチデータベース

UserInterface Design of MineS(Multimedia Information Network Environment Services)

Kenichi Minamino, Hiroshi Nunokawa, and Masatoshi Miyazaki

Faculty of Software and Information Science, Iwate Prefectural University

152-42, Sugo, Takizawa, Iwate, 020-0193, Japan

e-mail:{minamino, nunokawa, miyazaki}@soft.iwate-pu.ac.jp

Abstract This paper presents a new approach to use computer networks as sources of information. This approach includes an strategy for supplying information, as well as a new method for the user to manipulate that information through a different way of thinking about the media. In future computer networks these items should be integrated, so that they will be seen as an effective new electronic media with an unified media environment. In this work we explain UserInterface Design of MineS (Multimedia Information Network Environment Service). MineS is a system designed to provide information for individuals through the plural media, independent of an specific physical network.

Key words userinterface, information environment, media, multi database

1. まえがき

コンピュータネットワークの進展に伴い、ネットワーク上に存在する様々なデータベースにアクセスすることが可能になってきた。このようなデータベースを統合利用することができれば、多彩なデータを管理する仮想的な巨大データベースとして活用することができる。異種情報源統合利用[1]や時間的關係、空間的關係を定義しマルチデータベースのデータ統合利用[2]を行う研究等が行われているが、スキーマ設計の異なる独立なデータベースを統合利用する統一的方法は確立していない。

本研究では、スキーマ設計の異なる独立なデータベースに対する統合利用の実現を目的とする。コンピュータネットワーク上に存在するマルチメディアデータに対してエンドユーザが単一のデータベースを利用するように統合データベースを利用できる情報環境の実現を目指している。

本稿では、複数のデータベースから得られる検索結果データを表示するためのユーザインタフェースの実現方法を提案する。そのためにまずユーザインタフェースについての分析を行い、データベースの統合利用を可能にするユーザインタフェースアーキテクチャを示す。そしてそれに基づき開発した MineS (Multimedia Information Network Environment Services) のユーザインタフェースの実現方法について述べる。

本稿の構成は以下のとおりである。2章では、検索結果表示プロセスの分析について述べ、3章で2章で述べた分析に基づくユーザインタフェースアーキテクチャについて述べる。4章では MineS の実装について述べる。5章は本稿のまとめである。

2 検索結果表示プロセスの分析

2.1 単一のデータベースに対する問い合わせ

図1に示すように単一のデータベースに対する検索結果の表示プロセスは、問い合わせの結果得られるデータをデータベース(DB)のユーザインタフェース(UI)を介して表示することにより行われる。

UIにおける検索結果の表示プロセスは、図2に示すように、表示機能により検索結果のデータフォーマットを定め、ユーザのモニタ画面上にそのデータを

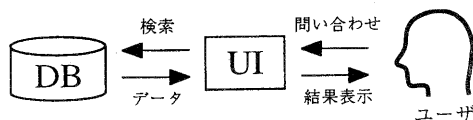


図1 検索結果表示プロセス

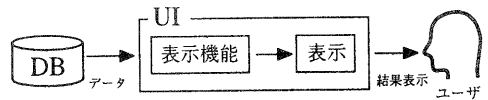


図2 UIにおける検索結果表示プロセス

表示することにより行われる。データベースアプリケーションの開発者は、この表示機能を実装する必要がある。現在、利用可能なデータベースアプリケーションにおいて実装されている表示機能は、検索結果データの件数、表示の種類により次の4つに分類することができる。ここで、1件分の検索結果データを R (Result)、表示機能を DF(Display Function)、表示を D (Display) と表記する。

(1) 1件の検索結果対1種類の表示

図3に示すように検索結果データを1件だけ受け取り、1種類のデータフォーマットによる表示機能を実現している。



図3 1件の検索結果対1種類の表示

(2) 1件の検索結果対数種類の表示

図4に示すように検索結果データを1件だけ受け取り、数種類のデータフォーマットによる表示機能を実現している。表示(D₁, D₂, ..., D_n)は、検索結果に含まれる属性値を様々な組み合わせで表示するために使用する。

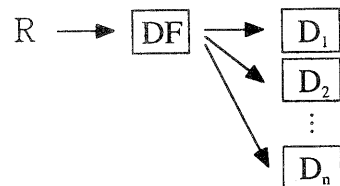


図4 1件の検索結果対数種類の表示

(3) 複数件の検索結果対1種類の表示

図5に示すように問い合わせの結果得られた数件のデータ(R₁, R₂, ..., R_m)を同時に受け取り、1種類のデータフォーマットによる表示機能を実現している。表示Dは、検索結果の一覧表示に使用する。

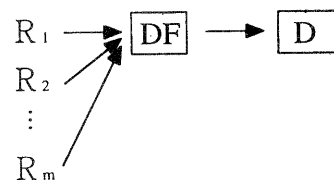


図5 複数件の検索結果対1種類の表示

(4) 複数件の検索結果対数種類の表示

図6に示すように(3)と(4)を組み合わせた場合である。問い合わせに対する検索結果データを様々な視点から表示することが可能であり、ユーザにとって利便性の高い表示機能であると言える。本研究では、この表示機能を統合データベースにおいて実現することを目的とする。

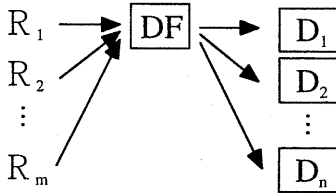


図6 数件の検索結果対数種類の表示

2.2 スキーマ設計の異なるデータベースに対する統合的な問い合わせ

複数のスキーマ設計が異なるデータベースへ問い合わせを行う場合に、図6と同等の表示機能を実現するには、図7に示すような表示機能が必要である。

図7の表示機能は、スキーマ設計の異なるデータベース(DB₁, DB₂, ..., DB_n)から、検索結果データ(R₁, R₂, ..., R_n)を受け取る。表示(D₁, D₂, ..., D_n)は、複数のデータベースから得られた検索結果に含まれる属性値を様々な組み合わせで表示するために使用する。

統合データベースには、データベース(DB₁, DB₂, ..., DB_n)に格納されているレコードオカレンスの属性値に関する関連データ(同一性, 類似性, 包含性等を表すデータ(メタデータ))が管理されている。このメタデータを利用することにより、スキーマ設計の異なる

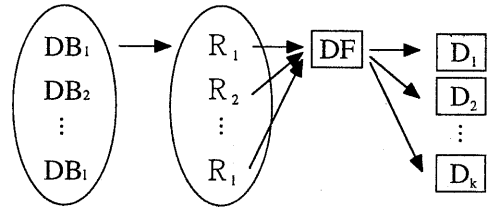


図7 統合データベースにおける検索結果表示

データベース(DB₁, DB₂, ..., DB_n)に対して関連を発見し、問い合わせの際のデータ操作を行うことが可能になる。検索結果データは、メタデータにより関連づけられた複数のデータベースの属性値の集合として得られたものとなる。メタデータは、属性値間の関連を意味的ルールとそれに対応し画面上に配置させるための空間的ルールから定義される。本研究では、このようなメタデータの作成を前提とし、スキーマ設計が異なるデータベースの統合利用を実現する。本稿では、メタデータの構造, 生成方法については触れない。

3 アーキテクチャ

2章で述べた分析により、ユーザにとって利便性の高いUIを実現するためには、複数件の検索結果に対して数種類の表示を用意する必要がある。ただし、複数のデータベースに対する問い合わせにより得られる検索結果データの数は非常に多く、開発者がすべてのユーザの好みに合う表示を作成することは困難である。このため検索結果データを表示する基本となるデータフォーマットをいくつか用意し、ユーザが検索した属性と属性値およびメディアタイプから自動生成

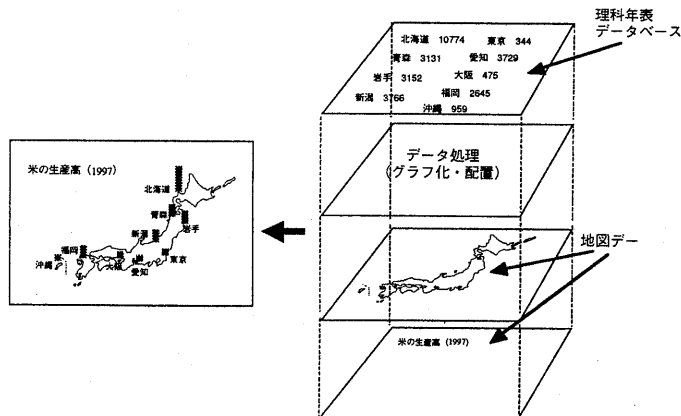


図8 データフォーマット例

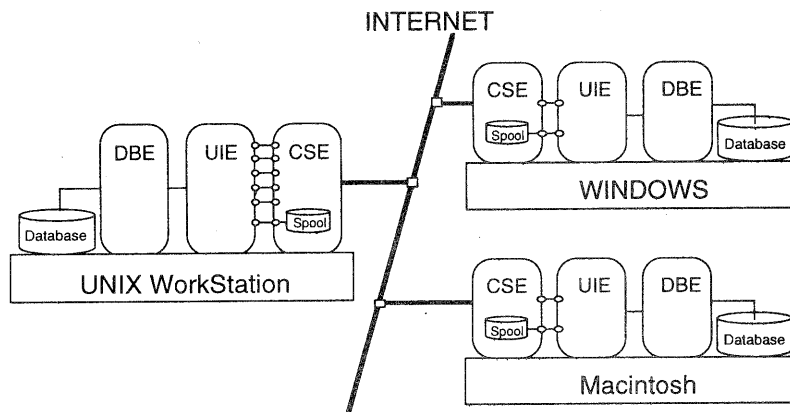


図10 MineSのシステム構成，及びその実装

することにする。そして個々のユーザに対応するため、検索結果を見ながらマウスを用いて検索結果データの表現形式、配置をカスタマイズすることができるようにする。

基本となるデータフォーマットをデータベースにより管理し、必要時に取得、実行できるように言語を用いてデータフォーマットを定義する。データフォーマットには、検索結果データを取得する問い合わせの記述や検索結果データの表現形式、表示位置の記述がある。ユーザのカスタマイズはこれらの値を書き換えることにより行われる。このデータフォーマット記述は動的に表示できるようにインタプリタにより実行する。インタプリタはデータフォーマット記述の実行だけでなく、データベースに対するデータ操作言語も実行する。このため、問い合わせをユーザが行い、検索結果がインタプリタに返って来た時には、検索結果データの属性、属性値およびメディアタイプを判別し、必要なデータフォーマット記述を選択、実行する。データフォーマット記述に適切なものがない場合に

は、メタデータを参照し属性値間の関連を変更し再定義を行い実行する。

例えば、ユーザが理科年表データベースに対して各地域の米の生産高のデータを取得し、地図データベースから日本地図を取得した場合には、図8に示すように検索結果の属性値の関連ルールにより、日本地図上にそれらのデータをプロットするフォーマット記述をインタプリタが実行する。このようなルールはユーザが問い合わせやカスタマイズを行う時に随時追加し、ユーザの好みに適応する。図9にインタプリタ内部のデータの流れを示す。

4 マルチメディアインフォメーションサーバ MineS[3]

MineS(Multimedia Information Network Environment Services system)は、多様なネットワーク・端末上でマルチメディア情報を利用することのできる、分散協調型のデータベース検索システムである[3]。MineSの利用者はデータベースを利用したコンピュータネットワーク環境において情報を発信し、収集することができる。MineSのシステム構成、およびその実装を図10に示す。MineSシステムは利用者とのインタフェースであるユーザインタフェースエンジン(UIE)、通信機能を司るコミュニケーションサポートエンジン(CSE)、およびデータベースに対する処理を行うデータベースエンジン(DBE)により構成される。UIEは利用目的に応じた使い易いUIを実現するためにMineS/UI記述言語を提供し、プログラマブルなUIカスタマイズ環境を提供する。MineS/UI記述言語の一部を図11に示す。

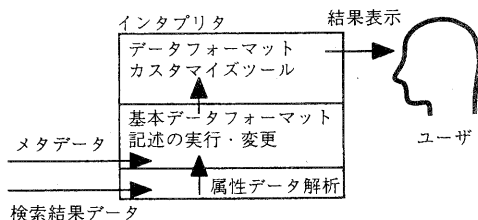


図9 インタプリタ内部のデータの流れ

関数	機能
(make-format <name> <description>)	<name>で指定された名のデータフォーマットを<description>により定義する。
(get-root-window-ID)	Window SystemにおけるルートウィンドウのIDを取得する。
(create-window <window-ID> <position-x> <position-y> <width> <height>)	<window-ID>で指定されたウィンドウ上の<position-x> <position-y>で指定された位置に<width> <height>で指定された大きさのウィンドウを生成する。
(get-data <database> <sql>)	<database>で指定されたデータベースに対して、<sql>で指定された問い合わせを実行する。
(show-picture <window-ID> <position-x> <position-y> <width> <height>)	<window-ID>で指定されたウィンドウ上の<position-x> <position-y>で指定された位置に<width> <height>で指定される大きさの静止画を表示する。
(send <port> <Media-expression>)	CSEにより提供される<port>で指定されたプロセス間通信のためチャンネルへ、<Media-expression>で指定されたメディア記述を送信する。<port>にはリアルタイム型ポート、蓄積型ポートがある。

図 11 MineS/UI 記述言語の一部

MineS/UI記述言語は、データベースに対する処理を行う関数、各種通信プロトコルに依存することなくデータ転送可能な関数、および各メディアデータに依存することなく表示可能な関数を提供する。UI記述の構造はLispと同様であり、高階関数を含む関数合成、再帰呼出しにより記述される。データベースアクセス、データ転送、GUI等の関数ライブラリが揃っており、比較的容易にUIカスタマイズが可能である。MineS/UI記述言語によるUI記述はインタプリタにより解釈、実行するため、プログラムをデータと同様に扱え、UIE間での授受による共有が可能となる。MineSの利用者は情報発信の点からも、情報収集の点からもUIを自由にカスタマイズすることが可能である。

データフォーマットはMineS/UI記述言語により記述され、UIEにより実行される。基本となるデータフォーマット記述、メタデータは、データベースに格納される。メタデータもMineS/UI記述言語により記述される。図12にデータフォーマット記述の実行画面を、図13にデータフォーマット記述を示す。この例では、気象データベースと地図データベースに対して問い合わせを行い、仙台と沖縄の気温の変化を表示したものである。データフォーマット記述には、メディアデータをボタンにより操作する記述がされている。

5. むすび

本稿では、複数のデータベースから得られる検索結果データを表示するためのユーザインタフェースの実現方法を提案した。そのためにまずユーザインタフェースについての分析を行い、データベースの統合利用を可能にするユーザインタフェースアーキテクチャを示した。そしてそれに基づき開発したMineS (Multimedia Information Network Environment Services) のユーザインタフェースの実現方法について述べた。

今後の課題としては、MineS/UI記述言語を機能拡張し、個々のユーザの利用目的に対応することのできる新しいメディアに対応できるようにすること、個々のデータベースのスキーマ設計を意識せずに統合利用ができるシステムを実現することが挙げられる。

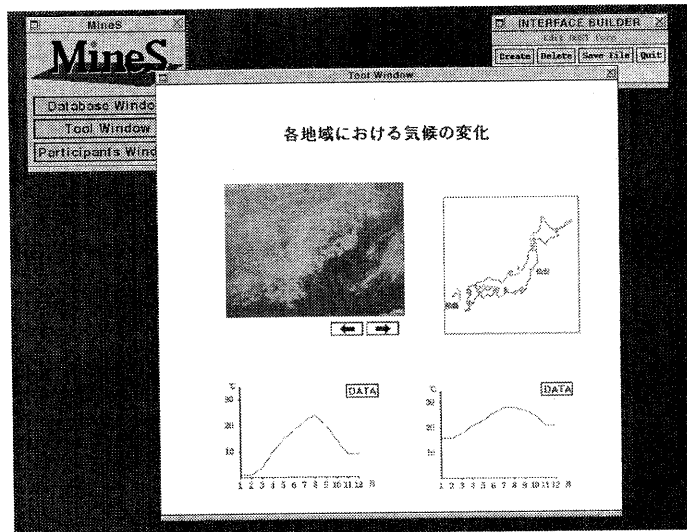


図 12 データフォーマット記述の実行画面

参考文献

- [1] 森嶋厚行, 北川博之: 視覚的操作系による異種情報源統合利用支援, 信学会論文誌, D-1, Vol. J82-D-1, No.1, pp.315-326, [1999].
- [2] 細川宜秀, 石橋直樹, 八代夕紀子, 清木康: マルチデータベース環境における時間的關係, および, 空間的關係に関する解釈を伴うデータ統合方式の実現, 情報処理学会研究会報告 99-DBS-119, Vol.99, No.61, pp.387-392, [1999].
- [4] 南野謙一, 布川博士, 水野晋一, 宮崎正俊, 岩本正敏: マルチメディアインフォメーションサーバ MineS のユーザインタフェース機構, 情報処理学会研究会報告 94-DBS-99, Vol.94, No.62, pp.193-200, (1994).

```
(make-format MySlide
  (progn (setq MyWindowID (create-window
    (get-root-window-id) 0 0 400 300
    (show-text MyWindow "My Analysis"
      "gothic" 18 40 20 80 15 0)
      .... ))
    (make-graph MyWindowID (get-data TemperatureDB
      (and (look-up "place" "Tokyo")
        (look-up "year" "1998")))) 20 300 100 100 1 1)

    (set-script "DATA" 70 230 150 30 2)
      (get-data TemperatureDB
        (and (look-up "place" "Tokyo")
          (look-up "year" "1998"))) 20 300 100 100 1 1)
    .... )
```

図 13 データフォーマット記述例