

5E-8

# データベースの複合ビューとしての 対話マルチメディアオブジェクトの構成 / 管理方法

有川正俊 河北秀世 上林彌彦

京都大学工学部

## 1 はじめに

従来のデータベースでは、ビューもまたデータベースであるという考えを基本としている<sup>[1]</sup>。

本稿では、実際に画面上に表示されたオブジェクト(表示オブジェクト(DO)と呼ぶ)をビューとして捉え、このビューとデータベースビューとの関係を明らかにする。データベース内のオブジェクトは入出力に依存しない形式で蓄えられ、利用する際には、利用者インタフェースを構成する対話可能な表示オブジェクトとして実現しなければならない。つまり、利用者インタフェースは、データベースのビューであり、必要に応じて動的に対応する表示オブジェクトを生成するための計算記述として定義する。ビュー定義としての計算記述により、動的に生成される表示オブジェクトを仮想ノードとし、ビュー定義を差分的に変更する計算記述により起動される仮想ノード間の遷移を仮想リンクと考えると、ビューを基本とする利用者インタフェースは仮想ハイパーメディアと見なせる。

## 2 ビューを構成要素とするハイパーメディア

本稿では、ハイパーメディアは、データベースビューであり、利用者の要求するデータおよび、それを利用者が理解もしくは操作するための環境と定義する。

### 2.1 概念オブジェクト・表示オブジェクト

概念オブジェクト(Conceptual Object:CO)は、データベース中のデータの構成単位(atomic data)であり、利用者インタフェースを持たず、多目的利用の形式のオブジェクトである。

表示オブジェクト(Display Object:DO)は、特定のCOに対する利用者インタフェースを表しており、対応するCOの表示と対話のためのデータおよびメソッドを持つ。

SmalltalkのModel-View-Controller(MVC)アーキテクチャに基づけば、COがmodel、DOがview-controller pairに相当する<sup>[2]</sup>。こうした立場でハイパーメディアの構成オブジェクトを眺めると、これら構成オブジェクトは対応するCO(COの集合)と、それに対応するDO(DOの集合)を保持しており、それ自体がMVCアーキテクチャに基づいた、GUIの機能部品となっている。

ハイパーメディアの一つのノードは、COの集合、COから対応するDOを生成する視覚化手続きのオブジェクト、そして表示結果を表すDOの集合から成ると考える。

### 2.2 ビューを基本としたハイパーメディアの枠組みの適用対象としての地理応用

コンピュータを利用した対話機能を持つ地図は、数値・文字データを1枚の複合機能図形として視覚化した対話マルチメディアオブジェクトである。地図応用は、データベースビューとMVCの対話モデルのビューとの関係を明らかにさせる適切な研究題材と考えている。地理応用では、地図が利用者インタフェースとしての表示オブジェクトであり、利用目的に合う視覚化手続きおよび対話操作手続きを構成メソッドとして持つ。一方、多目的利用を仮定した概念オブジェクトは、地理データベースに格納された生の地理データである。地図オブジェクトは、複数の地理オブジェクトを1つの代理オブジェクト(Proxy Object)<sup>[3]</sup>として取り扱ったものと考えられることができる。

以下では、開発中のプロトタイプシステムGeoProxyのビュー構成/管理方法を紹介することにより、具体的なデータベースの複合ビューとしての対話マルチメディアオブジェクトの構成/管理方法を議論する。

## 3 地理応用におけるビュー構成

### 3.1 表示上の制約を考慮した利用者ビュー定義支援環境

物理的に限られた画面サイズのウィンドウ中に、問い合わせの結果であるCOの集合全てを利用者が意図する図形化手続きでDOの集合として実現できない場合がよく起こる。たとえば、1000個の建物の情報をその位置と名前を図形化して10cm×10cmのウィンドウの中に表示することは困難である。そこで、表示すべきCOのクラスごとに重要度を割り当て、重要度が高いほど利用者の意図通りに図形化を行うという利用者の意図の自動的緩和機構は有用である。

図2では、プロトタイプシステムGeoProxy上でのこの緩和機構のための視覚的インタフェースの例である。右下の“Importances”ウィンドウでCOのクラスごとに重要度を設定でき、その上にある“V\_Rank Function”ウィンドウで、重要度に対する図形化手法のレベルの設定を行うことができる。

このように、地図の利用者インタフェースは、データベースのオブジェクト(CO)から導出されたDOだけではなく、地図の図形化手法を利用者が対話的に決定するための利用者インタフェース独自の局所的オブジェクトから導出したDOも構成要素となる。

プロトタイプシステムGeoProxyでは、COの集合であるデータベースビューおよび地図の特徴量を決定するためのデータとメソッドを含む利用者インタフェース・オブジェクトを定義するクラスを“GeoProxy”と呼ぶ。

クラスGeoProxyは、データベースビューと図形化手

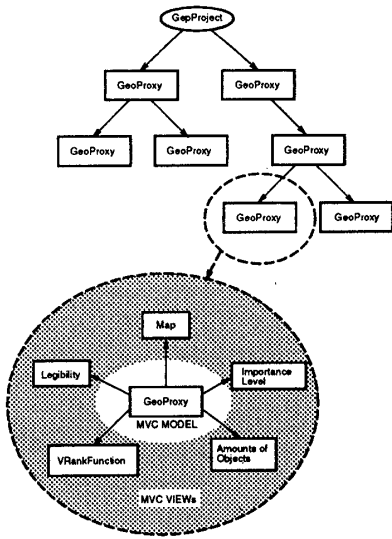


図 1: データベースビューとしての GUI オブジェクト

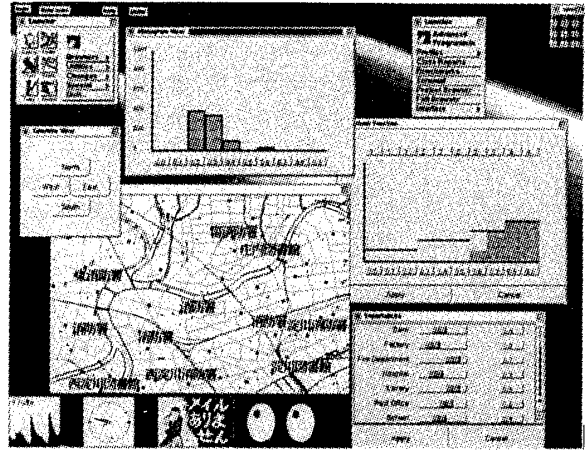


図 2: 図形化のための種々のビュー

法との2つの定義から構成されている。図形化手法もオブジェクトとしてデータベースに格納され共有されている。

MVCアーキテクチャにおいて、クラス GeoProxy のインスタンス（以下、このクラスのインスタンスを単に GeoProxy と呼ぶ）が model に対応し、地図、重要度設定、図形化手法の設定などのためのそれぞれのウィンドウが view-controller pair に対応している（図 1）。

### 3.2 ビューの派生と依存関係

図 1 の GeoProxy の木の根に在る “GeoProject” は、純粹に問い合わせにより検索された CO の集合である。この GeoProject を元に、利用者は、目的ごとにいろいろな図形化手法を施し、複数の種々の地図をビューとして生成し、1つの利用者環境を生成する。

図 1 の GeoProxy の木の親子関係の意味は、ある GeoProxy から表示領域、重要度、図形化手法を部分的に変更した GeoProxy を派生した結果を図示したものである。実際の表示の例が図 3 である。図 3 では、表示領域の違いより、広い範囲の地図は重要度の高い CO しか表示されてなく、狭い範囲の地図は重要度の低い CO も表示した詳細な表示を実現している。図 3 の上部に在る “ROOT MAP” ウィンドウの中の 2 つの長方形 (DO) は、それぞれ派生された下の 2 つの地図のウィンドウに対応する GeoProxy の “代表 DO” である。小さい白い長方形は、地球上の座標上で定義されたものであり、透明の大きな長方形は、ウィンドウ上の座標で定義されたものである。“ROOT MAP” ウィンドウをスクロールした場合に、小さい白い長方形は、一緒にスクロールするが、透明の大きな長方形は移動しない。また、小さい白い長方形を移動 / 変形すると、それに応じて、対応する地図の表示が変わる。逆に、派生された地図上でスクロールや拡大 / 縮小すると、対応する白い長方形 (代表 DO) が移動 / 変形する。

図 1 の GeoProxy の木は、派生関係だけを表現しており、各 GeoProxy の表示領域の包含関係は、前述のように代表 DO により視覚的に分かる。この代表 DO は、分散環境で

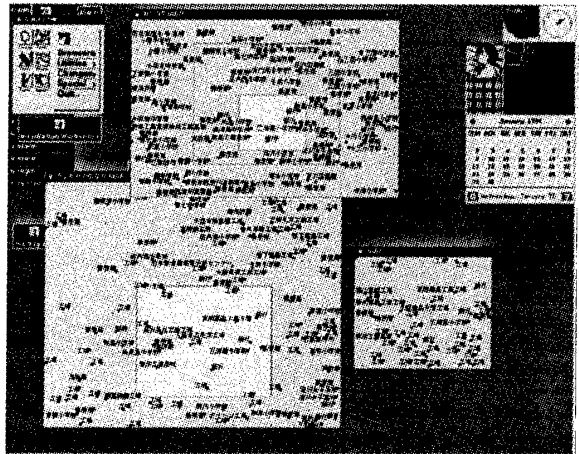


図 3: 派生したビューとそれらの依存関係

の GeoProject の共有を考えた場合、他の利用者がどこを見ているのか、あるいはどの部分の地理データを変更 / 追加しているのかとう共同作業の空間的な並行制御の処理単位としても理解することができる。

このように、GeoProject は、共有オブジェクト空間を意味し、共同作業におけるオブジェクトの変更 / 追加の伝播の範囲を規定することになる。一方、GeoProxy 独自の構成要素である重要度や図形化手法の設定の伝播（継承）の定義は、GeoProxy 間の派生関係を表す辺で行われる。

### 参考文献

- [1] S. Abiteboul and A. Bonner : Objects and Views, *Proc. of ACM SIGMOD Conf.*, pp.238-247, 1991.
- [2] ObjectWorks\Smalltalk Users Manual, PracPlace Systems, Inc., 1992.
- [3] D. Maier, J. B. Cushing : Treating Programs as Objects: The Computational Proxy Experience, *Proc. of Third Int'l Conf., Deductive and Object-Oriented Databases.*, pp.1-12, 1993.