

研究ノート ポップアップ・ノード機能に焦点を置いた パス歩行とフォーム生成

中西 昌武*

Path Walk and Form Generation Focused on Pop-up Node Function NAKANISHI, Masatake

1. はじめに

これまで筆者は、単一エントリと複数エントリを問わず、エントリノードから順次歩行可能なパスのみをフォーム生成対象とするパス歩行の原理を追及してきた。2002年[1]に示したフォーム生成パターンやテンプレートはその体系化であったし、2007年[2]に示したパス歩行行列法はその行列原理であったし、2010年[3]にはビュー構造の行列表現を提案した。いずれも最も基本的なフォーム生成の原理の確立を目指すものであった。昨年筆者は、これを土台とする新たなフォーム生成への探究として、要求分析工程でしばしば登場するスキーマ未定義の仮想定ノードの取り扱い方という困難な課題を指摘し、その解決方法を示した[4]。また要求解明の曖昧さが理由で現れる非連結データソースなどの問題を効率的に解決する数理的な検出方法についても提案した[5]。これらはいずれもエントリノードから順次歩行可能なパスのみをフォーム生成対象とするパス歩行を扱っている。

本稿ではこれとは別の観点の課題を提示する。カーナビゲーションを使って走行するとき、公共施設やマーク地点に近接するとそれらのアイコンがポップアップ表示される。また渋滞や事故車などの情報も表示される。これらの運転示唆情報は、データベース検索技術者が好んで言うところの「ブラウジング情報」に当たるが、情報過多にならない適度な表示であれば、利用者の情報環境を有益なものとする。この例を参考としつつ、本稿では、パス歩行にも、このようなポップアップ・サービス機能が期待できる場合があると考え。すなわち、あるノードまでパス歩行したときに、このノードと組み合わせてクロス参照軸を生成できるノードがあれば、そのことをユーザにポップアップ提示して、①従来型のパス歩行を続けるか、②クロス参照軸を表示してパス歩行を続けるか、の選択を促す、という機能である。これによりユーザに有益なブラウジング情報表示の機会を与えたい。本稿ではこれらのパターン処理に焦点を置いたパス歩行とフォーム生成を検討する。

2. ポップアップ・ノード機能

本稿の理解のため、重複するが、本第2章の冒頭ではあえて既発表の内容の一部を再掲する。図-1は5個のノードを持つ H_3 型データソース[1]の簡易表現である。ノードのシンボルはノード自体を、矢線はノード間の1対多結合を表す。グラフ理論でいえば、この例は木構造をなすが、幾つ

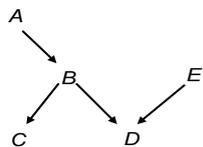


図-1 データソース例

★適用イメージ1

A: 部署, B: 社員, C: 扶養家族, D: 配属プロジェクト, E: プロジェクト

★適用イメージ2

A: 図書, B: 図書現品, C: 状態, D: 貸出, E: 利用者

*名古屋経済大学経営学部教授

もの適用イメージ [右に例示] が投影可能な構造である。

図-1 のノード E をエントリノードとし全ノードを渡り歩く DB アクセスは、歩行対象が木構造のとき、パス歩行経路パターンが常に 1 個だけ得られる。その点に留意し、フォーム構造を表すテンプレート要素を X, Y, Z, \dots で表すことにより考察しよう。概念フォーム式では以下の囲み欄の記号を使用する。

アルファベットで表記されたノードの上の矢線記号
 → 1 対多結合を多の側に向けてパス歩行するアクセスで親子構造を作る。
 ← 1 対多結合を 1 の側に向けてパス歩行するアクセス [下の例示みよ]
 \underline{DB} (下線記号) ノード D に対し B は非正規により 1 対 1 行結合を生む。
 ${}^B\bar{C}$ (左肩記号) ノード C は B から 1 対多の結合のアクセスで得られる。

このフォームの概念的な展開イメージを図-3 に記す。 E から D への 1 → 多のパス歩行は E をヘッダ、 D をボディとする 2 階層構造 $E\bar{D} \in X\bar{Y}$ を作る。 D から B への多 → 1 の歩行は 1 対 1 の行結合によりビュー \underline{DB} を作るから、2 階層が維持され $E\underline{DB} \in X\bar{Y}$ となる。 B から A へのアクセスも、多 → 1 の歩行であるから同様 1 対 1 の行結合により \underline{DB} は \underline{DBA} へと成長し、2 階層も維持され、 $E\underline{DBA} \in X\bar{Y}$ となる。 残る B から C への 1 → 多の歩行は B の成分を内含する \underline{DBA} をヘッダ、 C をボディとする 2 階層構造 $\underline{DBA} \cdot {}^B\bar{C} \in Y\bar{Z}$ を作る。ところが \underline{DBA} は、ヘッダである E に対するボディであるから、フォームの全体構造は 3 階層の $E\underline{DBA} \cdot {}^B\bar{C} \in X\bar{Y}\bar{Z}$ となる (図-2)。

$E \in X$, $E\underline{DBA} \in Y$, ${}^B\bar{C} \in Z$ は、概念フォームに内在して 3 階層構造の各層をなすビュー成分である。このようなビュー成分を「**フォーム生成ビュー**」と呼ぶ。

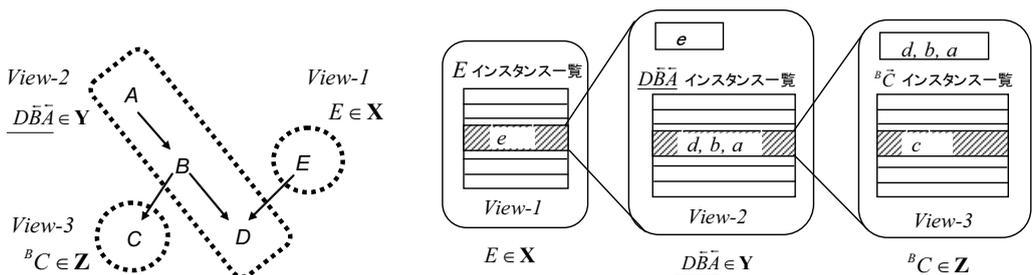


図-3 通常のパス歩行で得られた $E\underline{DBA} \cdot {}^B\bar{C} \in X\bar{Y}\bar{Z}$ のフォーム展開

図-2 図-1 の通常のパス歩行結果

以上が従来型のポップアップ無しのパス歩行だが、本稿では、図-2 のデータソース構造に、ポップアップ・サービスを装填できる機会がある点に大きく注目する。詳述する前に本稿での用語を整理しよう。あるノードまでパス歩行したとき、このノードと組み合わせてクロス参照軸を生成できるノードがあれば、そのことをユーザに提示して、①従来型のパス歩行を続けるか、②クロス参照軸を表示してパス歩行を続けるか、の選択を促すサービス機能を「**ポップアップ・ノード機能**」と呼ぶ。①の結果を「**ポップアップ無視**」、②の結果を「**ポップアップ選択**」、また①②の選択を促す局面を「**ポップアップ判断局面**」その結果を「**ポップアップ判断結果**」とそれぞれ呼ぶ。データソース構造によっては何度もポップアップ判断局面が現れるパス歩行もあるし、到達ノードによっては複数のポップアップ・ノードの選択肢が表示される場合もある。ここではポップアップ有りを

識別するシンボルをノード記号の右肩添字 \mathcal{J} で表す。ただしテンプレート要素では \mathcal{J} を表示しない。

図-2, 3 のパス歩行例はすべてのノードをポップアップ無視モードでパス歩行した結果だが、ポップアップ選択モードとした場合、このデータソースの E をエントリとする歩行では、 E そのものがポップアップ判断局面を作る。すなわち $B^{\mathcal{J}}$ との組み合わせを軸とする D へのクロス参照 $(E, B^{\mathcal{J}})\#D$ の生成をユーザに選択機会としてポップアップ表示できる。ユーザがこれを選択したあとのパス歩行は、 $B^{\mathcal{J}}$ を再開ノードとする未読の A および C への歩行を残すのみとなり、 $B^{\mathcal{J}}\bar{A}^{\mathcal{J}}\bar{C} \in X\bar{Y}$ を生成する (**図-4**)。

ところで上のクロス参照は以下の2つの表示方式がある。

α : 個別的クロス参照表示

初めに E を一覧表示したあと、特定のインスタンス $e \in E$ の選択結果に対し $B^{\mathcal{J}}$ のインスタンス全件 $\forall B^{\mathcal{J}}$ [非制限タイプ] —— もしくは E と D を共有する $B^{\mathcal{J}}$ インスタンス群 $e^{\bar{D}}\bar{B}^{\mathcal{J}}$ [制限タイプ] —— を表示し、両者を対照軸とする D へのクロス参照というパス歩行となる。 $\forall B^{\mathcal{J}}$ と $e^{\bar{D}}\bar{B}^{\mathcal{J}}$ を $B^{\mathcal{J}}$ で代表させると概念フォーム式は $E \cdot (e, B^{\mathcal{J}})\#e^{\bar{D}}\bar{D}$ となる。ただしいずれも、 E の軸側は e の1件のみのインスタンス表示となるため E 軸の次元が落ちて $B^{\mathcal{J}}$ 軸のみとなる。その結果、表示形式は E と結合する D インスタンスを子として持つ $B^{\mathcal{J}}$ のみを親とする親子イメージに帰着する。よって、概念フォーム式は $E \cdot B^{\mathcal{J}} \cdot e^{\bar{D}}\bar{D}$ (式1) となる。

β : 全般的クロス参照表示

E と上記 $B^{\mathcal{J}}$ のインスタンス群を各軸とする D のクロス参照結果を表示する表示方式。概念フォーム式は $(E, B^{\mathcal{J}})\#E^{\bar{D}}\bar{D}$ (式2) となる。

α 、 β とも有益なブラウジング情報表示の方式だが、いずれがユーザに使いやすいかは形式のみでは議論できないので、ユーザ選択できる実装方法が適当であろう。

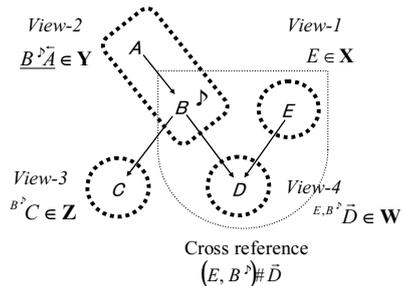


図-4 ポップアップ型パス歩行

注) 個別的クロス参照表示の方式ではクロス参照軸の次元落としが発生するが、文中で指摘の2つのタイプに対応する概念フォーム式は以下となる。

非制限タイプ : $E \cdot (e, B^{\mathcal{J}})\#e^{\bar{D}}\bar{D} = E \cdot (e, \forall B^{\mathcal{J}})\#e^{\bar{D}}\bar{D}$

制限タイプ : $E \cdot (e, B^{\mathcal{J}})\#e^{\bar{D}}\bar{D} = E \cdot (e, e^{\bar{D}}\bar{B}^{\mathcal{J}})\#e^{\bar{D}}\bar{D}$

ただし非制限と制限を問わず、いずれのタイプにおいても表示されるフォームは、データソース構造を問わず同じテンプレート構造となる [詳述省略]。

前半のパス歩行の2つのクロス参照方式 α 及び β による結果 (式1および2) を、後半のパス歩行結果の $B^{\mathcal{J}}\bar{A}^{\mathcal{J}}\bar{C} \in X\bar{Y}$ と統合したフォームの概念フォーム式は、それぞれ $E \cdot B^{\mathcal{J}}\bar{A}^{\mathcal{J}} \cdot (B^{\mathcal{J}}\bar{C}^{\mathcal{J}}, e^{\bar{D}}\bar{D}) \in XY(\bar{Z}, \bar{W})$ 、および、 $(E, B^{\mathcal{J}}\bar{A}^{\mathcal{J}}\bar{C})\#e^{\bar{D}}\bar{D} \in (X, Y\bar{Z})\#\bar{W}$ となる。**図-5, 6** にそれぞれのフォームイメージを示そう。いずれのポップアップ型のパス歩行も、従来の非ポップアップ型パス歩行で生成されたフォーム構造 $X\bar{Y}\bar{Z}$ と異なるが、これはポップアップ選択の結果、パス歩行経路が変容したためである。ただし、これらは既知のフォーム生成パターンとは異なるものの、フォームのテンプレート構造が既知のものと同じである点に注目したい。つまりユーザに対し近接ノードをポップアップ表示する機能は新たなサービスの開発対象だが、得られたデータ・インスタンス

を割り付けるフォーム・テンプレートは既知の構造が使える。これが、ここでの知見である。紙数の関係で詳述を避けるが、このことは他のデータソース・タイプにも等しく適用できる点である。

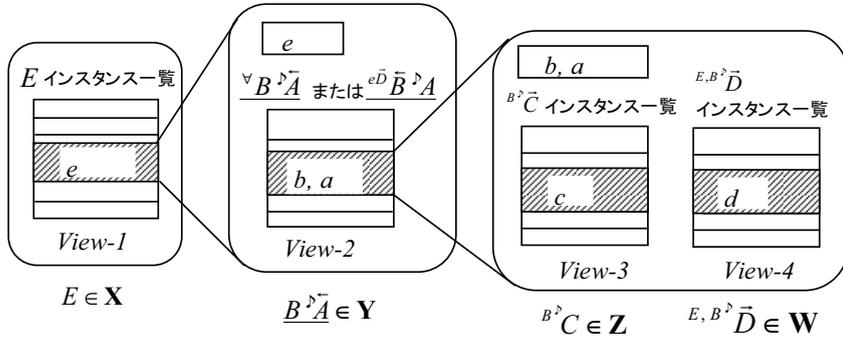


図-5 個別的クロス参照表示の方式によるパス歩行で得られた $E \cdot \underline{B^D A} \cdot (\overset{B^D}{C}, \overset{E, B^D}{D}) \in XY(\bar{Z}, \bar{W})$ のフォーム展開

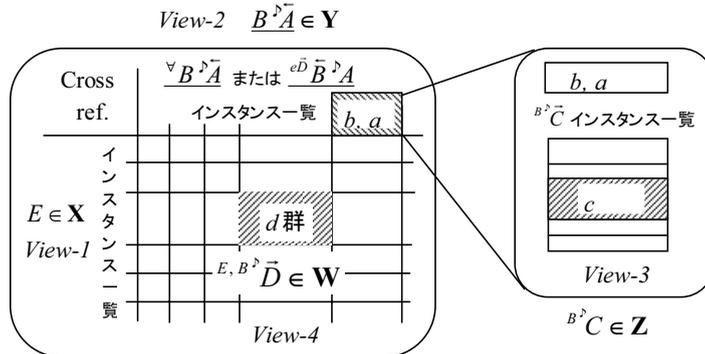


図-6 全般的クロス参照方式によるパス歩行で得られた $(E, \underline{B^D A}, \overset{B^D}{C}) \# \overset{E, B^D}{D} \in (X, Y\bar{Z}) \# \bar{W}$ のフォーム展開

3. ポップアップ型歩行とパス歩行行列

ノードの結合関係を隣接行列で表すパス歩行行列は概念フォーム式と等価の記述方式である[2].

パス歩行の前件側ノード (P) を列, 後件側ノード (Q) を行とし, ノード間の結合が実在してパス歩行の経路可能性がある行列成分を $[P]Q$ の式で, それ以外のものを 0 で表す隣接行列を, 「**パス歩行行列**」と呼ぶ。

この行列を隈なく歩くための従来の歩行ルールを**非ポップアップ型ルール**として再掲しよう[4].

パス歩行ルール (非ポップアップ型)

- i) 次に続くノードは, かならず読み進む。
- ii) 読み終えたノードと, 対称位置のノードは, 行列のノードの値を 0 にする。
- iii) 複数のノードに対し同じ方向から読めるものは, 同時に同じ方向から読み進む。
- iv) 0 行列を得たら妥当な歩行である。

図-1のデータソースのパス歩行行列は図-7のようになる。

$$\begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 & [E]\bar{D} \\ 0 & 0 & 0 & [D]\bar{E} & 0 \end{pmatrix}$$

図-7 図-1のデータソースのパス歩行行列

アクセスされたノードはパス歩行行列から抜き出され概念フォーム式の成分に順次組み込まれる。例えばノードEをエントリとする図-1への従来型〔非ポップアップ型〕のアクセスは、図-8のごとく概念フォーム式を成長させ、パス歩行行列はO行列に向けて縮退する。すなわち前件ノードに対応する列中の非0成分を見つけたら概念フォーム式に転記し、該当成分とその対称成分を0に変更する。この作業はパス歩行行列がO行列になったら完了し、それとともに概念フォーム式 $E\bar{D}\bar{B}(\bar{A}, \bar{C})$ を得る。

図-8の上段の最終ステップにあるこの式は、まだノード間の多→1関係がもたらす1対1の行結合の処理が施されていない。これを下段のように変換して最終の概念フォーム式を得ると三階層構造になる。

$$E \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 & [E]\bar{D} \\ 0 & 0 & 0 & [D]\bar{E} & 0 \end{pmatrix} \Rightarrow E\bar{D} \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \Rightarrow E\bar{D}\bar{B} \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & 0 & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \Rightarrow E\bar{D}\bar{B}(\bar{A}, \bar{C})$$

$$E\bar{D}\bar{B}(\bar{A}, \bar{C}) \Rightarrow E\bar{D}\bar{B}\bar{A} \cdot \bar{B}\bar{C} \in X\bar{Y}\bar{Z}$$

図-8 図-7への非ポップアップ型のパス歩行行列の適用

以上が非ポップアップ型のパス歩行のパス歩行行列の適用原理だが、この原理をそのままポップアップ型に適用しようとすると立ち行かなくなる。同じくEをエントリノードとする全般的クロス参照表示の例で見てみよう(図-9)。エントリEを認識したときの、Eの対応軸であるBは、いかなる方法で見つけることが出来るだろうか。それはEの次に到達するDをクロス参照する別の親を探す操作となる。そのためには最初にDまで歩行しなければならないが、先の非ポップアップ型ルールを適用すると、この時点でパス(E,D)に対応する行列成分が0に変更され、EとBをクロス参照軸とするDへのクロス参照への道が閉ざされてしまう。その結果、パス歩行は異常終了する。

$$E \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 & [E]\bar{D} \\ 0 & 0 & 0 & [D]\bar{E} & 0 \end{pmatrix} \Rightarrow E \text{ with } B^{\rightarrow} \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \Rightarrow (E, B^{\rightarrow})\# \bar{D}^{\rightarrow} \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

⇒ abnormal end. Dを共有する親ノードを行成分から見つける。

図-9 図-7へのポップアップ型に従来ルールを適用した場合のパス歩行の異常終了

そこで本稿では以下の補足条項のルール追加を提案する。

<p>補足条項 1 正式のパス歩行をする前に、ポップアップ・ノード有無を探索するパス歩行を実施する。 次に歩行するノードと行成分を共有する親成分があればポップアップ親ノードである。</p> <p>補足条項 2 ポップアップ・ノード探索のパス歩行では 0 に置き換える代わりに \wp の接頭辞を打ち、 当該ノードの親となるポップアップ・ノードをクロス参照対応軸ノードとして記録する。</p>

補足条項を適用したパス歩行を**図-10**に示す。当該ノードの親となるポップアップ・ノード B^{\flat} をクロス参照対応軸ノードとして記録したら、ただちに通常のカロス参照のステップを実施し、 \wp の接頭辞が打たれたノード D へのパス成分とその対称成分は正しく 0 に置き換えられる。このように修正条項を適用することで、ポップアップ・ノードの探索を行いながらパス歩行を最終ステップまで完遂することができる。なおポップアップ・ノードを検出したときに該当パス成分を 0 に置き換える必要はない。

$$\begin{aligned}
 E \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 & [E]\bar{D} \\ 0 & 0 & 0 & [D]\bar{E} & 0 \end{pmatrix} &\Rightarrow E \text{ with } B^{\flat} \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ \wp [B]\bar{D} & 0 & 0 & \wp [E]\bar{D} & 0 \\ 0 & 0 & 0 & \wp [D]\bar{E} & 0 \end{pmatrix} \Rightarrow (E, B^{\flat})\# \bar{D} \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & 0 & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
 \Rightarrow (E, B^{\flat}(\bar{A}, \bar{C}))\# \bar{D} &\Rightarrow (E, B^{\flat}(\bar{A}, B^{\flat} \bar{C}))\#^{E, B^{\flat}} \bar{D} \in (X, Y\bar{Z})\# \bar{W}
 \end{aligned}$$

D を共有する親ノードを行成分から見つける。

図-10 補足条項の適用による図-7 へのポップアップ型のパス歩行

データソース構造によっては複数のクロス参照対応軸からなるポップアップ・ノード群を検出する場合がある。**図-11**は、 W_5 型データソースにおけるノード C をエントリとするポップアップ型のパス歩行が複数のポップアップ・ノード A^{\flat} と E^{\flat} を検出する様子を示している。

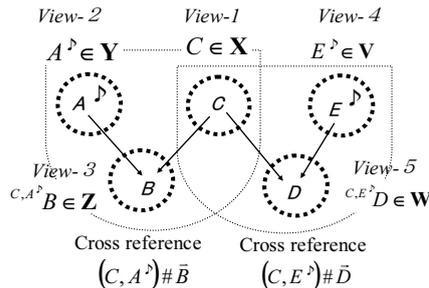


図-11 W_5 型データソースにおける複数ノードをポップアップするパス歩行

このパス歩行を仔細に検討しよう。エントリ C を認識した時点で二つのノード A^{\flat} と E^{\flat} がともに C のクロス参照対応軸として認識され、それぞれが $(C, A^{\flat})\# \bar{B}$ および $(C, E^{\flat})\# \bar{D}$ というクロス参照表を生成することになる。これら 2 つのクロス参照表を結合した概念フォーム式は、 $C \cdot \{(\hat{C}, A^{\flat})\#^{C, A^{\flat}} \bar{B}, (\hat{C}, E^{\flat})\#^{C, E^{\flat}} \bar{D}\}$ となる。ここでの \hat{C} はエントリノード C を軸として借り受けるとい

う意味だが、具体的なフォーム生成は、 C を共通軸とし、これを相手として A^{β} および E^{β} のクロス参照対応軸をそれぞれ結合させた —— C を折り目として A^{β} と E^{β} の2枚屏風を広げたようなイメージとなる。

この場合も上で述べたパス歩行行列の補足条項が適用できることを図-12に示そう。処理ステップの最初の行列が所与のパス歩行行列である。

$$C \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & 0 & 0 \\ 0 & [B]\bar{C} & 0 & [D]\bar{B} & 0 \\ 0 & 0 & [C]\bar{D} & 0 & [E]\bar{D} \\ 0 & 0 & 0 & [D]\bar{E} & 0 \end{pmatrix} \Rightarrow C \text{ with } A^{\beta}, E^{\beta} \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & \emptyset [C]\bar{B} & 0 & 0 \\ 0 & \emptyset [B]\bar{C} & 0 & [D]\bar{B} & 0 \\ 0 & 0 & \emptyset [C]\bar{D} & 0 & \emptyset [E]\bar{D} \\ 0 & 0 & 0 & [D]\bar{E} & 0 \end{pmatrix}$$

$\Rightarrow C \cdot \{(\bar{C}, A^{\beta}) \#^{C, A^{\beta}} \bar{B}, (\bar{C}, E^{\beta}) \#^{C, E^{\beta}} \bar{D}\}$
 D を共有する親ノードを行成分から見つける。

図-12 補足条項の適用による W_5 型データソースへのポップアップ型のパス歩行

上のパス歩行で生成される個別的クロス参照表示によるフォームイメージを図-13に示そう。有益なブラウジング情報表示方式である。フォーム展開はエントリノード C を折り目として A^{β} と E^{β} の2枚の屏風を広げたイメージとなっている。この表示方式では、 $C \cdot \{(\bar{C}, A^{\beta}) \#^{C, A^{\beta}} \bar{B}, (\bar{C}, E^{\beta}) \#^{C, E^{\beta}} \bar{D}\}$ の各クロス参照部分で C 軸の次元落としが発生し、その結果、帰着したフォームは、 $C \cdot \{A^{\beta, C, A^{\beta}} \bar{B}, E^{\beta, C, E^{\beta}} \bar{D}\} \in X(\bar{Y}\bar{Z}, \bar{V}\bar{W})$ のテンプレート構造となる。

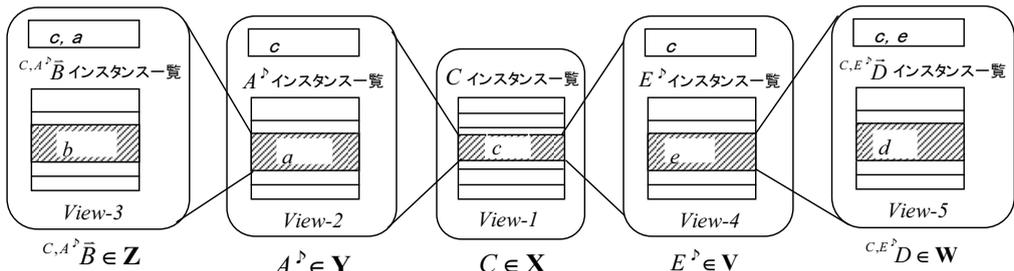


図-13 W_5 型データソースにおける個別的クロス参照表示によるパス歩行で得られた $C \cdot \{A^{\beta, C, A^{\beta}} \bar{B}, E^{\beta, C, E^{\beta}} \bar{D}\} \in X(\bar{Y}\bar{Z}, \bar{V}\bar{W})$ のフォーム展開

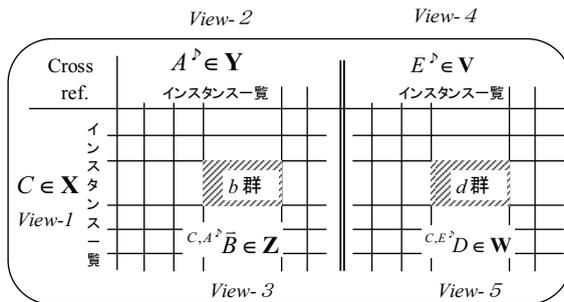


図-14 W_5 型データソースにおける全般的クロス参照表示によるパス歩行で得られた $C \cdot \{(\bar{C}, A^{\beta}) \#^{C, A^{\beta}} \bar{B}, (\bar{C}, E^{\beta}) \#^{C, E^{\beta}} \bar{D}\} \in X \cdot ((\bar{X}, \bar{Y}) \# \bar{Z}, (\bar{X}, \bar{V}) \# \bar{W})$ のフォーム展開

同じパス歩行で生成される全般的クロス参照表示によるフォームイメージを**図-14**に示そう。これも有益なブラウジング情報表示方式である。エントリノード C を折り目とする A^{\flat} と E^{\flat} の2枚の屏風イメージという点は同じだが、この場合の表示方式では、 $C \cdot \{(\hat{C}, A^{\flat}) \#^{C, A^{\flat}} \bar{B}, (\hat{C}, E^{\flat}) \#^{C, E^{\flat}} \bar{D}\}$ の各クロス参照部分で C 軸の次元落としが発生せず、 A^{\flat} と E^{\flat} の軸が連結され擬似的に1軸となって C 軸とクロス参照するイメージとなる。このフォームのテンプレート構造は、 $C \cdot \{(\hat{C}, A^{\flat}) \#^{C, A^{\flat}} \bar{B}, (\hat{C}, E^{\flat}) \#^{C, E^{\flat}} \bar{D}\} \in \mathbf{X} \cdot ((\bar{X}, \bar{Y}) \# \bar{Z}, (\bar{X}, \bar{V}) \# \bar{W})$ となる。

4. 考察

ポップアップ・ノードの表示は、V字 (V_3 型) もしくは逆V字 ($'V_3$ 型) のサブグラフを内含する全てのデータソースのパス歩行で提供の機会がある。このようなサブグラフ構造は、筆者がかつて網羅的に整理した50個のデータソース・タイプ基本形[1] (**図-15**)のほとんどのタイプが内含している。本稿で取り上げたポップアップ・ノード例のどれもが優れたブラウジング情報表示であった結果と考え合わせれば、ポップアップ・サービスは開発に旨みのある近接ノード示唆機能であると評価できよう。

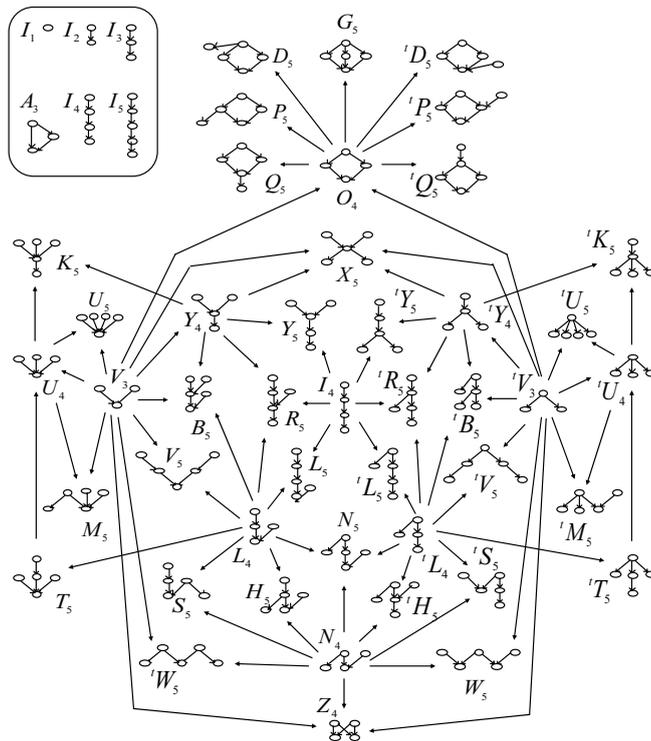


図-15 データソース・タイプの基本形

参考文献

- [1] 中西昌武, “データソース・タイプ, 概念帳票生成パターン, および概念帳票テンプレートのカタログ化”, 経営情報学会誌, 11巻, 1号, 2002.6.
- [2] M. Nakanishi. “Office Form Generator Supported with Path Reading Matrix”, *Economic and Business Review*, Vol.15, No.1, Nagoya Keizai University, 2007.9.
- [3] 中西昌武, “概念帳票における View 構造の行列表現とその応用”, 経営情報学会 2010 年春季全国研究発表大会予稿集, 2010.6.
- [4] 中西昌武, “パス歩行行列の仮定のノード及びパスへの拡張”, 情報システム学会誌, Vol.11, No.1, 2016.1.
- [5] 中西昌武, “パス歩行行列における非連結と閉路の検出法”, 経済経営論集, Vol.23, No.1, 2016.1.
- [6] その他の文献は, 上記文献のなかで記した参考文献を参照のこと。