データベースシステム 109－9
（1996．7．24）

宣言的なデータベース操作のための ビュー機能のインターフェース

村田 美友紀† 掛下 哲郎††
† 八代工業高等専門学校 情報電子工学科
†† 佐賀大学理工学部 情報科学科

我々は，これまでの研究でOODBを宣言的に検索・操作するためのビュー機能を提案した．ビュー機能はビュー間操作を定義することで，検索だけでなくDBの変更も行なえる．本稿では，ビュー機能の実装に向けて，初心者を対象としたGUIベースのユーザインターフェースを設計する．システムが表示するスキーマをコピー，編集することで，複雑な選択条件を容易に記述できる．また，検索されたインスタンスをビュー間でドラッグすることによって柔軟な操作が可能になる．これによって，初心者でも扱いやすいユーザインターフェースが設計できた．

User Interface of the View Function for Declarative Database Manipulation

Miyuki Murata† Tetsuro Kakeshita††
†Department of Infomation and Electronics Engineering,
Yatsushiro National College of Technology
††Department of Information Science, Saga University

We have proposed the view function for declarative manipulation of OODB. Along with the data retrieval function, the view function can update the database by defining couple of operations between two views. In this paper, we design GUI-based user interface of the view function for beginners. A user can easily define complex selection condition by copying and editing the database schema. By dragging retrived instances between views, a user can manipulate the database. Thus a beginner can use the view function easily.

1 はじめに

データベース (DB) 操作言語は，DB の使い易さを決定する重要な要因である．オブジェクト指向データベース (OODB) の操作には，永続データを操作しやすいように拡張した手続き型言語 C++が主として用いられる．しかし複雑な手続きの記述を必要としない非手続き型言語が望まれている．我々はこれまでの研究で OODB を宣言的に操作するためのビュー機能を提案した [1, 2, 3]．

ビュー機能は，ビュー定義とビュー間操作からなる．ビュー定義はインスタンスを検索するために用いる．ビュー間操作 (移動，追加，コピー，生成，削除) は DB を変更するために用いる．ビュー機能を用いた DB 操作は，(1) 宣言的な操作が可能である，(2) データモデルに依存しない，(3)GUI に対しての親和性が高い，(4) データ一貫性が検査できる，(5) インスタンス指向/集合指向の両方の操作を支援している，などの利点がある．

本稿では，初心者を対象としたビュー機能の実装に向けてユーザインターフェースの設計を行なう．ビューをウインドウ，ビュー間操作をポインティングデバイスで実現することにより，GUI との親和性を向上させる．GUI は，視覚情報 (メニュー，ボタンなど) により操作を容易にし，データ構造の理解を助ける．

ビュー機能のユーザインターフェースとして，スキーマを表示するスキーマウィンドウとビューを定義し，結果を表示するビューウィンドウを定義する．ビュー定義のための選択条件の設定は，スキーマウィンドウからビューウィンドウにコピーしたノード集合を編集 (質問グラフを生成) することで表現する．また，射影属性の設定はビューウインドウ上のノードの属性を選択することで表現する．ノードのコピー，編集はポインティングデバイスによる操作やメニューの選択によって実行される．ビュー間操作は，2 つのビューウィンドウ上で行なう．ビューウィンドウ上の組を選択し，操作することでビュー間操作を表現する．各操作はポインティングデバイスを用いて行なう．このとき，操作の区別は操作モード切替えによって行なう．

OODB のための宣言的な操作言語として，ODMG によって OQL が提案されている [4]．OQL は複合オブジェクトへのアクセスを許す SQL タイプの言語である．OQL は木構造でない複合オブジェクトの生成ができない点でビュー機能に劣る．さらに，OQL の GUI 化は容易ではない。また，GUI を持つものとして，Gyssens 等によって GOOD が提案されている [5]．GOOD はグラフとのパターンマッチングを基本操作とするため，ビュー機能と同様にビジュアル的な操作が可能である．しかし，データ操作機能が貧弱なので，DB の変更の際にマクロ定義など，複雑な操作が必要であるため，ビュー間操作に比べて操作性は劣る．

本稿は以下の通りに構成されている．2 節ではビュー機能について定義する．3 節では，ビュー機能ユーザインターフェースの基本構成を設計する．4 節では，ビュー定義のユーザインターフェースを設計する．5 節では，ビュー間操作のユーザインターフェースを設計する．最後に 6 節でまとめを行なう．

2 ビュー機能

本節では，ビュー機能 (ビューおよびビュー間操作) を定義する．ビュー機能を定義するに先立ち，基本データモデルを定義する．次に基本データモデル上でビュー機能を定義する．

2.1 基本データモデル

基本データモデルのスキーマ Sはノード Nの有限集合である．ノードの構成要素は属性とリンクである．属性は基本データ型であり，リンクはノードを参照するための型である．ノードには組ノード，集合ノード，選択ノードがあり，それぞれ以下のように定義される．

- 組ノードは，属性/リンクの集合から構成されるノードである．
- 集合ノードはただ一つのリンクから構成される．
- 選択ノードは一つ以上のリンクから構成される．

次に，インスタンスの定義を行なう．属性の値は対応する型を持ち，リンクの値はリンクが参照するノードのインスタンスの識別子である．各ノードインスタンスは次のように定義する．

- 組ノードのインスタンスは，構成要素の属性/リンクのそれぞれの値からなる組である．
- 集合ノードのインスタンスは，NIL でないリンク値の集合である．
- 選択ノードのインスタンスは，構成要素のリンクいずれか一つのリンク値からなる組である．

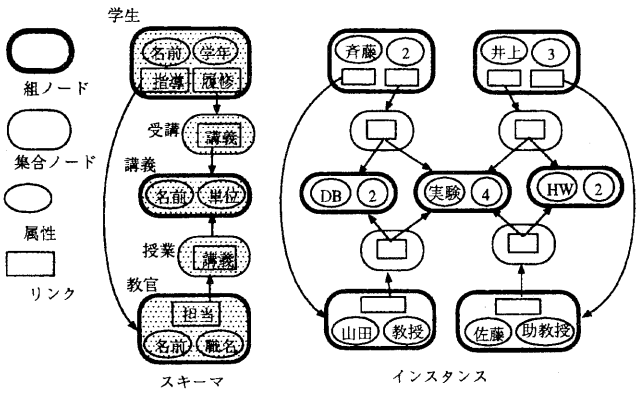


図 1: 基本データモデル

ノード Nのインスタンス I_Nの属性/リンク Xの値を $I_N.X$で表す．ノードのインスタンスはそれぞれ固有の識別子を持つ．スキーマ Sのインスタンスは，Sに属するノードのインスタンス全体からなる集合である．基本データモデルのスキーマとインスタンスの例を図 1に示す．

2.2 ビュー

本節ではビューを定義する．ビューは選択条件を満足するインスタンスについて射影属性で指定された属性の組を表示する．ビューを定義するに先立ち領域変数を定義する．ノード Nに対し，領域変数 $D_N^1, D_N^2, \cdots$を割り当てることができる．D_N^i は Nの任意のインスタンス I_N^jにマッチングできる．D_N^iが I_N^jにマッチングした時，Nの任意の属性/リンク Xについて $D_N^i = I_N^j, D_N^i.X = I_N^j.X$ が成立する．

ビュー Vは射影属性と選択条件の組によって定義される．射影属性 $PA(V)$ は，属性 $D_N^i.A$ を要素とする集合であり，ビュー上に表示する値を指定する．また，引数である領域変数にマッチングするインスタンスの数を返す関数 **Count** も射影属性に含めることができる．選択条件 $SC(V)$ は以下の 1〜3 で定義する論理式であり，インスタンスを検索するために用いる．関数 **Count** も選択条件に含めることができる．また，$SC(V)$ に含まれる領域変数の集合を $DV(V)$ と定義する[1]．

1. 項$_1$ θ 項$_2$ は論理式である．但し，項$_i$$(i = 1, 2)$ は領域変数 D_N^i，属性 $D_N^i.A$，リンク $D_N^i.L$，定数 c，**'undefined'** のいずれかである．また，θは '$=$', '$\neq$', '$<$', '$>$', '$\leq$', '$\geq$', '$\subset$', '$\supset$', '$\subseteq$', '$\supseteq$', '$\ni$', '$\not\ni$' のいずれかである．項$_1$と項$_2$ の間には，比較演算子 θが定義されていなければならない．

2. '**true**(D_N^i)' は選択条件である．

3. 式 F_1と F_2が選択条件ならば，$(F_1 \wedge F_2)$，$(F_1 \vee F_2)$，$\neg F_1$は選択条件である．

同一ノードに対し複数の領域変数を定義した場合，各領域変数にマッチングするインスタンスの集合は互いに素である．よって，ノード間の比較条件 $(D_N^1 \theta D_N^2)$ を省略できる．ビュー Vの選択条件 $SC(V)$ を真にするように $DV(V)$ に割り当てられたインスタンスの集合を，Vに属するインスタンスと呼ぶ．Vに属する各々のインスタンスに対して，$PA(V)$ で指定される属性値を射影することによって構成される組をビュー Vに属する組と定義する．ビュー Vは，Vに属するすべての組を表示する．ビューは **DBMS** によって識別子を割り当てられ，必要に応じて固有のビュー名を持つ．

2.3 ビュー間操作

ビュー間操作はインスタンスを変更するために用いられる．ビュー間操作は，ビューに属する組を操作することで，間接的にビューに属するインスタンスを操作する．ビュー間操作として，移動，追加，コピー，生成，削除を定義する．代表的なビュー間操作である移動操作の定義を以下に示す．

移動操作は，インスタンスの属性/リンクを変更するために利用される．ビュー V_s, V_dを定義し，V_sに属する組 Iを選択して V_dに移動する．この時，$D_N \in$

[1] $PA(V)$ に属する領域変数は，常に $SC(V)$ 中に出現する．

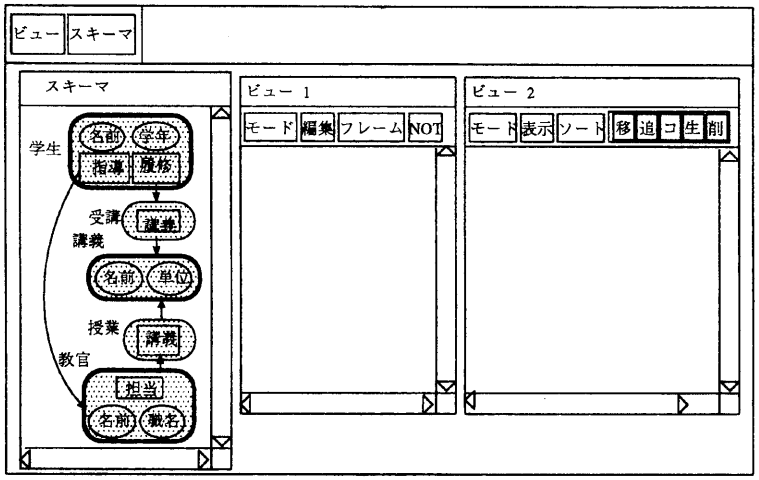


図 2: ビュー機能のユーザインターフェース

$DV(V_s) \cap DV(V_d)$ を満たす領域変数 D_N がマッチングするインスタンス $I_N(\in I)$ が変更の対象となる．移動操作により，I_N の属性/リンク X の値が $SC(V_d)$ を満足するように変更される．I_N が集合ノードのインスタンスでない場合，$SC(V_d)$ を満足する $I_N.X$ の値が唯一に決定されず，かつ $I_N.X$ の変更前の値が $SC(V_d)$ を満足しないならば，移動操作は拒否される．I_N が集合ノードのインスタンスである場合には，$SC(V_s)$ を満たすリンク値が $SC(V_d)$ を満足するリンク値の集合に置換される．

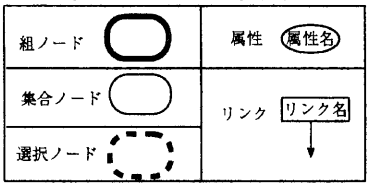
他のビュー間操作についてはその目的を示す[2]．追加操作は集合インスタンスに新たなリンク値を追加するために利用される．コピー操作は任意のインスタンスをコピーして，新しいインスタンスを生成するために利用される．生成操作は任意のインスタンスを生成するために利用される．削除操作は任意のインスタンスを削除するために利用される．

3 ビュー機能のユーザインターフェース

ビュー機能を起動すると図 2に示すメインウィンドウが開かれる．メインウィンドウにはスキーマウインドウとビューウインドウが表示される．スキーマウインドウはスキーマを表示する．スキーマウインドウ上に表示されたスキーマをテンプレートとしてビューを定義する．ビューウインドウはビューに対応する．ビューウインドウはビューの定義，インスタンスの検索を行なう．ウィンドウの基本的な機能 (スクロール，大きさの変更) は，標準的なウィンドに準拠する．

[2]詳しくは文献 [3] を参照すること．

表 1: スキーマの表記法



3.1 スキーマウィンドウ

スキーマウインドウは，スキーマの全体構造を表示する．スキーマウィンドウは，メインウィンドウのメニュー「スキーマ」から「開く」を選択することで開かれる．スキーマウインドウ上では，スキーマに属する全てのノード，各ノードに属する全ての属性/リンクを表 1に示す記号を用いて表示する．リンクによるノード間の参照は，有向枝で表現する．ノード，属性/リンクを選択すると，それらの特性 (名前など) が表示される．また，選択されたノード集合はビューウインドウにコピーできる．

モード	種類	ラベル	サブメニューラベル，ラジオボタンラベル
選択条件	メニュー	モード	射影属性モードへ，表示モードへ
	メニュー	編集	コピー，ペースト，カット
	メニュー	フレーム	フレーム定義，フレーム未定義，ページ生成，ページ削除，ページ切替
	実行ボタン	NOT	
射影属性	メニュー	モード	選択条件モードへ，表示モードへ
	実行ボタン	Count	
表示	メニュー	モード	選択条件へ，射影属性へ
	メニュー	表示	組，グラフ
	メニュー	ソート	ソート実行，条件設定
	ラジオボタン		移動，追加，コピー，生成，削除

表 2: ビューウィンドウのメニューとボタン

3.2 ビューウィンドウ

ビューウィンドウはビューを定義し，検索結果を表示する．ビューウィンドウはメインウィンドウのメニュー「ビュー」から「新規作成」または「開く」を選択することで開かれる．ビューウィンドウは，選択条件モード，射影属性モード，表示モードを持つ．各モードにおけるメニュー，ボタンを表 2に示す．

以上でビュー機能ユーザインターフェースの全体構成が設計できた．これより，ビュー定義，ビュー間操作ユーザインターフェースの詳細設計を行なう．

4 ビューウインドウのユーザインターフェース

4.1 ビュー定義の表現と機能

領域変数の定義は，スキーマウィンドウまたはビューウィンドウ上のノードをコピーすることで表現する．コピーされたノードにはシステムによって領域変数が与えられる．ビューウィンドウ上のノードは領域変数に対応する．

選択条件の設定は，定義ウィンドウ上で質問グラフを作成することで表現される．2節で定義した選択条件は以下のように分類できる[3]．ここで，$D_N, D_{N'}$ は，ノード N, N' に対して定義される領域変数とする．また，Xは属性/リンク，A, Bは属性，L はリンクとする．

1. 属性値と定数の比較 $D_N.A\theta c$ (θは '=', '$\neq$', '<', '>', '$\leq$', '$\geq$' のいずれか)
2. リンク値とノード識別子の比較 $D_N.L\theta D_{N'}$ (θ は集合ノードでない場合，'=' または '$\neq$'，ノードの場合，'$\ni$' または '$\not\ni$')
3. 属性/リンク値同士の比較集合 $D_N.X\theta D_{N'}.X$ (θ は集合ノードでない場合，'=', '$\neq$', '<', '>', '$\leq$', '$\geq$' のいずれか，集合ノードである場合，'=', '$\neq$', '$\subset$', '$\supset$', '$\subseteq$', '$\supseteq$' のいずれか)
4. 属性値，リンク値の未定義 $D_N.X$='undefined'
5. ノード Nの全てのインスタンスについて真になる true(D_N)
6. 式 F_1, F_2の AND, OR, NOT

1 から 5 は，図 3のように表現する．

3 については，どちらかのノードの属性/リンクに他方のノードの属性/リンクをドラッグし，演算子を記入する．それぞれの属性/リンクには，等価な演算子と互いの'ノード名．属性/リンク名'が表示される．

6 を表現するためにフレームを定義する．フレームは領域変数の集合に対して定義される．フレームは複数のページを持つ．各ページは任意の選択条件を設定できる．フレームは入れ子で定義できる．フレームの操作はメニュー中のフレームを用いて行なう．$F_1 \wedge F_2$ は両式を同一ページに記述することで表現する．$F_1 \vee F_2$ は，両式を同一フレームの異な

[3] 選択条件には，関数 Count も記述できる．しかし，Count はキー制約などのデータ一貫性を検査するために，システムが用いる関数である．よって本論文では，その表現法を定義しなかった．しかし，ユーザがこれらの関数を設定できるような拡張は可能である．

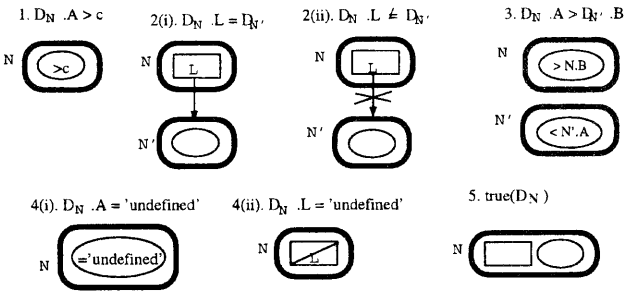


図 3: 選択条件の表現法

るページに記述することで表現する．$\neg F_1$はページにNOTフラグを立てることで表現する．各ページには属性/リンク単位ではなく，ノード単位で表示する．これによって，属性/リンクが所属するノードが明示できる．表示単位を属性/リンクとするならば，その所属ノードの情報を別に表示しなければならず，ユーザにとって好ましくないと思われる．フレームを用いたOR条件表現の例を示す．

例 1 図 4に示すスキーマに対して，次の選択条件を考える．

$$(D_N.L = D_{N'}) \wedge ((D_N.A = \text{'}a\text{'}) \vee (D_{N'}.B = \text{'}b\text{'}))$$

まず，ノードN, N'に対してフレームを定義する．定義したフレームに対してページを2つ生成し，それぞれのページ上で図 4に示すように属性値，リンク値を設定する． 各ページはOR条件を構成する各項に対応する．同一フレームに対するページは重なって表示されるが，ページの切替機能を設けたことで各ページを自由に移動できる． □

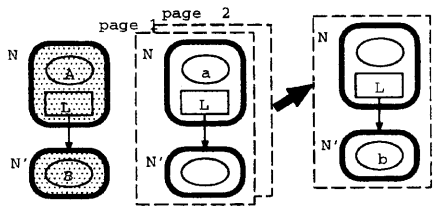


図 4: OR条件の表現法

射影属性の設定は，射影属性設定モードで行なわれる．射影属性に指定された属性およびCountに対応するアイコンは識別表示される．

表 3にビューを定義するために必要な機能および操作法をモードごとにまとめて示す．

4.2 ビュー表示の表現と機能

表示モードにおける表示形式は，組またはグラフが設定できる．それぞれは図5のように表現する．グラフ形式において，定義されている領域変数にマッチングするインスタンス中で連結しているインスタンス間は無向枝で連結される．組またはグラフの各要素について対応する属性またはCountがウィンドウの上段に表示される．属性は‘ノード名.属性名’，Countは‘Count (ノード名)’と表記される．

表示されている組は属性集合について設定した条件についてソートできる．設定できる条件は，属性の優先順位と属性値の優先順位 (文字コード順，数値順，昇順，降順など) の選択である．

ビュー表示に必要な機能と操作法を表 4に示す．

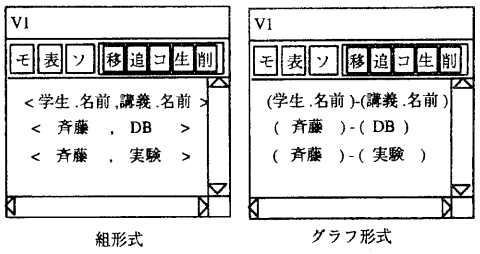


図 5: 検索結果の表示形式

4.3 インスタンス検索の例

ビュー定義を用いたインスタンス検索の例を示す．

例 2 図 1のスキーマとインスタンスを考える．名前が斉藤である学生が履修する講義名を検索するためのビューV_1を定義する (図6)．まず，スキーマウインドウから関連するノード (学生， 受講，講義) を定義ウィンドウにコピーする．選択条件モードで

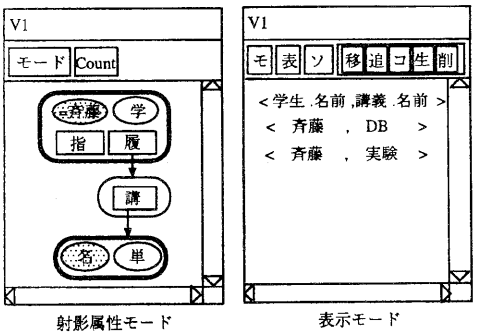


図 6: ビュー定義

選択条件の設定を行なう．ここでは，属性 '学生.名前' に '= 斉藤' を設定する．次に射影属性モードに切替えて射影属性の設定を行なう．ここでは，'学生.講義', '講義.名前' を選択する．最後に表示モードに切替えると検索結果が表示される． □

5 ビュー間操作のユーザインターフェース

ビュー間操作は，表示モードであるビューウィンドウ間で行なわれる．ビュー間操作は，次の手順で行なわれる．2 つのビューウインドウ V_s, V_d を開く．V_s では，変更対象のインスタンスを検索する．V_d では，変更対象のノードを V_s からコピーし，変更後のインスタンスの状態を表現するように編集する．コピーされたノードはコピー元のノードと同一の領域変数に対応する．選択条件設定に必要なその他のノードは固有の領域変数を持たせるために，スキーマウインドウからコピーする．これによって，変更対象のノードと参照のために用いられるノードを区別できる．ビュー間操作を実行するのに必要な機能は，組の選択，組の移動である．組の選択は，表示ビューが組形式の場合は組，グラフ形式の場合はインスタンスを連結する無向枝を選択する．ビュー間操作はメニューよりいずれかの操作モードボタンを選択することで指定する．組の移動は，識別されたインスタンスをドラッグすることで表現する．ビュー間操作の例を挙げる．

例 3 図 1のスキーマとインスタンスについて，DB を履修する学生の学年を 3 に変更する操作を考える． まず V_2 を定義し，変更の対象となるインスタンスを検索する．V_3 では，変更対象であるインスタンスにマッチングする V_2 上のノード '学生' をコピーし，それを編集して変更後の属性値を設定する (図 7(a))．次に 2 つのビューウィンドウを表示モードに切替える (図 7(b))．V_2 のビュー間操作ボタンから移動を選択し， V_2 上の操作対象の組<2， DB>を選択し， V_3 の表示ウィンドウにドラッグする．これにより移動操作が実行され， V_3 の表示ウィンドウ上に変更された組 <斉藤，3>が表示される． □

6 おわりに

本稿では，宣言的な DB 操作のためのビュー機能を実現するためのユーザインターフェースを設計した．ビュー定義はスキーマウィンドウからコピーしたノードに対して，ビューウィンドウ上で質問グラフを作成することで実現する．ビュー間操作はビューウインドウ上の組を選択し，それを他のビューウインドウにドラッグすることで実現する．本稿で提案したユーザインターフェースは，特別な言語や文法の習得を必要としないため，初心者にとっても扱い易い．

今後は，Tcl/Tk [6] を用いてプロトタイプを作成し，評価を行なう．上級者も考慮したユーザインターフェースは，しばしば使う機能や，操作列をプロトタイプ上でリストアップして，それらに対応するショートカットキーやツールバーの定義，またはマクロ定義機能などを充実させることで行なう [7].

参考文献

[1] 掛下, "ビュー機能を用いた構造化オブジェクトの段階的検索と編集操作", Proc. ADBS'93, pp.93–102, 1993.

[2] 村田, 掛下, "ビュー機能を用いた E-R モデルデータベースの操作", 情報処理学会 DB システム研究会 101–5, 1995.

[3] 村田, "宣言的なデータベース操作のためのビュー機能", 佐賀大学大学院修士論文, 1996.

[4] F. Bancilhon and G. Ferran, "The ODMG standard for object databases". Proc. DASFAA, pp. 273–283, 1995.

[5] M. Gyssens, et al., "A graph-oriented object model for database end-user interface", Proc. SIGMOD, pp.24–33, 1990.

[6] J. K. Ousterhout, "Tcl & Tk ツールキット", ソフトバンク株式会社, 1995.

[7] A. Cooper, "ユーザーインターフェースデザイン", 翔泳社, 1996.

モード	機能	操作
スキーマウインドウ		
	コピー	標準的な **draw** 系ツールに準ずる.
	選択	同上
ビューウインドウ		
選択	ノード集合の選択	標準的な **draw** 系ツールに準ずる.
条件	コピー, ペースト, カット	同上
	フレームの選択	同上
	フレーム定義	対象ノード集合を選択した後, フレーム定義を選択する.
	フレーム定義の解除	対象フレームを選択した後, フレーム定義解除を選択する.
	ページの生成	対象フレームを選択した後, ページ生成を選択する.
	ページの削除	対象ページを選択した後, ページ削除を選択する.
	ページの切替え	対象フレームを選択した後, ページ切替えを選択する.
	属性値の設定	対象属性に値を記入する.
	リンク値の設定	対象リンクの矢印が参照する先を変更する.
	属性/リンクの選択	標準的な **draw** 系ツールに準ずる.
	属性/リンクの移動	対象属性/リンクを選択し, ドラッグする.
射影	属性, **Count** の選択	標準的な **draw** 系ツールに準ずる.
属性	射影属性の設定	対象属性を選択する.

表 3: ビュー定義に必要な機能

機能	操作
表示形式の切替	組またはグラフを選択する.
属性, **Count** の選択	標準的な **draw** 系ツールに準ずる.
ソート条件の設定	条件設定を選択し, 開かれるダイアログボックスで設定する.
ソートの実行	属性または **Count** を選択した後, ソート実行を選択する.

表 4: ビュー表示に必要な機能

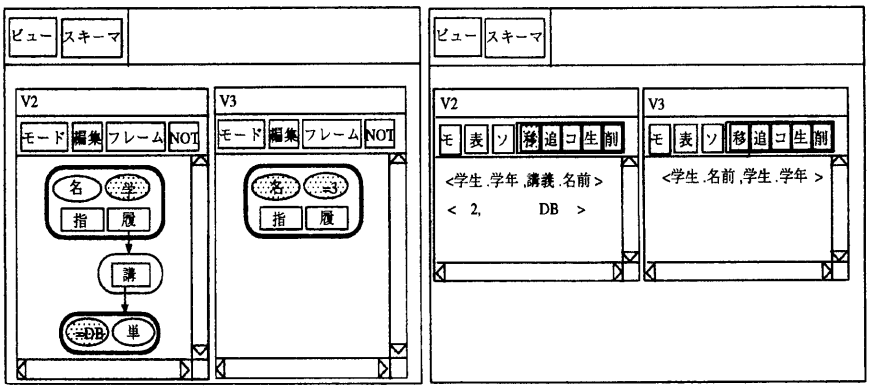


図 7: ビュー間操作