

# アプリケーション開発環境MADEについて

MADE:MeSOD Application Development Environment for Hyperbook

田淵 仁浩 村岡 洋一  
Masahiro TABUCHI Yoichi MURAOKA

早稲田大学 理工学部  
School of Science and Engineering, Waseda University

あらまし 本稿では、マルチメディア電子化図書システムHyperbookの開発環境MADEを提案する。MADEは具体的には、鳥類図鑑HyperbookシステムをデータモデルMeSODに基づいたアプリケーションとして構築するためのアプリケーション開発環境である。鳥類図鑑Hyperbookシステムでは、利用者の検索要求の入力や検索手順に、より多くの自由度が要求されるので、アプリケーションの機能記述に負担が大きい。MADEは、このようなHyperbook実装者に対する負担を軽減することを目的としている。MADEでは3層構造を採る一方で、Hyperbook実装者に対するインターフェースをひとつの層にまとめることによって、システム記述に専念できる環境となっている。

Abstract The purpose of this paper is to propose an application development environment (MADE) for multimedia electronic book system (Hyperbook). MADE is an application development environment to build the Birds Hyperbook system based on MeSOD model. Because the Birds Hyperbook system is required flexibility for input and retrieval by end-users, its development is much load for a Hyperbook developer. The purpose of MADE is to reduce such load. Because MADE consists of three layers and has only one programming interface (layer), MADE provides the Hyperbook developers with an excellent development environment.

## 1. はじめに

一般に、鳥類図鑑では利用者の検索要求は、図鑑に述べられている鳥の特徴や性質とは、必ずしも一致しない。例えば、鳥のシルエットを一つに特定できると、検索の対象とする鳥の集合を絞り込めるが、普通の利用者は、見た鳥のイメージからシルエットを一つに特定できない。このように、鳥類図鑑における利用者の検索要求は、図鑑に記されている特徴や性質と必ずしも一致しないイメージ情報であるという意味において、「曖昧」である。

マルチメディアデータベース応用として、現在開発中の鳥類図鑑Hyperbookシステムは、このような曖昧なイメージから鳥を突き止めるようなアプリケーションである。曖昧なイメージ情報から鳥を捜せるためには、鳥類図鑑Hyperbookシステムは、検索要求の曖昧さを吸収し、鳥の確認環境を支援できなくてはならない。そこで、我々は鳥類図鑑Hyperbookシステムの核となるマルチメディアデータベースシステムのデータモデルとして、MeSODを提案し、採用している。

MeSODは、関連度の概念を導入することによって、上述したような曖昧な検索要求を扱える。すなわち、

検索要求とデータとの関連度が近いデータの集合を取り出すことによって、検索要求の曖昧性を吸収する。例えば、鳥の鳴き真似を検索要求とした場合には、鳴き真似と鳴き声間の類似度を使って、鳴き真似で表現している鳴き声のイメージ情報を利用できる。

また、MeSODでは検索要求に対する候補として取り出されたデータの集合を利用者が容易に確認できるように、個々の鳥や鳥の属性をオブジェクトとして捉えている。つまり、個々の鳥は、図鑑に書いてあるような属性を持った自律体としてモデル化されている。したがって、鳥を確認する際には、鳥に対して「鳴け」とか「右向け」などと命令できる。

このようなHyperbookシステムでは、利用者の入力方法や検索手順に、より多くの自由度が要求されるので、ユーザインタフェースや検索環境の機能記述に負担が大きい。そこで、Hyperbook実装者の負担を軽減するために、MeSODに基づくアプリケーション開発環境MADE(MeSOD Application Development Environment)を提案する。MADEは曖昧な入力を促したり候補を確認するためのユーザインタフェースの構築や、鳥類図鑑の機能記述、鳥をオブジェクトとして扱うための記述

を容易にするためのツール群である。

本論文では、このMADEの思想と機能について述べる。本稿の構成としては、まず、鳥類図鑑Hyperbookシステムとマルチメディアデータモデルの要件について述べたあと、MeSODについて簡単に説明する。次いで、MeSODを元にしたアプリケーション開発環境MADEとその実装方法について述べる。

## 2. 鳥類図鑑Hyperbookシステム

鳥類図鑑Hyperbookシステムは、マルチメディアデータベースを核にしたマルチメディア電子化図書システム[2]である。鳥類図鑑Hyperbookシステムでは、利用者の検索要求と図鑑にあるデータとは、必ずしも一致しない。これは、利用者の検索要求が利用者個人のイメージを表現したものであるため、その意味で曖昧な検索要求である。

以下では、鳥類図鑑における鳥の特徴と利用者の検索要求について述べた後、鳥類図鑑Hyperbookシステムの要件について考察する。

### 2.1 鳥類図鑑と利用者の検索要求

一般に、市販されている鳥類図鑑から見聞きした鳥を捜すのは、難しい。利用者が見聞きした鳥に対するイメージと、図鑑にある写真や絵の鳥や、その特徴が必ずしも一致しないからである。その結果、利用者は、鳥に対するイメージを元に捜そうとしても、その確認が思うようにできない。また、利用者自身が持つイメージと図鑑の記述の違いに気付いても、検索をもう一度初めからやり直さなくてはならない。

例えば、鳥類検索図鑑[1]では、シルエット、生息環境、鳴き声、色のパターン、大きさが重要な検索項目であるが、これらの情報は、特定しにくいものばかりである。

(1)シルエット: 実際、図鑑にある写真を良く見てもらっておき、検索とは逆にシルエットを指定してもらうと、いくつものシルエットが候補に上がり、特定は困難である。

(2)生息環境: 生息環境の場合、「水辺と陸地」という情報もかなり有効だが、水を吞まない鳥はいないので、見た場所が湖の周辺であったからといって、水辺の鳥であるとは言えない。

(3)鳴き声: 鳴き声は、本の中では、表現できないので、非常に有力な手掛かりであることが分かっているが、利用する方法がない。

(4)色のパターン: 色の場合は、見たときの角度、光の当り方の違いで感じ方が違う。また、色のパターンに特徴のある鳥は、様々な色とパターンを持っている

ので、色とその位置や模様との関係が曖昧になりがちである。

(5)大きさ: 大きさは、利用者個人のイメージにある相対的な大きさでしかなく、同じ鳥を見ても大きさの表し方は、一意ではない。

このように、鳥類図鑑に対する利用者の検索要求は、曖昧になりがちなので、検索の手順や方法が固定的な従来の本から、見聞きした鳥を捜すのは困難である。

一方、上述の(1)~(5)の項目に対する利用者の検索要求は曖昧になりがちだが、的外れという訳ではない。例えば、(1)の鳥類検索図鑑を使ったシルエットの逆引きでは、複数個のシルエットの指定を許すと、かなり良い確率で、見てもらった写真の鳥が属するシルエットを含んでいる。

### 2.2 鳥類図鑑Hyperbookシステム

鳥類図鑑Hyperbookシステムの目標は、2.1で述べたような曖昧さを持つ検索要求を扱えるようなデータベースシステムである。そこで、鳥類図鑑における利用者の検索過程を大きく、検索要求の入力、システムの応答、検索結果の確認、知識の獲得(利用)の4つに分けて、Hyperbookシステムにおける要件を考察する。

#### (1)検索要求の入力

検索要求の入力では、利用者の入力があまり当てにならないことを考慮して、できるだけ利用者が望む入力方式を採れる必要がある。例えば、シルエットならば、システムが提示したものの中から選ぶ方法、利用者自身が描く方法、自然言語による方法が考えられるが、何れの方法を採ることも可能であって欲しい。鳴き声ならば、鳴き真似や印象語を使った入力ができるとう良い。大きさの場合は、相対的な大きさの指定方法として、良く知られた雀を基準としたり、マッチ箱を基準とする方法などが考えられる。

#### (2)システムの応答

利用者の検索要求は曖昧であるので、いくつかの候補が利用者に提示される。しかしながら、利用者にとっては、どのように検索要求が評価されて候補群が選ばれたのかを知ることのできる環境であって欲しい。そこで、利用者の検索要求に最も適合したものから、あまり適合しなかったものまですぐに分かるような表現を行うべきである。例えば、空間的な配置に意味を持っていて、既知の鳥がどのような評価を受けたかが直観的に分かれば、利用者はシステム側の評価を推測できる。

#### (3)検索結果の確認

利用者主体の確認環境は、曖昧な検索要求の対処と対をなして、重要である。なぜなら、鳥類図鑑で、

「候補として上がった鳥の中でどの鳥が正解か、あるいは、候補の中にはないか」を判断できるのは、利用者自身であるからである。そこで、Hyperbookシステムでは、見聞きした鳥を確認する手段を提供しなくてはならない。例えば、見たときと同じ姿勢や行動を鳥にさせられるとか、ある種の鳥の鳴き声のバリエーションを聴けると良い。この場合、候補の鳥達の一羽に対して、ある姿勢を直接指定すると全ての候補の鳥がその姿勢に変わったり、季節などの環境を変えることによって、鳴き声がさえずりになったり、地鳴きになったりすると一層良い。

#### (4)知識の獲得と利用

最終的に鳥が特定できれば、その鳥を検索するための特徴だけではなく、鳥に関する知識を得ることができなくてはならない。また、このようにして確認して得た鳥を既知の鳥として自分なりに管理できると便利である。例えば、鳥類学による分類も個人的な好感度による分類も同じHyperbookシステム内で管理できると、「ある時点で好感度の高い鳥は何科の鳥に多いか」などと言う質問を簡単に書ける。

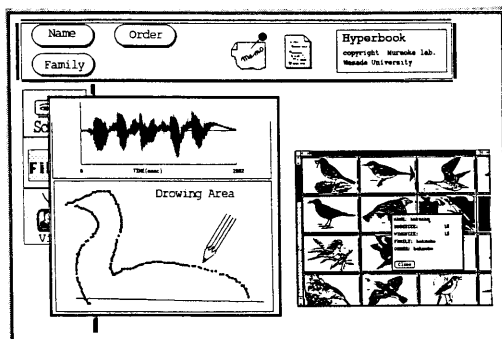


図1 Hyperbookシステムの例

### 3 鳥類図鑑Hyperbookシステムのデータモデル

上述したようなHyperbookシステムの核をなすマルチメディアデータベースのデータモデルは、マルチメディアを対話の手段に利用するマルチメディアコミュニケーション、様々なメディアを統合した実体の直接的表現、意味的な相対性を反映した関連表現を支援できなくてはならない。マルチメディアコミュニケーションは、利用者がイメージを伝えるための手段として不可欠である。実体の直接的表現は、曖昧なイメージ情報を元に取り出されたデータの集合を直接的に確認するのに有効である。なぜなら、利用者は個々の鳥を一定の属性で識別しているのではなく、比較対象の鳥によって確認する属性を変えるからである。また、意味の相対的な表現は、利用者のイメージ情報を扱う以上、重要な機能である。すなわち、利用者の鳥の特徴

に対する見方は、多種多様だからである。

しかしながら、既存のデータモデルでは、以上の要件を同時に満たすことができない。そこで、我々は、Hyperbookシステムのデータモデルとして、距離空間データモデルMeSOD[3]を提案し、採用している。

以下では、HyperbookシステムのデータモデルMeSODについて述べる。

#### 3.1 MeSODの概要

データモデルMeSODは、上述の問題点を解決するために、関連度の概念とオブジェクト概念を導入している。

関連度の概念は、MeSODにおいて最も重要な概念であり、マルチメディア検索など曖昧さを含んだ検索を関連度に基づく検索として一般化する。

MeSODのオブジェクト概念は、実世界の概念的なモデルの中でも関連度の概念を利用できる仕組みである。すなわち、実世界の实体に、複数の定義域を対応付けることができ、各定義域上で定義される距離空間に写像できる。この仕組みは実世界のより複雑な実体についても適用可能で、部品と全体の関係にある実体を直積距離空間の中で捉えたり、同値関係を元にした分類階層を距離空間の中で捉えることができる。そのため、実世界のより多様な意味を反映できる。

##### 3.1.1 関連度の概念による曖昧検索

MeSODにおいて最も重要な概念は、関連度の概念であり、従来のデータモデルでは扱われていない新しい概念である。関連度は、実世界の実体間の関連や実体集合間の関連の度合を意味する。例えば、実体間の類似度や依存度は、関連度の良い例である。

以下では、この関連度の概念が鳥類図鑑に対する検索要求のような曖昧な検索要求を扱えることを示し、関連度の具体的な表現について述べる。

##### (1)鳥類図鑑における曖昧検索と関連度の概念

2.で示した鳥類図鑑に対する曖昧な検索要求は、大きく分けてマルチメディア検索要求とイメージに基づく類似検索とに分けられる。マルチメディア検索は、鳴き真似やスケッチを検索要求とするような検索であり、イメージに基づく類似検索は、利用者が抱いているイメージと実際のデータとは一致しないが類似している、大きさや色のような情報を検索要求とする検索である。

どちらの検索も、MeSODでは関連度に基づく検索として一般化できる。すなわち、どちらの検索もMeSODでは、検索要求とデータベース内の実体との関連度がある範囲内にある実体集合の部分集合を取り出す操作

として表される。例えば、鳥類図鑑で、利用者の鳴き真似は、データベース内の鳥の鳴き声データと決して一致しないが、関連度として音声データ間の類似度を定義すれば、利用者の鳴き真似に似た鳴き方をする鳥の集合を取り出せる。また、大きさや色の場合は、大きさや色の近さを関連度とすれば、利用者の指定した大きさや色に近い鳥の集合を取り出せる。

(2) MeSODにおける関連度概念の実現

このように曖昧さを吸収する関連度の概念をMeSODは距離値によって表している。すなわち、MeSODは、関連度に基づく検索を距離空間における任意の検索要求の近傍を取り出す操作として表している。

先の鳴き真似の例では、MeSODでは鳴き真似と音声データとの距離を使って鳴き真似の近傍を取り出す。また、大きさや色も同様に、検索要求との距離値を関連度と見なして、近傍検索する。

3.1.2 MeSODのオブジェクト概念とその利点

MeSODのオブジェクト概念は、実世界の概念的モデルをその表現に基づく距離空間の世界に写像する。このオブジェクト概念は素オブジェクト、型オブジェクトと型付きオブジェクト(OBJ)の三種類からなる。

素オブジェクトは、実世界の概念的なモデルを記述するための単位概念であり、例えば、鳥類図鑑では鳥や鳥の属性に対応する。また、素オブジェクトの集合や、粗も素オブジェクトなので、個々の雀を素オブジェクトで表せば、雀の集合や雄雉の対も素オブジェクトである。

型オブジェクト(空間型)は、素オブジェクトを表現(メディア表現も含む)するための単位概念であり、その形式的背景は距離空間にある。例えば、鳥のシルエットを表現する場合、面図形集合を定義域とする1個以上の距離空間に対応する。

OBJは、素オブジェクトと空間型を統合するための単位概念で、素オブジェクトと空間型の組で表される。例えば、鳥のシルエットとその表現に対応する面図形距離空間の組は、一つのOBJである。

以上の概念によって、MeSODでは、実体の直接的な表現を可能にしつつ、曖昧なイメージ情報を元にした検索が可能である。例えば、鳥のシルエットの表現である空間型として、鳥の頭部に対応する部分の違いに強く反応する距離空間や尾部に対応する部分の違いに強く反応する距離空間を用意したとする。この場合、個々の鳥のシルエットは一つだが、写像する距離空間によって異なる曖昧検索ができる。つまり、利用者が頭の形がよく似ていると思えば、頭部に強く反応する距離空間に写像するし、尾羽根の形が似ていると思え

ば、尾部に反応する距離空間に写像する(図2)。

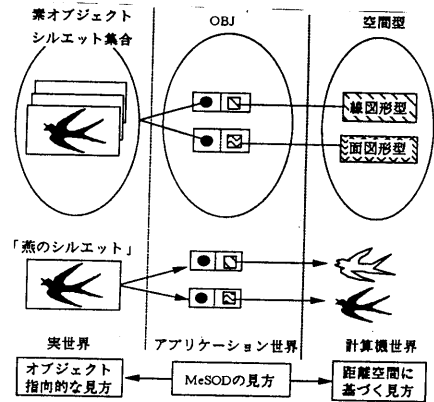


図2 MeSODのオブジェクト概念

このような実体の曖昧検索をより複雑な実体に適用する仕組みを、MeSODではOBJの概念から導出される複合OBJ概念として導入している。この複合OBJ概念には、積空間OBJと商空間OBJという、2種類の概念がある。

積空間OBJは、実体間のpart-of関連を距離空間に写像するメカニズムである。例えば、鳥の大きさが、翼長と全長からなるとすると、「鳥の大きさ」に対応する素オブジェクトは、翼長OBJの空間型と全長OBJの空間型から合成した積空間型を空間型として結合する。この時、「翼長」と「全長」を「大きさ」のプロパティ空間OBJと呼ぶ。一方、飛んでいる鳥を見たので、翼長でしか大きさを判断できない場合には、「大きさ」を積空間OBJとして定義しない方がよい。このように、状況に応じて積空間OBJを定義することによって、part-of関連を隔に使うことも、使わないこともできる。(図3)

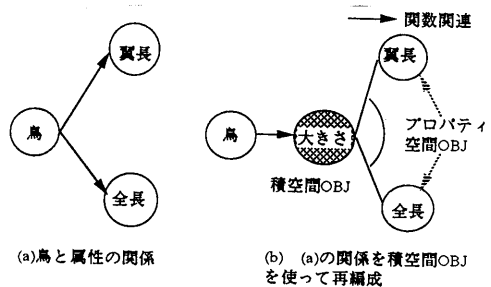


図3 積空間OBJとpart-of関連

商空間OBJは、汎化階層を同値関係による分類階層とみなすことによって、汎化階層を距離空間の世界で管理するOBJである。したがって、商空間OBJは多様な汎化階層の動的管理を可能にする。

例えば、「飛べる鳥」と「飛べない鳥」について、それぞれに「飛翔形」、「移動手段」という特徴を加

えたいとする。このとき、鳥空間OBJを{飛べる、飛べない}という同値関係で分割し、分割された二つの部分集合を空間OBJにできる(図4)。これを特化空間OBJ

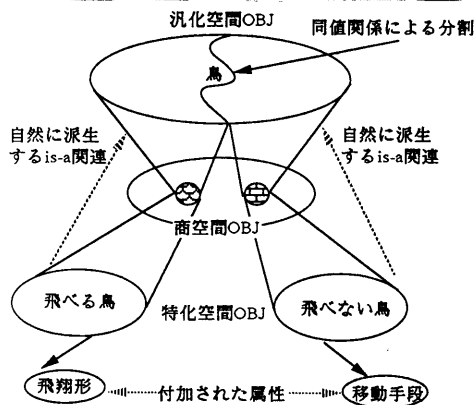


図4 汎化階層と商空間OBJ

といい、元の鳥空間OBJのことを相対的に汎化空間OBJという。商空間OBJは、これらの特化空間OBJを点OBJとして保持している。つまり、同値関係(汎化空間OBJの分割基準)に応じて、異なる商空間OBJが生成される。このように商空間OBJは特化空間OBJと汎化空間OBJの間にある is-a 関連を管理しているので、異なる観点からの分類も支援できる。

#### 4. Hyperbookシステムの開発環境

2節で述べたように、鳥類図鑑Hyperbookシステムは、利用者の入力方法や検索手順に、より多くの自由度が要求されるので、ユーザインタフェースと検索環境としての機能記述の負担が大きい。

アプリケーション開発環境MADE(MeSOD Application Development Environment)は、Hyperbook実装者のこのような負担を軽減し、システムの開発を容易にするための開発環境である。MADEの思想は、Hyperbook実装者がシステムの主要な振舞/機能を記述することに専念できる環境を提供することである。そこで、MADEは、MeSODに基づくデータベース管理システム(MeSOD層)の他に、プログラミングユーティリティ群(アプリケーションインタフェース層)、ユーザインタフェース構築ツール群(ユーザインタフェース層)を装備している。

以下ではMADEの提案理由、ソフトウェアアーキテクチャについて述べたあと、各層の概念と機能について説明する。

##### 4.1 MADEの提案理由

2節で述べたように、鳥類図鑑Hyperbookシステムは、

利用者の入力方法や検索手順に、より多くの自由度が要求されるので、ユーザインタフェースと検索環境としての機能記述の負担が大きい。例えば、ユーザインタフェースは、2.1(1)のシルエットの例のように、検索要求の入りに様々な利用者の好みがあり、複数の入力方法を支援するだけでも負担は大きい。

また、普通の図鑑と違って、利用者主体の検索手順を支援するための検索環境は、設計時に想定される利用者の様々な振舞に対応できなくてはならないので、より多くの機能記述が必要である。

このようなHyperbook実装者の負担を軽減し、容易にシステムを開発できるようなアプリケーション開発環境がMADEである。MADEは、Hyperbookシステムに必要な入力の多様性や検索手順の柔軟性を前提としたプログラミングツール群である。

##### 4.2 MADEの思想とソフトウェアアーキテクチャ

MADEの思想は、Hyperbook実装者がシステムの主要な振舞/機能の記述に専念できる環境を提供することである。システムの主要な振舞/機能の記述は、Hyperbookシステムにおいては、1つの検索項目にいくつかの入力方法に対応付けること、入力を検索要求として解釈しデータベースに要求を出すこと、検索結果を複数の表現に対応付けることなどである。

そこで、MADEに対する要件を以下の三つとする。

- (1)異なる種類のHyperbookシステムに共通なユーザインタフェースを容易に構築できること。つまり、鳥類図鑑でも昆虫図鑑でも2.で述べたような利用者の曖昧な記憶を表現するための入力手段、検索要求と検索結果である候補との関係が容易に分かるような表現を各々一つの機能とすること、
- (2)ユーザインタフェースとアプリケーションの振舞いの記述を分離する一方で、一つのプログラミングインタフェースを介して両者の対応付けを一括管理する機能を持つこと、
- (3)MeSODのオブジェクト概念に基づくデータ維持、管理の機能、関連度の概念に基づく近傍検索機能を具備していること。

これらの要件を受けて、MADEではユーザインタフェースを構築するための機能群とMeSODに基づくデータベース機能群を用意する一方で、統一的なインタフェースを介して、これらの機能をアプリケーションの記述に反映するための機能群の3層構造を採用している。各層は、それぞれ、ユーザインタフェース(UI)層、アプリケーションインタフェース層(APP)、MeSOD層と呼ばれる(図5)。

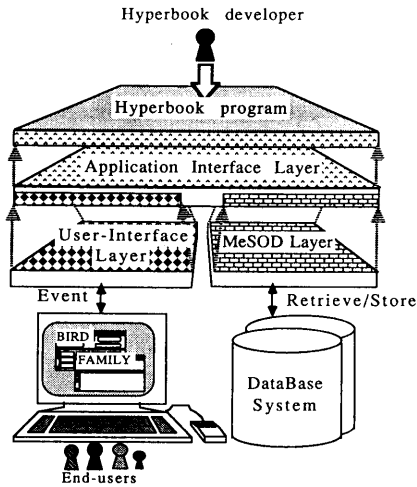


図5 MADEのソフトウェアアーキテクチャ

UI層は、Hyperbookシステムに必要な曖昧な検索要求の入力手段や曖昧検索の結果を効果的に表現手段をまとめる機能単位を持っている。APP層は、データベース内のデータ群を一つのオブジェクトとして扱う一方で、そのオブジェクトとユーザインタフェースとの対応も管理する機能単位を持っている。MeSOD層は、APP層にオブジェクト単位のアクセスや、距離に基づく近傍検索を提供する機能単位を持っている。

MADEは、以上のような3層構造を採ることによって、以下の2つの利点を持つ。

(1)アプリケーションの変更が局所的に押さえられる。すなわち図6に示すように、UI層やMeSOD層のプログラミングインタフェースが変更された場合でも、APP層で用意している機能単位の実装部を変更すれば、アプリケーションプログラムには本質的に影響が及ばない。例えば、UI層の一つの機能単位で色を表示する関数が、出力装置に依存しなくなった場合、UI層の機能単位の呼び出し方法が変わらなければ、APP層には変更はない。また、呼び出し方法が変わった場合でも、APP層内の機能単位の実装を変更すれば、アプリケーションプログラムには影響が及ばない。

(2)アプリケーション実装者が実世界の記述に専念できる。すなわち、APP層ではUI層の機能単位とMeSOD層の機能単位をまとめて扱う仕組(マッピングオブジェクトという)を用意しているので、実世界の振舞を記述することに専念できる。例えば、ある実体を削除する場合、HyperbookシステムはCRT上からもデータベース上からもその実体を消去しなくてはならない。APP層ではビューとデータとの対応の管理を行っているので、一連の操作をまとめた消去命令を用意できる。したがって、実体の消去を記述するには、実装者はAPP

層で持っている機能単位に対して消去の指令を送るようプログラムするだけである。

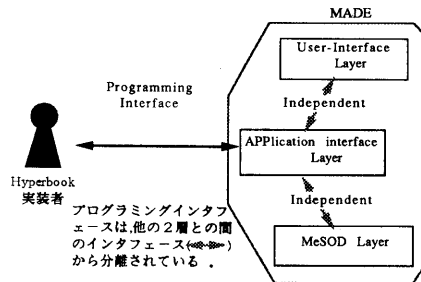


図6 MADEとHyperbook実装者

### 4.3 各層の概念と機能

#### (1)UI層 [6]

UI層は、入出力インタフェースを構築するための機能単位の集まりである。これらの機能単位を利用すれば、以下のような入出力インターフェースを容易に構築できる。利用者に曖昧なイメージ情報を検索要求として表現させる入力インタフェース、曖昧な検索要求に対する検索結果を効果的に表示する出力インタフェース、出力に対する直接操作を可能にする入出力インタフェース。これらの機能単位をUI層では、視覚オブジェクトと呼んでいる。

曖昧な検索要求の表現用入力インタフェースを構築するための視覚オブジェクトは、利用者の持つイメージを数量に変換する機能単位である。この視覚オブジェクトを使うと、複数の入力方法を持つ入力インターフェースを容易に構築できる。例えば、「鳥の大きさイメージ」の入力インタフェースを作る場合、スライダーによる入力も、手振りによる入力も同じ数値に変換できる。このような場合、両方の入力方式を同一の視覚オブジェクトの一状態として定義しておけば、一つの検索要求に複数個の入力方法を容易に対応付けられる。

検索結果を効果的に表示する出力インタフェースを構築するための視覚オブジェクトは、利用者の検索要求と検索結果であるデータとの関係の具体表示を司る機能単位である。例えば、UI層に実装されているスコープと呼ばれる機能は、データベースを広大な距離空間と見なし、検索要求の近傍にあるデータとの関係を「大きさ、遠近感、表示優先度」などを使って表現できる(図7)。

出力に対する操作を入力とする視覚オブジェクトは、いわゆる直接操作を可能にする機能単位である。具体的には、イベント管理機構で、任意の視覚オブジェクトに取り付けられる。例えば、スコープに表示された検索結果のアイコンをマウスでクリックした場合、こ

れをAPP層の機能単位に伝える。APP層の機能単位は、これを確認要求と見なして、よりリアルな写真なり、

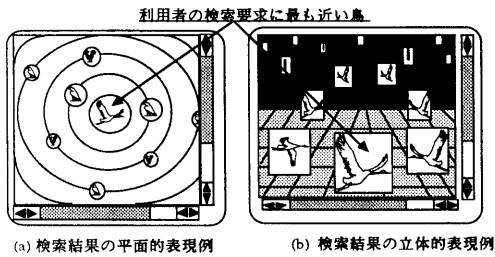


図7 スコープ機能の例

鳥のCGなりを利用者に提示する。

## (2) APP層[7]

APP層はMADEの利用者、すなわちHyperbook実装者に対するプログラミングインタフェースである。この層の最も重要な機能単位は、マッピングオブジェクトと呼ばれる。マッピングオブジェクトとは、データとその表現を対応付け、管理するための機能単位である。つまり、マッピングオブジェクトは、MeSOD層とUI層の両方の機能単位を結合する。図8の例では、鳥の大きさとそれに基づく曖昧検索をするためのユーザインタフェースを結合している。この時、マッピングオブジェクトでは、主に(1)UI層の機能であるスライダーからの入力を検索要求として、MeSOD層に近傍検索を依頼し、(2)その結果を構造体として、再びUI層に返すという操作を行っている。

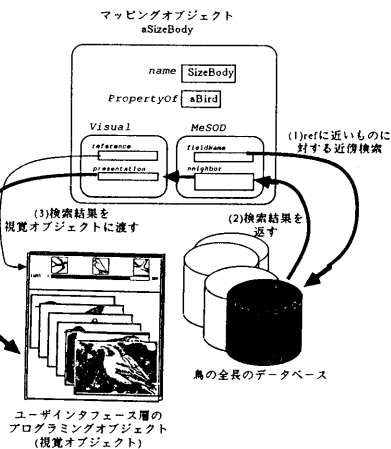


図8 マッピングオブジェクトの振舞い例

このマッピングオブジェクトを利用した結果、図鑑としての機能の記述に専念できる。これは、データとビューの対応付けの容易さとマクロ命令による高機能化によるものである。

## (3) MeSOD層(MeSODサーバー)

MeSOD層は二つの主要な機能単位があり、MeSODに基づくDBMSの機能をAPP層に提供するデータベースサーバである。1つの機能単位は、空間型に対応する機能単位で、Type-Objectと呼ばれる。もう一つは、素オブジェクト単位のアクセスや距離に基づく近傍検索を提供し、Allot-Objectと呼ばれる。Type-Objectは定義域に対する制約や距離関数の選択操作を持っている。Allot-Objectは、MeSODの素オブジェクトに対応する機能単位で、共通の素オブジェクトを含む複数個のOBJを表す。この時、1つのOBJはAllot-Objectの内部状態で表される。つまり、Allot-Objectはインスタンス変数によってType-Objectを参照しているので、この変数を変更することによって、異なるOBJを表す。

こうした対応関係は、MeSODの3種類のオブジェクト用の関係表によって管理されている。すなわち、素オブジェクトと空間型は独立した関係表で管理し、OBJを素オブジェクト集合と空間型集合間の関連集合の要素として管理している(図9)。

また、いずれのObjectもより複雑なObjectを組操作、集合操作により生成できるので、積空間OBJや商空間OBJを動的に導出できる。

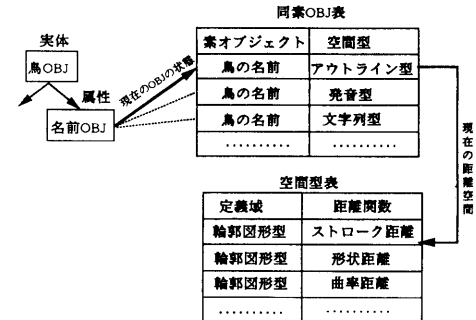


図9 OBJの実現方法

図9のようにAllot-Objectの内部状態として実現されているOBJは、関係表が管理している制約の範囲内で、アプリケーションから変更できる。そのため、Allot-Objectは、オブジェクト指向的な見方と距離空間による見方の両方ができる。

オブジェクト指向的な見方には、実体の直接的表現とメディアによらないモデル記述がある。例えば、鳥の鳴き声の集合に対応するAllot-Objectは、空間型として文字列型と音声型を結合できる。文字列型は、鳴き声を文字列で表すし、音声型はデジタル化した鳴き声を表す。したがって、例えば、利用者の鳴き真似が検索要求として入力された場合は音声型を結合するし、文字列の場合は文字列型を結合できる。どちらの場合

も、MeSOD層では、Allot-Objectの一つの状態として表されるので、アプリケーションからは「鳴き声」をメディア変換したように見える。

一方、距離空間に基づく見方としては、単純な構造のOBJから複合OBJまで、関連度による検索が可能である。例えば、一般に鳥の大きさは、全長と翼長で表される。この時、個々の鳥は、全長と翼長を合せて大きさという概念を持つと考えれば、大きさに対応する積空間OBJ(=Allot-Object)を生成し、大きさが近いということを全長も翼長も同じ位という意味で表現する。個々の鳥は、単に全長と翼長を持つと考えれば、全長、翼長とも個別に対応してAllot-Objectを生成する。

また、「科や目」、「飛行能力の有無」などの同値関係で表現可能な分類に基づくis-a関連は、商空間OBJにより管理されるので、一つのアプリケーションで同時に扱うことができる。つまり、ひとつのis-a階層は、一つの商空間OBJ(Allot-Object)を動的に生成することにより表されるので、複数の階層を同時に管理できる。これらの階層は、より分類が細かければ細かいほど、各オブジェクトを詳細に区別する距離空間から眺められる。

#### 4.4 サンプルアプリケーションによるMADEの評価

簡単な検索アプリケーションの作成に、MADEにすでに実装済みの機能(約8000行)を使った結果は良好であった。例えば、図10は図8の例のプログラムの一部で、1行目から4行目まででマッピングオブジェクトの定義を、それ以降で視覚オブジェクトの割り当てをしている。これらのマッピングオブジェクトを使ったコードは、Objective-Cで書いたアプリケーションプログラム約450行中、約85%までを占めている。したがって、アプリケーションコードの大半は、アプリケーション

```
1: aBird = [Appl create:"Bird"];
2: [aBird setProperty:5,
   "Name","Size","Habitat","Family","Order"];
3: [[aBird getProp:"Size"]
   setProp:2, "SizeBody","SizeWing"];
4: aSizeBody = [aBird getProp:"SizeBody"];
5: [aSizeBody setReference:
   [Scale setScaleWindow]];
6: [aSizeBody atRef:"setFunc:f:" with ALL
   with:borowseByScale];
7: [aSizeBody setPresentation:CARD_CANVAS
   on:[aSizeBody getRefFrame]];
```

図10 MADEを使ったコーディング例

の機能記述に関するコードで占められていることが確認できた。

## 5. おわりに

本論文では、距離空間データモデルMeSODに基づくアプリケーション開発環境MADEについて述べた。MADEがターゲットとする鳥類図鑑Hyperbookシステムのようなマルチメディアアプリケーションでは、利用者の検索要求は、曖昧であり、要求に近い検索結果を利用者が確認し、判断することが重要である。

そのためMADEでは、UI層、APP層、MeSOD層において以下のような機能を備えている。

- (1)UI層では、曖昧な入力を支援する複数の入力方式や検索結果の確認を支援する出力方式を視覚オブジェクトという機能単位で一元管理し、提供する。
- (2)APP層では、マッピングオブジェクトという機能単位でUI層とMeSOD層の機能単位を有機的に結合する。
- (3)MeSOD層では、Type-Object、Allot-Objectという二つの機能単位でMeSODのオブジェクト概念に基づくデータアクセス、関連度概念に基づく近傍検索を提供する。

また、簡単な検索アプリケーションの作成に、MADEに実装済みの機能を使った結果は良好であった。

今後の課題としては、MADEの各層には、(1)未実装の機能もあるので、その早急なる実装、(2)最終的な目標であるHyperbookシステムを実現するために、具体的なアプリケーション機能を想定した各層の機能追加及びその実装がある。

## 参考文献

- [1]中村:"野鳥の図鑑①~④",保育社, Apr. 1986
- [2]田淵,村岡:"電子化図書構想とデータモデリング",情処第35回全大6Bb-5, pp.363-364
- [3]田淵,村岡:"距離空間に基づいたデータモデルの提案について-MeSOD",情処研資料87-DB-61-1
- [4]Tabuchi, M and Muraoka, Y: "MeSOD: the Metric Spatial Object Data model for a multimedia application: Hyperbook", IEEE COMPCON'89 Spring, pp396-401, Mar. 1989
- [5]堀浩一: "単語の意味の学習について", コンピュータ科学, Vol. 3, No. 4 (1986), pp. 65-72.
- [6]星野他: アプリケーション開発環境MADEにおけるUser Interface層の概念, 情処第38回全大7R-7, 1989
- [7]山本他: 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ーフェース層の設計概念, 情処第38回全大7R-6, 1989