

パス歩行行列の仮想定のノード及びパスへの拡張

中西 昌武

要 旨

パス歩行行列は、帳票・伝票・画面フォームなどのフォーム生成の標的となるデータソースにおけるデータノード間の1対多の結合を、グラフ理論の隣接行列で表したものである。この行列を使ってDBアクセスの歩行経路の証跡を消しこむと、生成可能なフォームパターン群が洗い出せ、種々のフォームのイメージ提示による要求分析が支援できるようになる。要求分析者がデータソースを構成するとき、しばしば実データを持つ実相ノード以外に仮想定のノードやパスを積極的に用いて議論するが、これらを同時に扱いつつ適切なフォーム生成が行えるためのパス歩行の理論は未整備である。本稿では、スキン識別とサブビュー概念の新たな導入により、これらを適切に扱うパス歩行の方法を提案すると共に、それを前提としたフォーム生成の表示類型を示し、要求分析の環境作りに理論面から貢献する。

Abstract

Path walking matrix is an adjacency matrix which maps every walk on one-to-many relationship path between each pair of data nodes on target data source for generating actual business forms. Through walking this, a set of generative form patterns are obtained, which gives requirement analysis various form images. In constituting data source, requirements analysts often put makeshift nodes and paths for active discussions, but method to treat those "virtual" elements is not in the present scope of our theory. This paper proposes an enhanced theory of walking and form generation method with virtual nodes and paths, where we apply new concepts "skin identification" and "subview" for establishing a set of form presentation types derived with real/virtual skin elements.

1. はじめに

要求分析の担当者(「**分析者**」と略称)は、実際の帳票・フォーム(「**フォーム**」と略称^{注)})の使用局面を描き、大胆な使用をも試して手応えある要求仕様を模索する。データソースとユーザ・アクセスに関する既往研究

[1-4]を踏まえ、筆者はデータソースとフォームの構造をグラフ理論の隣接行列(「**パス歩行行列**」と呼ぶ)を用いて数理科学的に連動させ、フォーム自動生成も扱える概念フォームモデル[5-7]を開発した。これによればデータソースから生成可能なフォーム構造パターンは方程式を解くように理論的に特定できるので、実装したフォーム分析ツールは、カーナビゲーションや列車検索エンジンと似た操作イメージの「**フォーム生成シミュレータ**」となる。ツール利用者は、理論的に定義された網羅的体系性の中でフォーム生成の感触を確かめ要求分析に

Enhanced Path Walking Matrix with Virtual Node and Path

Masatake Nakanishi

名古屋経済大学

Nagoya University of Economics

[論文]

2015年 8月 30日 受付

2015年 11月 29日 受理

© 情報システム学会

注) 筆者はこれまで「帳票」と呼んできたが、英語は form を用いてきた。「帳票」は紙出力媒体のイメージが強いとの指摘があったので、今後は「フォーム」と呼ぶことにする。

勤しめる。ここには理論に基づき経験と技芸を活かす工学的世界へのいざないがある。

インスタンス登録を予定した個々のデータの器をデータノードと呼ぶ(「ノード」と略称)。パス歩行行列はノード結合の行列表現に過ぎず、結合可能なノードは全て扱えるので、データ登録済みの実体でなくとも良く、仮想定のポンチ絵的ノードでも構わない。そんな曖昧なノードやパスまで含めたフォーム生成シミュレータが実現すれば、理想と現実の狭間でしつこく仮説的に揉みこむ要求分析現場の助けとなるだろう。

ただしフォーム生成において実データを持つ**実相ノード**とそれ以外の**仮想定ノード**を同時に扱うには、両者を結合するフォーム構築技術が求められたため、仮想定ノードへの拡張は構想どまり[7]だった。本稿ではそのための理論拡張を提案し、実相と仮想定を問わず全てのノードとパスの組合せに応じたフォーム生成の表示類型を示す。

2 概念フォームとパス歩行行列

2.1. 概念フォームモデル

本稿の理解のため、重複するが、本第2章ではあえて既発表の内容の一部を再掲する。

実業務のフォームからレイアウト制約、ビジュアル効果、行数制限、表示媒体等の一切の物理的要素を捨象したデータ構造表現を「**概念フォーム**」、その数式表現を「**概念フォーム式**」と呼ぶ[3-7]。標的データソー

スの構造が特定できれば、アクセスする概念フォームの構造パターンは有限個に特定でき、標的が木構造の場合はエントリノードの特定によって唯一に決まる[3-5]。

図1は、5個のノードを持つデータソースの簡易表現である。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、幾つもの適用イメージ[右に例示]が投影可能な構造である。

このノードEをエントリとし木構造の全ノードを歩くデータソース・アクセスの経路パターンは唯一となる。その点に留意しフォーム構造のテンプレート要素をX,Y,Z...で表すことにして考察しよう。概念フォーム式では以下の記号を用いる[4]。

アルファベットで表記されたノードの上の矢線記号

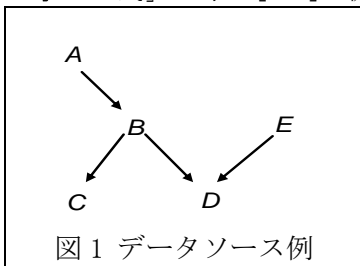
→ 1対多結合を多の側に向けてパス歩行するアクセスで親子構造を作る。

← 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は非正規化により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



★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: 状態履歴, D: 貸出, E: 利用者

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 $\bar{D}\bar{B}$ を作るから, 2階層が維持され $E\bar{D}\bar{B} \in X\bar{Y}$ 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 $\bar{D}\bar{B}$ は $\bar{D}\bar{B}\bar{A}$ へと成長し, 2階層も維持され, $E\bar{D}\bar{B}\bar{A} \in X\bar{Y}$ 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 $\bar{D}\bar{B}\bar{A}$ をヘッダ, C をボディとする2階層構造 $\bar{D}\bar{B}\bar{A} \cdot {}^B\bar{C} \in Y\bar{Z}$ を作る. ところが $\bar{D}\bar{B}\bar{A}$ 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 $E\bar{D}\bar{B}\bar{A} \cdot {}^B\bar{C} \in X\bar{Y}\bar{Z}$ となる.

$E \in X$, ${}^E\bar{D}\bar{B}\bar{A} \in Y$, ${}^B\bar{C} \in Z$ 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(図2). このようなビュー成分を「フォーム生成ビュー」と呼ぶ.

2.2. パス歩行行列

概念フォーム式の導出については, ノード間の結合関係を記述した隣接行列であるパス歩行行列の適用が確立している[5-7].

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のある行列成分を $[P]Q$ 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.

これはコンピュータ可読の方式である. この行列を隈なく歩くための歩行ルールを再掲しよう[4-7]. ivは完了基準である.

- 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) O 行列を得たら妥当な歩行である.

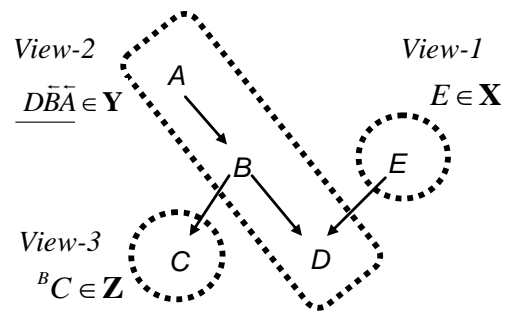


図2 図1のパス歩行結果

図1のデータソースのパス歩行行列は図4【式(3)】のようになる.

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へのアクセスは, 図5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

$$E\bar{D}\bar{B}(\bar{A}, \bar{C}) \quad (1)$$

を得る. 図5の最終行にあるこの式は, 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. これを以下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.

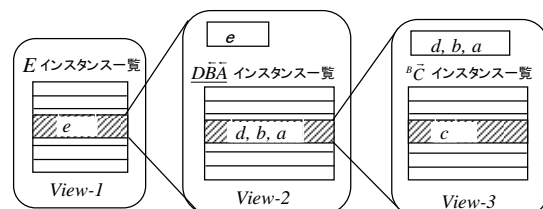


図3 パス歩行の結果, 得られた $E\bar{D}\bar{B}\bar{A} \cdot {}^B\bar{C} \in X\bar{Y}\bar{Z}$ のフォーム展開

$$E\bar{D}\bar{B}(\bar{A}, \bar{C}) \Rightarrow E\bar{D}\bar{B}\bar{A} \cdot \bar{B}\bar{C} \in X\bar{Y}\bar{Z} \quad (2)$$

式(2)の $E\bar{D}\bar{B}\bar{A} \in Y$ のようなビュー成分は、これと等価のコンピュータ可読の隣接行列で表すことができ、これを用いてフォームを自動生成する機構が既にある[6,7]. 本稿の理論拡張も、これを前提としている.

$$\begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 & [E]\bar{D} \\ 0 & 0 & 0 & [D]\bar{E} & 0 \end{pmatrix} \quad (3)$$

図4 図1のデータソースのパス歩行行列

$$\begin{aligned} E \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 & [E]\bar{D} \\ 0 & 0 & 0 & [D]\bar{E} & 0 \end{pmatrix} \\ \Rightarrow E\bar{D} \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} & \Rightarrow E\bar{D}\bar{B} \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & 0 & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\ & \Rightarrow E\bar{D}\bar{B}(\bar{A}, \bar{C}) \end{aligned} \quad (4)$$

図5 図2へのパス歩行行列の適用

3. 仮定定のノード及びパス

3.1 仮定定のノード及びパスの扱い

本章では、新たに仮定定のノード及びパスの扱い方を提案する. パス歩行行列は、ノード同士の結合関係の隣接行列であり、結合可能なものは全て扱える. ノードは、ヘテロ

なDB実体でも良く、「そのようなデータ実体として仮想定されたノード」でも構わない. また、いかなるメディア形式でも構わないし、正規形であるかどうかは問わない.

図6のCase-1のデータソースは、4つのノード(A:部署, B:社員, C:連絡先メールアドレス, D:取引先)からなる. ノードB及びDの多対多関係は顧客に社員を貼り付ける分析者の意図の反映だが、要求分析はまだ粗く、今後の分析の成り行きは、BとDの双方を親とする新たな関連性ノードの解明に至るのか(Case-2), それがいかに複数のノードに分解されるのか(Case-3), 結局、単なる1対多関係に落ち着くのか(Case-4), がらりと変わり顧客に社員を貼り付ける考えを捨てて部署総括の管理事項とするか(Case-5), なんとも予測のつかない曖昧状況だが、とりあえずCase-1から考えてみよう. BとDの多対多関係はパスのスキーマの不備を伺わせる.

この場合のパス歩行行列は、図7のように多対多関係をそのまま行列に反映した状態でフォーム生成結果を分析者に提供し、要求分析の判断材料に資することができる. パス歩行行列のパス(B, D)の対称成分 $[B]\bar{D}$, $[D]\bar{B}$ の矢印は、通常と異なり互いに反転せず、同じ右向きとなる. これはフォーム生成するとき、BとDのどちらも相手ノードの親となりうる仕様を意味する.

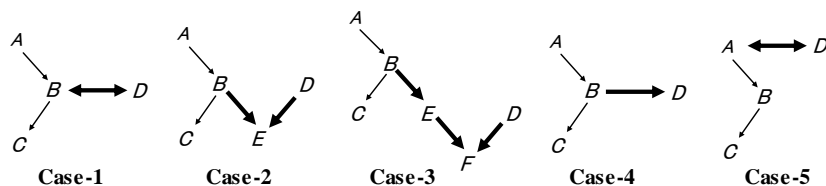


図6 サンプル・データソースの検討ケース

$$A \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} \\ 0 & [B]\bar{C} & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 \end{pmatrix} \quad (5)$$

図 7 Case-1 に対応するパス歩行行列

$$A \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} \\ 0 & [B]\bar{C} & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 \end{pmatrix} = A\bar{B} \cdot \begin{pmatrix} 0 & \mathbf{0} & 0 & 0 \\ \mathbf{0} & 0 & [C]\bar{B} & [D]\bar{B} \\ 0 & [B]\bar{C} & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 \end{pmatrix} \\ = A\bar{B}(\bar{C}, \bar{D}) \quad (6)$$

ノード A をエントリとするパス歩行は式 (6) のように三階層フォームを生成し、その中で D は C とともに B の子ノードとなる。

ノード D をエントリとする場合はどうか。 B は D の子ノードとなるが、 $D\bar{B}(\bar{A}, \bar{C})$ の \bar{A} は B に対する多→1 の行結合により吸収されてビュー $\bar{B}\bar{A}$ を作り、 $\bar{B}\bar{A}$ は B 成分を結合点として子ノード C と結ぶから、以下の式 (7) のような三階層のフォーム構造となる。

$$D \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} \\ 0 & [B]\bar{C} & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 \end{pmatrix} = D\bar{B} \cdot \begin{pmatrix} 0 & [B]\bar{A} & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & \mathbf{0} \\ 0 & [B]\bar{C} & 0 & 0 \\ 0 & \mathbf{0} & 0 & 0 \end{pmatrix} \\ = D\bar{B}(\bar{A}, \bar{C}) = D\bar{B}\bar{A} \cdot \bar{B}\bar{C} \in XY\bar{Z}. \quad (7)$$

このように多対多の曖昧さを含むデータソースであっても、パス歩行で得た概念フォーム式を見れば、どのようなフォーム構造が生成可能かは数理的に確認できる。

3.2 スキン識別及びサブビューの概念

以上のようにフォーム構造の解析は、ノードやパスにスキーマ定義が無くとも行える。ただしパス歩行結果を実際のフォームイメージに展開するときは注意が必要である。

今 **Case-1** の D がスキーマ未定義で、ノード名と若干のノード説明しか無いまま、スキーマ定義済みのノード B とパス結合しているとする [**Case-1.1**]。両者のパス認識は漠然としており結合仕様も未定義である。この状態で A をエントリとするパス歩行結果のフォームを生成しようとする、通常のスキーマ操作言語の処理だけでは B から D への結合の実現方法で躓くことになる。

そこで本稿では「**スキン識別**」に基づく「**サブビュー**」概念の導入を提案する。インスタンス操作言語の扱い可否を識別するノードやパスの帰属属性を「スキン」と呼び、扱えるものに **Real**、扱えないものに **Virtual** の識別値を与えることにする。
Real/Virtual は便宜的な呼称である。仮想ノードのスキンは **Virtual** である。またノード間のインスタンス連結用のパス属性のあるパスのスキンは **Real**、そうでないスキンは **Virtual** である。**Virtual** スキンのノードを連結相手に含むパスのスキンは必然的に **Virtual** となる。**Real** スキンのノード同士のパスであってもインスタンス連結を定義する属性が未設定のパスのスキンは **Virtual** である。**Case-1.1** では、ノード B のスキンは **Real**、 D のスキンは **Virtual**、パス (B, D) のスキンは **Virtual** である。

データノードに以下の3属性を与える。

- (i) 「**ノード・スキン識別**」 上述の通り。
- (ii) 「**ノード説明**」 ノードの特徴及び運営にまつわる諸事項を記載する。
- (iii) 「**スキーマ操作言語**」 スキーマ操作言語 (SQL, Xquery など) を記載する。独自の操作言語も登録可能とする。

パスについても「**パス・スキン識別**」の属性を与える。上のスキン識別に基づく「**サブ**

ビュー」概念を以下のように定義する。

フォーム生成ビューにあって、同じスキンを持つ連結ノードからなる成分を「**サブビュー**」と呼び、以下の仕様で与える。
 スキーマ操作命令でインスタンス結合可能な極大区画の Real ノード連結を **Real サブビュー**、Virtual スキンの単体ノードを以て **Virtual サブビュー** と定義する。

親子フォームは子側に3つの対応がある。

- ① (☀) 親子各サブビューおよび親子パスのスキンが全て Real の場合、親が子のインスタンスを特定できるため通常のスキーマ操作で子インスタンスを表示すれば済む。
- ② (☘) 親ノードや親子パスが Virtual だったり、親の前件側の事情で親を特定できない、などの理由で子インスタンスを特定できない場合、結合候補の子インスタンス集合と状況説明の注記を分析者に提示する。
- ③ (☂) 子のスキンが Virtual の場合は子インスタンスが存在しないので、ノード説明と状況説明の注記を分析者に提示する。

表 1 は上の 3 対応から導出した親子フォームの類型である。フォーム構造の基本形は同じだが、スキンの違いで親子フォーム各部の埋め込み内容やフォームへの注記が異なる。ただし分析者に結果を直視させ更なる分析へと促す表示意図は共通である。なお親と結合する子サブビューが存在しない場合は子：ボディ部は非表示となる。

Case-1 の各ノードをエン트리とする概念フォーム式を表 2 に示した。ここでは Virtual ノード D を $D(v)$ 、それ以外の Real ノードを A, B, C とし、Virtual パスの矢線部に Δ を付した。またスキン識別に基づくフォーム表示の類型名を右列に示した。

B をエン트리とするフォームは図 8, 9 に示した。 B を親とする子ノードのうち Real スキンの C は、通常の親子フォームのインスタンス表示となる一方、Virtual スキンの $D(v)$ はノード説明と注記により要求分析に供するフォームイメージとしている。

B を仮想定から検討する場合は Virtual スキンの $B(v)$ となる。View-1 はスキンの異なるサブビュー [$subView-1.1 : B(v)$ 及び $subView-1.2 : A$] の連結となり、図 9 のインスタンス一覧が表示できないので、表 1 の親子類型を以下のように援用する (図 10, 11)。 $B(v)$ のノード説明と注記を表示すると共に B から A にリンクを渡す。 A の側では、やがて B のスキンが Real になったとき 1 対 1 結合する候補となる A の全インスタンスを一覧表示し、その旨を注記する。その上で B の 2 つの子ノード C 及び $D(v)$ の親子フォームの表示に繋ぐ。

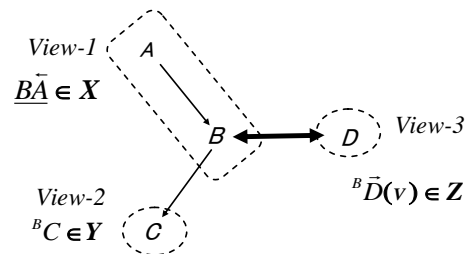


図 8 Case-1 のデータソースの B をエン트리とするパス歩行結果

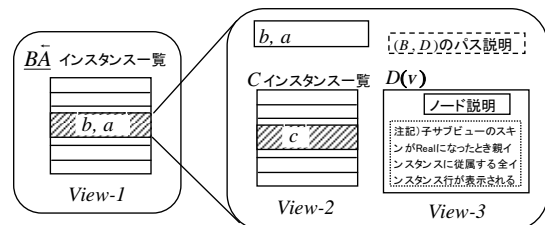


図 9 パス歩行の結果、得られた $BA \cdot ({}^B C, {}^{AB} D(v)) \in X(Y, Z)$ のフォーム展開

表 1 Real/Virtual スキンを含む親子フォームの表示類型

<p>【親の類型 P1】 親が Real サブビューである場合は, P1a, P1b のようにする.</p> <p>【親の類型 P1a】 親がフォーム生成ビューの先頭サブビューである場合 (親: ヘッダ部) 当該フォームで呼び出された親インスタンスを表示する.</p> <p>【親の類型 P1b】 親がフォーム生成ビューの先頭サブビューでない場合 (親: ヘッダ部) 親の前件サブビューまたは前件からのパスのスキンが Virtual のため親の前件と行結合すべき親インスタンスが特定できない. そこで, 当該サブビューを構成する全インスタンスを親インスタンス候補として一覧表示し「前件サブビューのスキンが Real になったとき親インスタンスが特定される」と注記する. その上で, 親インスタンスが特定できた暁に従属する子インスタンス一覧を表示する別フォームにリンクする.</p> <p>【親の類型 P2】 親が Virtual サブビューである場合 (親: ヘッダ部) 親インスタンスが存在しないので, 親のノード説明をフォーム構造に添え, 「当該サブビューのスキンが Real になったとき親インスタンスが特定される」と注記する.</p>
<p>【子の類型 C1】 親子双方が Real サブビューである場合は, C1.1, C1.2 のようにする.</p> <p>【子の類型 C1.1】 親子間のパスが Real スキンである場合は, C1.1a, C1.1b のようにする.</p> <p>【子の類型 C1.1a】 親の類型が P1a である場合 (☀) (子: ボディ部) 親インスタンスに従属した子インスタンス一覧を表示する.</p> <p>【子の類型 C1.1b】 親の類型が P1b である場合 (☼) (子: ボディ部) P1b 表示の親インスタンスに従属する子インスタンス一覧を表示する.</p> <p>【子の類型 C1.2】 親子間のパスが Virtual スキンである場合 (☾) (子: ボディ部) 親インスタンスに従属する子インスタンスが特定できないので, 子サブビュー全てのインスタンス行を表示し, 「親子パスのスキンが Real になったとき実際の親子関係を満たす行のみが選択表示される」と注記する.</p> <p>【子の類型 C2】 親が Real サブビュー [P1a, P1b] で子が Virtual サブビューの場合 (☯) (子: ボディ部) 親インスタンスに従属する子インスタンスが存在しないので, 子のノード説明をフォーム構造に添え, 「子サブビューのスキンが Real になったとき親インスタンスに従属する全インスタンス行が表示される」と注記する.</p> <p>【子の類型 C3】 親が Virtual サブビュー [P2] で子が Real サブビューである場合 (☾) (子: ボディ部) 親インスタンスに従属する子インスタンスが特定できないので, 子サブビュー全てのインスタンス行を表示し, 「親のスキンが Real になったとき実際の親子関係を満たす行のみが選択表示される」と注記する.</p> <p>【子の類型 C4】 親子双方が Virtual サブビューである場合 (☯) (子: ボディ部) 親同様, 子のノード説明をフォーム構造に添え, 「親子サブビューのスキンが Real になったとき, 実際の親子関係を満たすインスタンス行が表示される」と注記する.</p> <p>注), 「サブビューのスキンが Real になったとき」云々は, 読み手に配慮した表現が求められよう.</p>

表 2 Case-1 のデータソースの各ノードを
エントリとするパス歩行結果

エン トリ	概念フォーム式	フォーム 構造	Virtualスキンの親子パスにおける フォーム表現の種類
A	$\overline{AB}(\overline{C}, \overline{D}(v))$	$XY(\overline{Z}, \overline{V})$	$B^A \overline{D}(v)$ は、親:P1と子:C2
B	$B(\overline{A}, \overline{C}, \overline{D}(v)) = \overline{BA} \cdot (\overline{C}, \overline{AB} \overline{D}(v))$	$X(\overline{Y}, \overline{Z})$	$\overline{BA} \cdot \overline{AB} \overline{D}(v)$ は、親:P1と子:C2
C	$\overline{CB}(\overline{A}, \overline{D}(v)) = \overline{CBA} \cdot \overline{AB} \overline{D}(v)$	XY	$\overline{CBA} \cdot \overline{AB} \overline{D}(v)$ は、親:P1と子:C2
D(v)	$D(v) \cdot \overline{B}(\overline{A}, \overline{C}) = D(v) \cdot \overline{BA} \cdot \overline{C}$	XYZ	$D(v) \cdot \overline{BA}$ は、親:P2と子:C3

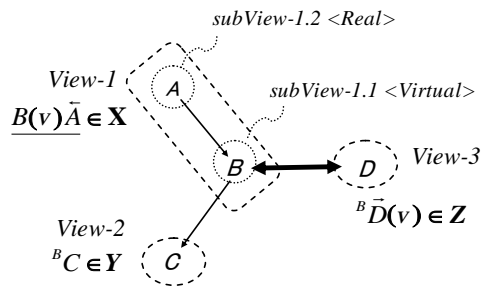


図 10 Case-1 の B(v) をエントリとするパス
歩行結果

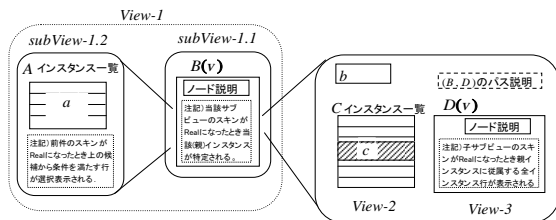


図 11 パス歩行の結果, 得られた
 $B(v) \overline{A} \cdot (\overline{AB} \overline{C}, \overline{AB} \overline{D}(v)) \in X(\overline{Y}, \overline{Z})$ のフ
ォーム展開

4. 議論と展望

本稿では実相と仮想定を問わず全てのノードとパスをパス歩行行列で扱う方法及びその親子フォームの表示類型を示した。これらのフォームは分析者の選択の解析結果として方程式を解くように与えられる。

要求分析はスキーマ定義を固める前に多様に行うことが望ましく、仮想定ノードやパスは分析者にとって馴染みある分析対象である。本稿の理論拡張で新たに扱えるようになった仮想定ノードの分析対象は実装制約

を受けず簡単にツール上で変更管理できるので、分析者がポンチ絵レベルのまま自由にデータソースの形態変化やパス歩行を味わいつつ自らの発想を確かめる要求分析の環境づくりに理論面から貢献できる。

ただし要求分析は多様な要因が複雑に錯綜する領域でありフォーム分析ひとつの理論化で全体が劇的に変わるものでもない。

この論点については稿を改めて議論したい。

実装実験の結果と動作イメージは、情報システム学会の「超上流工程における要求分析への科学的アプローチ」研究会の第1回(2015.5.25)例会で実機を用いた説明により報告したが、要求分析に勤しむ分析者の探究活動を促すために有効なユーザインタフェースの検討は今後の課題である。

参考文献

- [1] G. Santucci and L.Tarantino, "A Dynamic Form-Based Data Visualiser for Semantic Query Languages", *Proceedings of 2nd International Workshop on Interfaces to Database Systems*, 1994, pp.249-265.
- [2] A. Shoshani, "CABLE: A Language Based on the Entity-Relationship Model", *Lawrence Berkeley Laboratory*, no.22033, 1978.
- [3] 中西昌武, "概念帳票モデルによる帳票構造の代数的分析", *経営情報学会誌*, 6巻, 4号, 1998.5.
- [4] 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.
- [5] M. Nakanishi. "Office Form Generator Supported with Path Reading Matrix", *Economic and Business Review*, Vo1.15,

No.1, Nagoya Keizai University, 2007.9.

- [6] 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.
- [7] 中西昌武, “パス歩行行列を媒介した帳票検討プロセス”, 情報システム学会 2014 年全国大会予稿集, 2014.11.