

7R-6

アプリケーション開発環境 MADE と
アプリケーションインタフェース層について

山本 貴義 田淵 仁浩 星野 准一 村岡 洋一

早稲田大学 理工学部

1. はじめに

本稿では、データモデル MeSOD に基づくアプリケーション開発環境 MADE(MeSOD Application Developer's Environment)の基本思想とその実装について述べる。データモデル MeSOD は、実世界の实体を距離空間内の点として捉えるデータモデルである。MeSOD では、実体間の関連度を距離値によって表すので、類似性や曖昧性を含んだ質問を使った検索が可能である。この MeSOD に基づくアプリケーションの1つが Hyperbook システム [3] である。Hyperbook システムは、自己完結型のマルチメディアデータベースシステムであり、「本」のように自由に「読んだり、書いたり」できるシステムとして設計されている。

MADE は、この Hyperbook システムをターゲットアプリケーションとした開発環境である。以下では、MADE の基本的な設計思想について述べる。まず、Hyperbook システムを概観した上で、MADE に対する要件について述べる。その要件に対する対策として、MADE の3層構造について議論し、続いてアプリケーション開発者のインタフェース層について述べる。残りの1つの層については、[1] で述べる。DBMS に対応する層については、[2] に詳しい。

2. MADE

2.1 MADE に対する要件

Hyperbook システムを構築するための開発環境である MADE に対する要件は、以下の3つである。

- (1) さまざまな Hyperbook システムに共通なユーザーインタフェースを容易に構築できること、
- (2) MADE が支援しているさまざまな機能を容易、かつ効果的に利用できること、
- (3) データを維持、管理するための機能を具備していること。

(1) は、分かりやすく、親しみやすいインタフェースを提供することを実現するための要件であり、アプリケーション間のユーザーインタフェースの共通性を要件としている。この共通性は、ユーザーがシステムに慣れ親しむまでの労力を軽減するうえで重要な要素である。(2) は、MADE とアプリケーション開発者とのインタフェースに関する要件である。MADE は実装している機能をアプリケーション開発者が包括的に利用できるようにしなくてはならない。(3) は、Hyperbook システムが、自己完結型のマルチメディアデータベースシステムとして構築されるための要件である。この要件は、データモデル MeSOD に基づくデータベース管理システム(DBMS)を容易にアプリケーションに組み込めることを意味している。

以上の要件を満たすために、MADE はユーザーインタフェー

MADE - an Application Developer's Environment -
and its Application-Interface Layer.

Takayoshi YAMAMOTO, Masahiro TABUCHI,
Jun-ichi HOSHINO and Yoichi MURAOKA.
Waseda University.

ス層、アプリケーションインタフェース層、MeSOD 層の3つの層から構成されており、その結果、きわめて汎用性の高いソフトウェアアーキテクチャを実現している。

2.2 MADE の構成

MADE は、図1のような構成をしている。以下では、MADE の各層の役割について概観する。

(1) ユーザーインタフェース層

ユーザーインタフェース層は、距離に基づく実世界の表現と、システムとユーザーとの対話を実現するための様々な機能を持っている。この層の機能を利用することにより、MeSOD に基づくアプリケーションのユーザーインタフェースを容易に構築できる。

(2) アプリケーションインタフェース層

アプリケーションインタフェース層は、MADE のユーザー、すなわち開発者に対するインタフェースである。開発者は、この層の機能を利用することにより MeSOD 層とユーザーインタフェース層の両方の機能を対応付けることができる。

(3) MeSOD 層

MeSOD 層は、MeSOD に基づく DBMS に対応する。この層は、「距離空間」の定義、変更、操作などの機能を持っている。これらの機能は、アプリケーションインタフェース層によって利用される。

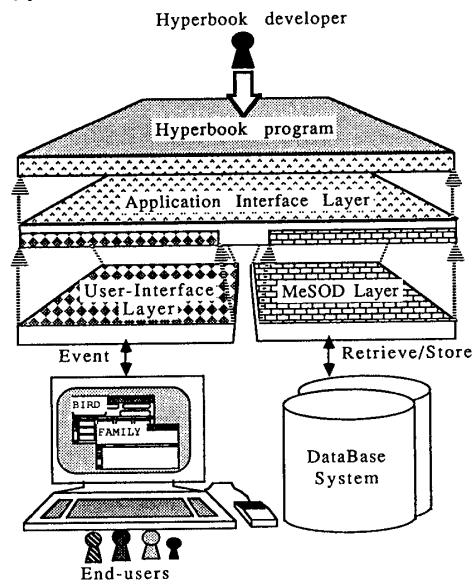


図1:MADE の構成図

3. アプリケーションインタフェース層

アプリケーションインタフェース層 (APP 層) とは、MADE において、アプリケーションの実装者が直接に接するプログラミングインタフェースである。

2.1 で述べたように、この層の機能を利用することにより、MeSOD 層に用意されたデータベースの機能と、ユーザーインタフェー

ス層 (UI 層) に用意された機能を組み合わせたアプリケーションを容易に構築できなくてはならない。しかし、他の 2 層の機能を従来の言語のみを使って対応付けるには実装者が各層の仕様を熟知する必要があり、かつアプリケーションプログラムが煩雑になる。そのため、アプリケーション実装者が他の 2 層の機能を直接使わずにアプリケーションを構築できるユーティリティを APP 層に用意しなくてはならない。

以上の要件を満たすため、APP 層にマッピングオブジェクトという概念を導入した。マッピングオブジェクトは、プログラミングオブジェクトであり、他の 2 層の機能とデータ構造を結合する手段とアプリケーション構成のための機能を持つ。

3.1 アプリケーションインタフェース層の要件

従来の言語を使って 2 つの層の機能を利用したアプリケーションを構築する場合、実装者はアプリケーションの実世界での振舞いの記述に専念できない。

例えば、APP 層で 1 つのオブジェクトを削除することを考える。その場合に、データベースからそのオブジェクトを削除し、さらに画面上との一貫性を保つため、表示を更新しなければならない。このため、実装者は MeSOD 層との通信プロトコルや、UI 層とのデータのやりとりについての知識を持つ必要がある。

実装者がこのようなことに意識を傾倒せずに、実世界の振舞いの記述に専念できるようにするために、他の 2 層の機能を包括的に利用できる仕組みを APP 層に実現する。このような機能を提供するためには以下の 2 点が必要となる。

- (1) MeSOD 層と UI 層との対応を管理すること。
- (2) これらの 2 層の機能を合わせ持った機能を用意すること。

(1) は、例えば、鳥類を Hyperbook 上で表現する場合に、データベース上にある鳥のデータと画面上の視覚オブジェクト [1] との対応を管理することである。

また (2) は、(1) のように対応付けられた鳥のデータと視覚オブジェクトに対する操作を統合することである。

3.2 マッピングオブジェクト

上に述べた要件を満たすため、APP 層にマッピングオブジェクトなる概念を導入する。マッピングオブジェクトとは、APP 層でのプログラミングオブジェクトである。マッピングオブジェクトは、(1) 私有データとして MeSOD 層と UI 層におけるプログラミングオブジェクトを保持し、(2) プログラミングインタフェースとして 2 層の機能を結合したメッセージを持つ。

このようなマッピングオブジェクトを用意することにより、MeSOD 層と UI 層とを対応付けることができる。一方、2 層の機能を結合したメッセージを多数用意することにより、実装者は他の 2 層を利用するための仕様を知らなくてもこの層で用意された機能を使いアプリケーションを構築できる。このため、実装者は、各層の機能の結合をアプリケーションごとに記述する必要がなくなる。

例えば、距離の概念を使った検索を実現する場合、以下のような手順となる。まず画面上に入力された参照値を UI 層から受け取らなければならない。次に MeSOD 層に対し、参照値に似ているものをあらかじめ定められた距離関数を使って検索するように命令する。それから検索された結果を MeSOD 層から受け取り、最後に、UI 層にその結果の表示を命令することが必要となる。

このような動作を APP 層を使わずに記述するのは、アプリケー

ション実装者にとって以下の 2 点から煩わしい作業となる。

- (1) 実装者は UI 層でのオブジェクトと、データベース上のデータの対応をプログラムを書くことによって管理しなければならない。
- (2) 実装者は、MeSOD 層や UI 層とのデータの受渡しを定められた作法に従ってプログラミングする必要がある。

APP 層に用意されたマッピングオブジェクトを使うと以上の問題点が次のように解決される。

(1) 実装者はマッピングオブジェクトに、データベース内のオブジェクトと UI 層内の視覚オブジェクトを割り当てることによって、2 つの層の間での対応をとることができる (図 2-1,2)。このため、UI 層から受け取った参照値をデータベースのどのデータに対して適用するのか、また MeSOD 層での検索結果をどの視覚オブジェクトに送るのかといった点についてはマッピングオブジェクトにまかせればよい。

(2) 実装者は、APP 層に用意されているマッピングオブジェクトの基本的なメッセージ群を利用することにより、各層とのデータの受渡しの作法等をプログラミングしなくてもアプリケーションの構築ができる (図 2-3)。例えば、マッピングオブジェクトは検索結果の受渡しのために必要な構造体を内部変数として持ち、それを利用して各層と通信し合うメッセージを備えている。

その結果、アプリケーションの構築の手間が大幅に軽減される。

図 2 は、現在作成中の鳥についての Hyperbook のプログラムコードの一部である。

```
[aBird setProperty:5,
  "Name", "Size", "Habitat", "Family", "Order"];
[[aBird getProp:"Size"]
  setProperty:2, "SizeBody", "SizeWing"];
aSizeBody = [aBird getProp:"SizeBody"];
  (1) マッピングオブジェクトの定義

[aSizeBody setReference:[Scale setScaleWindow] ];
[aSizeBody atRef:"setFunc:f:"
  with:ALL with:browseByScale];
[aSizeBody setPresentation:
  [UsrI setCanvas:CARD base:[aFamily getRefFrame] ]];
  (2) マッピングオブジェクトへの各層に対する対応付け

[aSizeBody browseNdisplay];
  (3) 検索要求
```

図2: マッピングオブジェクトの利用例

4. おわりに

本稿では、MADE 及びそのプログラミングインタフェースであるアプリケーションインタフェース層について述べた。

MADE を利用することにより、優れたデータベースアプリケーションを構築することができ、またアプリケーションインタフェース層を利用することによりそのようなアプリケーションが容易に構築できる。

【参考文献】

- [1] 星野, 田淵, 山本, 村岡: 「アプリケーション開発環境 MADE における UI (User Interface) 層の設計概念」, 情処 38 回全大に発表予定, 1989.
- [2] 田淵, 村岡: "距離空間データモデル MeSOD に基づいたオブジェクト指向データベース管理システムの設計", 情処研報 Vol.88, No33, 88-DBS-65-3, 1988.
- [3] 田淵, 星野, 山本, 村岡: 「距離空間データモデル MeSOD に基づく TinyHyperBook の概要」, 情処 37 回全大 4Q-3, pp.359-360, 1988.
- [4] B.J.Cox: "Object Oriented Programming - An Evolutionary Approach -", Addison-Wesley, 1986.