

マルチメディアデータベースのブラウザ記述
Multimedia Presentation System Based on Semantic Database Model

吉川耕平、花田恵太郎、佐藤亮一、宮本雅之
Kouhei YOSHIKAWA, Keitaro HANADA,
Ryoichi SATO, Masayuki, MIYAMOTO

シャープ株式会社 技術本部 コンピュータシステム研究所
Computer Systems Laboratories, SHARP Corporation

要約: SDMをもとにしたマルチメディアデータベースシステム(MMDB)と、これに対するデータブラウジングのためのユーザインターフェイス仕様記述言語を設計し、実装した。SDMは実世界に存在する個体をエンティティとして扱い、これをクラス、属性により構造化し、派生属性、属性連鎖などを容易に記述できる静的なデータ表現に優れたモデルである。このMMDBのデータブラウズを行うカードを使ったハイパーメディア型の環境の実現と、そのためのカードの形式及びカード間関係、カード上での操作をエンドユーザが簡単に記述し、カスタマイズすることを目指した仕様記述言語について述べる。カード上の表示オブジェクトには各種の操作を定義できるが、ここでは基本的なカードオープン、データビジュアライズの機能を実現した。

Abstract: We implemented DBMS with Multimedia Data Presentation Subsystem (DPS). We adopted SDM as a model of the DBMS to deal with multimedia in a unified concept. SDM is a powerful model to represent static data structure, that uses entity, class, and attribute, and has many rules to make derived-attributes. DPS realizes card-oriented interface like hyper-media environment. DPS allows users to specify a form of card which presents multimedia data, and has a facility to traverse data directly, such as open-card or visualize-object.

In this paper, overview of the SDM, discussion of how to retrieve multimedia information, specification of DPS, and presentation specification language are described.

[1] はじめに

情報処理のマルチメディア化に伴い、データベースシステムにおいてもマルチメディアを扱うための

- 1) 強力なデータモデル
- 2) 高度なユーザインターフェイス

が求められている。これらを満たすために我々はSDM (Semantic Database Model) [1] をもとに特定の応用を前提としないマルチメディアデータベースシステム(MMDB)を実現した。次にこのMMDBのデータブラウジングのためのユーザインターフェイスを記述する言語(ブラウザ記述言語)を設計、実装した。

マルチメディアデータベースシステム(MMDB)の基本的な機能として

1. マルチメディアデータの蓄積、検索
2. マルチメディアデータの編集
3. マルチメディアデータの表示

(プレゼンテーション)

があるが、本稿では特に、データ検索とその結果表示のユーザインターフェイス(UI/F)をエンドユーザにも簡単に構築できるようにするための、マルチメディアデータ表示に関する考察、及びブラウザ記述言語の仕様について紹介する。

意味論データモデルに基づくデータベースのビジュアルインターフェイスに関する研究として文献[2],[3]などがあるが、いずれもマルチメディア対応していない、ユーザによるカスタマイズができないなどの問題がある。

[2] SDM

SDMでは実世界に存在するものをそのまま表現するためエンティティと呼ぶデータ構成基本単位を用い、エンティティの集合としてのクラスや、エンティティ間の属性により構造化を行う。SDMの概念を図1に簡単に示す。

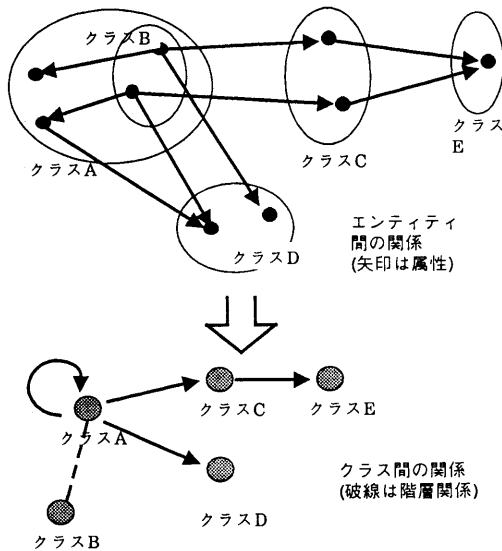


図1. SDM概念図

SDMで用いる概念を以下に掲げる。

- ・エンティティ
- ・クラス
- ・クラス間結合 (interclass connection)
- ・属性
 - ・派生属性
 - ・属性の継承

我々が使った概念を簡単に解説する。

1. エンティティ

エンティティはそれ自体は印字不可能なオブジェクトとしてデータベース内に存在する。システムが管理するエンティティIDはユーザには見えない。ユーザは固有テキストと呼ばれる文字列(名前)

前)を付けることができるが、その一意性は保証しない。エンティティは

- (1) 属するクラス
- (2) 属性とその値から定義する。値もまたエンティティである。

2. クラス

クラスは、主に以下のものから定義する。

- (1) 属性 (member attribute / class attribute)
- (2) 名前 (identifier)
- (3) クラスの is-a 階層関係

クラスには、階層の根になる BaseClass と他のクラスのサブクラスになる NonbaseClass がある。クラスの名前は BaseClass 中でユニークである。

3. 属性

属性は、

- (1) 属性名 (各々の BaseClass に対して一意)
- (2) 値クラス
- (3) member attribute / class attribute の指定
- (4) 単一値 / 複数値の指定

などから定義する。属性の値クラスが持つ属性を再帰的にアクセスしたものを属性連鎖と呼び、今後属性という際には属性連鎖も含むものとする。

4. 派生属性 (attribute derivation)

ある属性から別の属性を派生するため以下の機能がある。

- (1) 属性連鎖

$A1 = A2.A3.A4...$

- (2) その他

属性の ALIAS

属性値間の集合演算

属性値を用いる計算

属性値の最大値、最小値、平均

RECURSIVE

属性値による順序付け

我々がデータモデルとして SDM を採用したのは SDM の持つ以下の特徴に注目したからである。

▶ データ拡張性

- ・既存のスキーマに影響を与えずに新たなスキーマを定義できる。
- ・既存のスキーマを用いて新たなスキーマを定義できる。

- ・既存のスキーマに従うデータ(エンティティ)を後で定義したスキーマにも対応させることが可能。
- ・新たなメディアデータの取り込みが簡単(メディア独立性)。

▶エンティティ間の属性の多様性(冗長性)

- ・派生属性の定義が柔軟。

▶属性追跡の容易性

- ・属性連鎖を簡単に記述、処理可能。

属性の追跡とは、任意のエンティティから、それと何らかの属性で関連付けられたエンティティに直接的にアクセスすることを言い、この機能をトラバースと呼んでいる。トラバースの機能によりSDMデータモデルをもとに属性をリンクとみなす、いわゆるハイパーメディア型の環境が実現される。単なる属性だけでなく属性連鎖を直接トラバースすることで、直列の複数のリンクを一つのリンクと同じようにトラバースすることができる。

[3] マルチメディアデータベースシステム

SDMをもとにしたデータベースシステムとして以下の機能に関し実装を行った。(図2参照)

1) データ管理部、ファイル管理部

エンティティ、クラス、属性といった概念をもとに、数値、(不定長)文字列、画像、図形等のメディアのそれぞれをSDMにおけるベースクラスに対応させ、SDMデータモデルにおける統一的管理を実現した。ファイル管理部は、以下の情報を格納する。

- ▶ クラス定義情報
 - クラススキーマの管理
- ▶ 属性定義情報
 - 属性スキーマの管理
- ▶ エンティティ定義情報
 - エンティティとそれが持つ属性の管理
- ▶ 個々の属性情報
 - 属性と属性の値の管理

2) スキーマ/データ記述言語の設計及び同解釈部

SDMのスキーマ定義、およびデータ入力を行うためのスキーマ/データ記述言語を設計し、その解釈系を実装した。

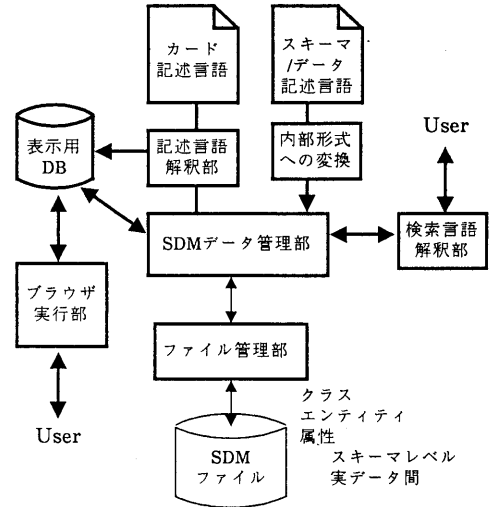


図2 SDMデータベースシステム

3) データ検索言語の設計及び同解釈部の実装

SDMデータベース上で条件に合致するエンティティを検索するための言語を設計し、実装している。検索は仮想的なクラスを抽出する、すなわちエンティティの集合を作ることととらえる。検索文は

クラス名 Δ where Δ 条件節

の形をしており、必要に応じて取り出す属性の限定、取り出すエンティティを並べる順序などを参照文として付加することができる。

この言語の特徴は、属性連鎖の記述が可能なこと、条件記述における多値属性に対応した比較演算子の存在、及び検索文中に検索文自身を入れ子にできることである。

条件記述における比較演算子については以下の機能を用意している。以下、 $A \text{ op } B$ という記法において、 A 、 B は共通のベースクラスに属するエンティティの集合を表す。

a) すべてのクラスに適用可能な比較演算子

- ・ $A \subset B$: A の全てのエンティティが B に属すれば真
- ・ $A = B$: A と B が共通の要素を1つ以上持てば真
- ・ $A !\subset B$: A が B に属さない要素を1つ以上持てば真
- ・ $A != B$: A と B が共通の要素を持たなければ真
- ・ $A \subset\supset B$: A と B が集合として、一致すれば真

b) 数値クラスの場合のみ適用可能な比較演算子

・ $A > B$: Bの全ての数値よりAの全ての数値が大きければ真

・ $A \geq B$: Bの全ての数値よりAの全ての数値が大きい
か等しければ真

c) 文字クラスの場合のみ適用可能な比較演算子

・ $A * = B$: Bの文字列の中に、それを前方に拡張すれば、Aのどれかの文字列に一致するものが1つ以上あるとき真

・ $A = * B$: Bの文字列の中に、それを後方に拡張すれば、Aのどれかの文字列に一致するものが1つ以上あるとき真

・ $A * = * B$: Bの文字列の中に、Aのどれかの文字列に部分文字列として含まれるものが1つ以上あるとき真

より複雑な検索を可能にするため、検索文の条件中にさらに検索文を含む形をとれるようにした。入れ子の深さに制限はない。例えば、「夏に咲くどの花よりも背が高い秋に咲く花」という内容に対応する検索文は以下のように記述できる。

```
植物 where 花.開花時期 = {季節|秋} and  
            高さ >  
            植物 where 花.開花時期 = {季節|夏}; :高さ;
```

検索文、参照分の記述では、ドット(.)による連結により属性連鎖を表現している。

検索に関しては、各種のクラス間集合演算、属性に対する条件など柔軟な表現が可能になっているが、エンドユーザに直接これを記述させるにはスキーマのすべてを提示しなければならず、またデータがマルチメディアであることを考えると、結果の表示は大きな問題となる。我々はトラバースという限られた機能であっても、その概念が分かりやすいこと、UI/Fが容易なこと、マルチメディアデータの表示に適していることなどから、トラバースを中心としたデータブラウズ機能を実現した。

[4] マルチメディア情報表示のUI/F

SDMをもとにしたマルチメディアデータベースにおけるデータ表示機能を考える時、画面上に表示されるもの(表示オブジェクトと呼ぶ)には、

・データそのものを表す機能

・ユーザが起動できる何らかの操作を表象する機能の二つの機能があると考えられる。ここでデータ編集などのデータ表示以外のことを考えなければ、操作については以下のようなものが考えられる。

・表示データの詳細を得る。(属性の追跡)

・一度に表示しきれないデータを一時的に表示する。(属性または属性値の一覧)

・表示データに付いている複数のリンクから一つを選択する。(表示する属性の選択)

我々の実装したデータベースにおいては、数値、文字列、イメージ等のシステムがあらかじめ用意した特定のクラス以外のクラスに属するエンティティは、単なるノードであり、その実態は各属性、属性連鎖の値を見ることにより把握される。そのため注目するエンティティを、カードという概念を用いて、必要な属性および属性連鎖とその値をこのカード上に配置し、ユーザに提示する。(図3参照)

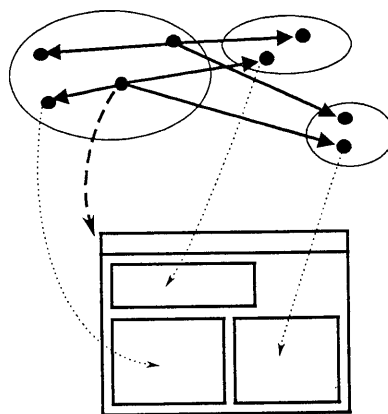


図3 SDMのエンティティをカードで表示する。

ここでカード上に配置する表示オブジェクトにフィールド、アイコンという概念を導入する。

フィールドは注目しているエンティティが持つ属性または属性連鎖とその値を表示するものである。ただし、値を表示するだけでなくその値を注目するエンティティとして新しいカードを開く、すなわち属性をリンクとみなしてそれを追跡するトラバース機能を持たせる。トラバースの機能はマルチメディアデータの最もプリミティブな検索である。

アイコンは注目するエンティティとそれが持つ属性を使う何らかの操作を表象するもので、アイコン自

体はSDMデータを表示している必要はない。ユーザがそれを選択すれば、付加されている操作が実行される。

これにより、あるクラスに対して、カードを作成し、そのカードにフィールド、アイコンを配置することによりSDMデータブラウザのインターフェイスが記述できる。

アイコンから起動する操作として現在カードオープン以外にビジュアライズという機能を実現している。ビジュアライズは別の表示オブジェクト(フィールド、アイコン、グループ(後述))を一時的に表示するものである。フィールドを表示させる操作は、カード内に配置しきれなかった属性、属性値を表示し、アイコンを表示させる操作は、さらに細分化された操作を起動することができる。

[5] カード記述言語

5.1. 表示オブジェクト

カードの形式を記述するため、枠組みとなるカードとカード内に表示する表示オブジェクトとして以下の三種類を用意する。カードと、これらの表示オブジェクトには名前を付けることができる。表示のための形状などは以下のパラメータを用いて定義する。各パラメータの記述で a) は省略不可能なもの、b) は省略可能なものである。定義する操作(アクション)については後に述べる。

A. カード

注目するエンティティに関する情報をひとつにまとめて表示するために用いる。

- a) 対応するクラス、カード内に表示するオブジェクト群
- b) カードのディスプレイ上での位置、サイズ

b. フィールド

注目するエンティティが持つ属性の値を表示するもので、文字、画像のいずれかを用いてカード内に矩形領域を取って表示される。SDMの性質上、属性は複数個の値を取ることを前提にする必要があり、文字型の値はスクロールバーを用いてそれらを同時に表示することができる。ただし、画像型の値はディスプレイのサイズの制約から同時には一つだけしか表示できないこととした。このた

め、他の値(画像)を見るためページめくりボタンを用意している。

- a) 表示する属性または属性連鎖
- b) カード内での位置、サイズ、データ表示個数、スクロールバーの有無、行数、枠、タイトル、文字色、背景色、その他

C. アイコン

注目するエンティティを基にした何らかの手続を表象するもので、フィールドと違って必ずしも何らかの属性に対応する必要はない。またフィールドと同じく文字、画像を用いてカード内に矩形領域を取って表示されるが、SDMデータベース内に存在する文字データ、画像データとは直接関係しなくともよい。

- a) なし
- b) カード内の位置、サイズ、枠、タイトル、アイコン画像(SDM外のファイルを参照可能)

D. グループ

表示法が全く同じ、複数のフィールド、またはアイコンを繰り返し記述する手間を省き、またそれらを規則正しく並べて表示するためグループという概念を用いる。これにより、ユーザの言語記述の負担を軽減する。ただしグループに階層はない。

- a) グループに属するオブジェクト群
- b) カード内での位置、サイズ、配列法

サイズ、位置に関しては省略する以外に、予約語による指定も可能となっており、その解釈を含むデフォルト値の処理は言語解釈部が一括して行う。またフィールドやアイコンにタイトルを付けることができるが、タイトルに関してもその文字列や位置、サイズなどの指定もできる(省略もできる)ようになっている。

5.2. 表示オブジェクトが持つアクション

個々の表示オブジェクトには先に述べたように、カードオープンまたはビジュアライズを定義することができる。

A. フィールド

カードオープンを定義できる。オープンするカー

ドの名前だけ記述すればよい。フィールドはなんらかの属性に対応しているの、その値を検索し、その結果を注目するエンティティとしてカードを開く。値が複数値の場合にはカードめくり機能が付いた複数のカードがオープンする。

B. アイコン

カードオープンまたはビジュアライズを定義できる。カードオープンの場合、一つの属性とカード名を記述する。ビジュアライズの場合は、その表示オブジェクトの名前を記述する。

カードを用いるハイパーメディア型の環境を実現するシステムをSDMデータブラウザと呼ぶ。ブラウザは、カード記述解釈部、ブラウザ実行部、SDMI/F部、表示用データベースからなる。記述解釈部はカードの形式、マウスイベントとそれに対応した操作との関係、及び操作の内容を記述したカード記述を読み込み、それを解釈して内容を表示用データベースに格納する。ブラウザ実行部はこの表示用データベースと実時間で通信しながらデータブラウザI/Fを実現する。システムの概要を図4に示す。

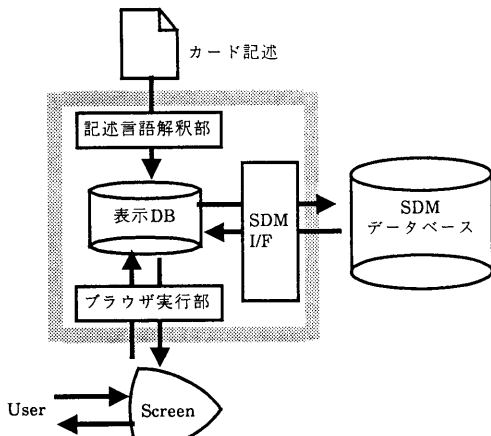


図4 SDMデータブラウザ

[6] 記述言語の解釈

カード記述言語に対し、下に述べるような処理、およびエラーチェックを行ったのち、その内容は表示用データベースに格納する。

6.1. サイズ、位置予約語の解釈

カード記述ではユーザに厳密な数値の記述の負担をかけないため、オブジェクトの位置やサイズに関しては以下の予約語を用意している。これらの予約語は解釈部により適当な数値に変換される。

・位置語：

予約語	解釈
TOP	配置領域の上辺に接する部分
BOTTOM	配置領域の下辺に接する部分
LEFT	配置領域の左辺に接する部分
RIGHT	配置領域の右辺に接する部分
CENTER	配置領域の上下左右の中央部
TOP_LEFT	配置領域の左上四等分領域内
TOP_RIGHT	配置領域の右上四等分領域内
BOTTOM_LEFT	配置領域の左下四等分領域内
BOTTOM_RIGHT	配置領域の右下四等分領域内

・サイズ語：

予約語	解釈
FULL	表示領域のすべてを使用
LARGE	表示領域の中で比較的大きな領域
NORMAL	文字型データ表示用標準領域
SMALL	画像型データ表示用標準領域
TALL	表示領域の中で比較的に長い領域
WIDE	表示領域の中で比較的横に長い領域
LONG	表示領域のたて幅すべてを使う、たて長領域
THIN	表示領域の横幅すべてを使う、横長領域

6.2. デフォルト値の補てん

カード記述の中で省略された部分のうち、解釈部において決定可能な部分、例えばフィールドのサイズ、データ表示個数、行数、文字の色やサイズ、フィールド、アイコンの枠の有無などにつきデフォルトを用いて数値を補てんする。なおデフォルト値はユーザ毎に設定可能となっている。

解釈部においても決定不可能な部分、例えばカードやビジュアライズオブジェクトのディスプレイ上での表示位置などは表示用データベースにも格納せず、ブラウザ実行部において、マウスイベントを受けてから決定する。

[7] ブラウザ実行部

ブラウザ実行部のシステム構成図を図5に示す。システムはイベントドライバ、表示データ管理、表示

ツール層からなる。表示ツール層はX-windowで実現している。

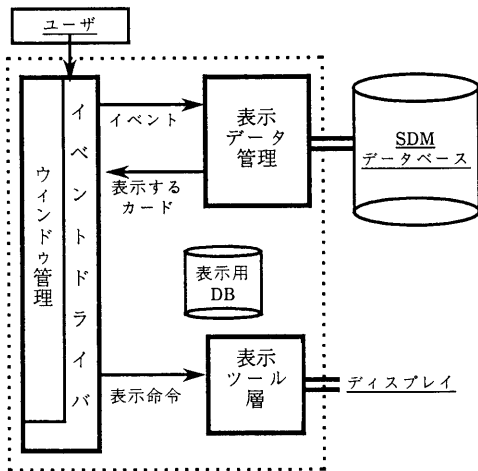


図5 ブラウザ実行部

7.1. スキーマグラフの提示

データブラウザのため、最初に足がかりとなるスキーマ構造の提示が必要となる。この提示インターフェイスの作成でもシステム内部ではカード、表示オブジェクトの概念を利用する。カードには一つの親クラスにつながるサブクラスがアイコンで一覧表示されており、その中で一つのアイコン、すなわちサブクラスを選択すれば

▶ さらにサブクラスを表示

▶ 属するエンティティをオープン

のメニューがフィールドとしてビジュアライズされ、ユーザが選択可能になる(図8参照)。これらのカードの構成はSDMデータベースが与えられれば自動的に生成され、表示用データベースに格納される。

7.2. イベントドライバ

イベントドライバはユーザからのマウス、キーボードの入力を受け、それが表示オブジェクトに対するものであれば、その表示オブジェクトをイベントとしてデータ管理部に送り、返り値として新たなカードの仕様を受け取る。これに実際の表示位置などの情報を補い、表示ツール層を使って表示を行う。イベントドライバは、アクション起動以外のカードの移動や上下関係の変更など単なるウィンドウシステムとしての機能も持つ。

7.3. 表示データ管理部

ここでは、必要に応じSDMデータベースのI/Fを実行しながらイベントドライバの要求に対しカード仕様を生成する。なおビジュアライズアクションにより表示されるオブジェクトはアクションが起動されるまで表示されることはないが、カードに付随し、見えないけれどもそのカード上に存在するものとして、カードの仕様をイベントドライバに渡す時に一緒に送られている。

図6、図7、図8にそれぞれカード記述の例、カードの表示例、データブラウザの実行イメージを示す。

[8] おわりに

カード記述言語によりマルチメディアデータベースでのデータ間のトラバース、および基本的なデータ表示のためのユーザインターフェイスが簡単に記述、生成可能となった。エンドユーザにとって、通常の検索言語によるインターフェイスに比べかなり操作性が上がっている。

今回はオープン、ビジュアライズの二つの機能だけを実装したが、その他にも自由なメソッドの記述を許すことにより、マルチメディアデータベースを使う様々なアプリケーションが容易に作成可能になると考えられる。

なお本研究は、通商産業省工業技術院大型プロジェクトの一環としてINTAP(財)情報処理相互運用技術協会)がNEDOから委託を受けて、実施したものである。

参考文献

- [1] M. Hammer and D. Mcleod "Database Description with SDM: A Semantic Database Model," ACM Trans. Database Syst., Vol. 6, No.3, pp.351-386, 1981.
- [2] J. A. Larson "A Visual Approach to Browsing in a Database Environment," IEEE Computer, Vol. 19, No. 6, pp. 62-71, 1986.
- [3] K. J. Goldman, S. A. Goldman, P. C. Kanellakis, and S. B. Zdonik "ISIS: Interface for a Semantic Information System," Proc. of ACM SIGMOD Int. Conf. on the Management of Data, pp. 328-342, 1985.

```

(CARD ka2
  CLASS 科
  OBLIST
    ( (OBJECT ka2_fields)
      (OBJECT ka2_icons)))
(ICON ka2_kuki
  LABEL 茎)
(ICON ka2_ha
  LABEL 葉)
(ICON ka2_mi
  LABEL 実)
(ICON ka2_ne
  LABEL 根)
(GROUP ka2_icons
  ICONLIST
    (ka2_kuki ka2_ha
     ka2_mi ka2_ne)
  SIZE LONG
  COLUMN 4
)
(FIELDka2_syoku
  ATTRIBUTE 花形
  SIZE FULL
  COUNT 3
  BAR ON
  LABEL (TEXT 花の形
        SIZE MAX
        ALIGN CENTER)
)
(FIELDka2_toku
  ATTRIBUTE 特徴
  SIZE FULL
  COUNT 3
  BAR ON
  LABEL (TEXT 特徴
        SIZE MAX
        ALIGN CENTER)
)
(GROUP ka2_fields
  FIELDLIST
    (ka2_syoku ka2_toku)
  COLUMN 2
  SIZE LARGE
)

```

図6 記述言語でのカード表現

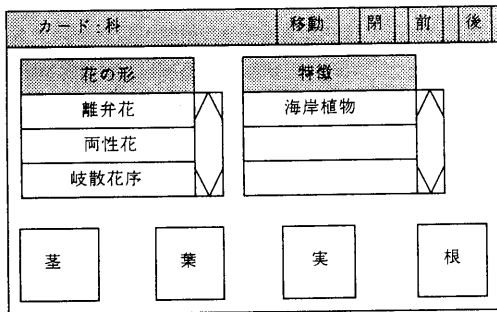


図7 カード表示例

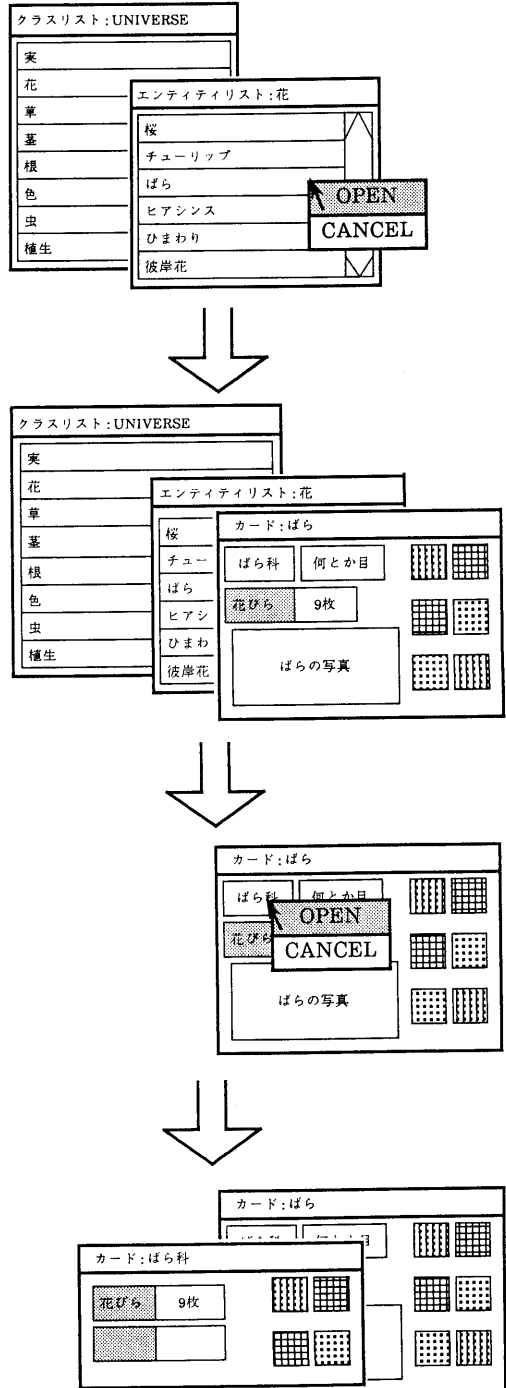


図8 実行例