

仮想定のノードとパスを扱うためのパス歩行行列の拡張と応用

Enhancing path walking matrix for handling virtual node and path

中西昌武[†]

Masatake Nakanishi[†]

[†]名古屋経済大学 経営学部

[†] Faculty of Business Management, Nagoya University of Economics.

要旨

パス歩行行列は、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り、帳票・フォーム構造を DB アクセスの歩行経路構造とみなすモデルである。帳票・フォームの論理構造は標的 DB の論理構造や業務の要求仕様と不可分の関係にあるが、本稿では、仮想定のノードやパスを前提とする「仮想定パス歩行行列」への理論拡張を提案する。仮想定のノードやパスと実データを持つノードやパスを連結するには実装制約の克服が不可欠だが、本研究では「Real/Virtual スキン識別」と「サブビュー」の導入によって克服した。本稿では仮想定のノードやパスを前提とするパス歩行方法、及び、それに基づくフォーム生成方法やフォーム検討の展開を提示する。

1. はじめに

要求分析の担当者、特にユーザは、実際の帳票・フォーム（「**フォーム**」と略称）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[1-3]。中西の**概念フォームモデル**[4-8]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）で表して数理科学的に連動させ、フォーム自動生成に足場を持つフォーム分析に道を開いた理論である。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、この理論を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。

パス歩行行列はデータノード（「**ノード**」と略称）結合の隣接行列表現に過ぎず、結合可能なノードなら全て扱える。従ってデータ登録済みの実体でなくとも良く、仮想定のポンチ絵のノードでも構わない。 そんな曖昧さで措定したノードも含めてフォーム分析できれば、理想と現実の狭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。仮想定ノード対応のツールが理論的に可能であることは[8]で主張したが、実装は構想段階に留まっていた。今回は理論的に補強した上での実装検証である。

フォーム構造を見るだけなら従来のパス歩行行列で十分可能だが、フォーム生成において実データを持つノードやパスと仮想定のノードやパスを同時に扱うには、実装制約を克服するフォーム構築技術が必要となる。 本研究では「**Real/Virtual スキン識別**」と「**サブビュー**」を導入したパス歩行行列の拡張によりこの問題を克服した。本稿は、このような機能を持つ「**仮想定パス歩行行列**」の提案である。

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

筆者が提唱する「**概念フォーム^{注)}**」は、帳票・フォームから物理的成分を除去して得た概念的なデータ表示構造であり、概念データモデルにアクセスする外部データの構造を概念的にモデル表現したものである。また「**概念フォーム式**」は、概念フォームの構造を数式で表現したものである。こ

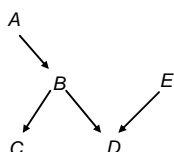


図-1 サンプル・データソース

★サンプルの適用イメージ 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

★サンプルの適用イメージ 2

A: 図書, B: 図書現品, C: 状態履歴, D: 貸出, E: 利用者

^{注)} 筆者はこれまで「帳票」と総称してきたが、英語は form を用いてきた。一部のシステム従事者から、「帳票」は紙出力された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」と総称することにする。

のモデルの定義と規則については前回大会予稿[8]の詳述に譲り、本第 2 章では基本的事項のみ触れる。

例で示そう。図-1 は、5 個のノードを持つデータソース構造の簡易表現である。各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、このサンプルは木構造をなすが、幾つもの適用イメージ [右に例示] が投影可能な構造となっている。

概念フォーム式では、以下の記号を使用する。

- 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]。
- $\underline{D\bar{B}}$ (下線記号) ノード D に対し B は非正規化により 1 対 1 の行結合を生む。
- ${}^B\bar{C}$ (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。

図-1 の E をエントリノードとする全ノードのパス歩行 (図-2) が作る概念フォームの展開イメージを図-3 に記す。 X, Y, Z はフォーム構造を示すテンプレート要素である。 E から D への 1 から多へのパス歩行は E の特定インスタンスをヘッダ、 D の特定インスタンス一覧をボディとする 2 階層構造 $E\bar{D}\bar{B} \in X\bar{Y}$ をひとまず作る。 D から B への多から 1 へのパス歩行は、 D と 1 対 1 の行結合 $\underline{D\bar{B}}$ を作るから、2 階層構造を維持し $E\underline{D\bar{B}} \in X\bar{Y}$ に成長する。 B から A へのアクセスも、多から 1 へのパス歩行のため 1 対 1 の行結合により 2 階層構造を維持し $\underline{D\bar{B}}$ は $\underline{D\bar{B}\bar{A}}$ へと再成長し $E\underline{D\bar{B}\bar{A}} \in X\bar{Y}$ となる。 残るノード C へは B からアクセスするが、これは 1 から多へのパス歩行であるから B の成分を内含する $\underline{D\bar{B}\bar{A}}$ をヘッダ、 C をボディとする 2 階層構造 $\underline{D\bar{B}\bar{A}} \cdot {}^B\bar{C} \in Y\bar{Z}$ をひとまず作る。 ところが $\underline{D\bar{B}\bar{A}}$ 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 $E\underline{D\bar{B}\bar{A}} \cdot {}^B\bar{C} \in X\bar{Y}\bar{Z}$ を作る。

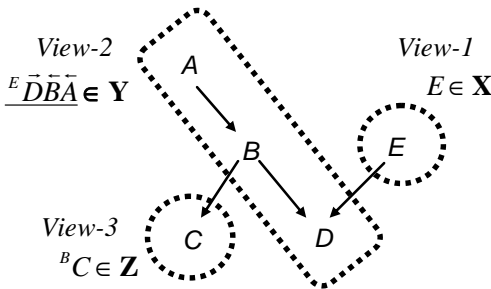


図-2 図-1 へのパス歩行結果

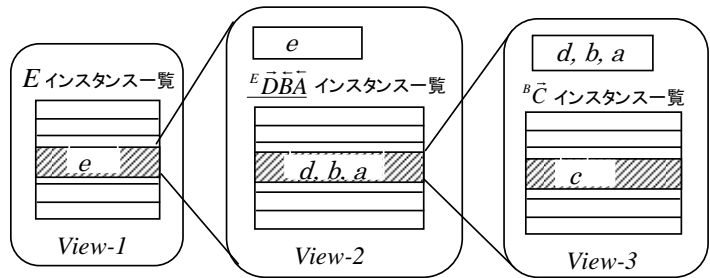


図-3 パス歩行の結果, 得られた $E\underline{D\bar{B}\bar{A}} \cdot {}^B\bar{C} \in X\bar{Y}\bar{Z}$ のフォーム展開

ここで $E \in X$, $E\underline{D\bar{B}\bar{A}} \in Y$, ${}^B\bar{C} \in Z$ は、それぞれ概念フォームに内在して 3 階層構造の各階層を形成するビュー成分である (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を使う方法が既に確立[6]しており、コンピュータ可読の方式として有用である。

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, 行列成分として, 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場合を $[P]Q$ の式で, それ以外の場合を 0 で表す隣接行列を, 「パス歩行行列」と呼ぶ。

パス歩行行列の歩行ルールは以下のとおりである。ii) はノードの 2 度読みを排除する原則を内含する。

- 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こと。
- iv) 最後に 0 行列を得たら妥当な歩行である。

図-1 に対応するパス歩行行列 U [下記の先頭行列] への図-2 のパス歩行は以下のように展開する。

$$E \bullet U = E \bullet \begin{bmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 & [E]\bar{D} \\ 0 & 0 & 0 & [D]\bar{E} & 0 \end{bmatrix} \Rightarrow E\bar{D} \bullet \begin{bmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow E\bar{D}\bar{B} \bullet \begin{bmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & 0 & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow E\bar{D}\bar{B}(\bar{A}, \bar{C}) \bullet \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow$$

$$E\bar{D}\bar{B}(\bar{A}, \bar{C}) \bullet \begin{matrix} \text{View 1} \triangleright \triangleleft \\ \text{View 2} \\ \text{View 3} \end{matrix} = \begin{bmatrix} 0 & 0 & 0 \\ [View 1]^E \text{View 2} & 0 & 0 \\ 0 & [View 2]^B \text{View 3} & 0 \end{bmatrix} \text{ s.t. } \begin{cases} \text{View 1: } E \in \mathbf{X} \\ \text{View 2: } \bar{E}\bar{D}\bar{B} \in \mathbf{Y} \text{ where } \text{View 2} = D \bullet \\ \text{View 3: } \bar{B}\bar{C} \in \mathbf{Z}. \end{cases} \begin{bmatrix} 0 & [B]\bar{A} & 0 \\ 0 & 0 & [D]\bar{B} \\ 0 & 0 & 0 \end{bmatrix}$$

注) $\triangleright \triangleleft$ の結合記号の右はビュー構造行列[7]. なお View2 はパス足跡小行列[7]で構造表現されている.

2.3. パス足跡小行列からのプログラム生成

上のパス足跡小行列は, フォーム生成ビューを構成するノードを結合してビュー生成する情報として用いられる. ビュー構造行列はフォーム生成ビューを結合してフォーム生成する情報として用いられる. 標的DBがRDBの場合, ノードはテーブルとなる. ビュー構成テーブル群を T_1, T_2, \dots, T_n とすれば, これらはそれぞれの主キーと外部キーで結合されるので, SQL 文は以下ようになる. where 以下の結合条件については, すべての結合関係が, $\triangleright \triangleleft$ の前件項とパス足跡小行列内の成分によって定義されているので, これを用いて網羅的に SQL 文を生成し, 実行することができる.

Select * from T_1, T_2, \dots, T_n where 結合条件

このような方法は他のタイプの実装DBにも適用可能であり, ノードと結合要件さえ分かれば, 生成されたパス歩行行列を基底とする全ての帳票・フォームのパターン群が網羅できる.

3. 仮想定ノードとパスの扱い

3.1. Real/Virtual スキン識別とサブビューの導入

仮想定ノードとパスを扱おうとすれば新たな議論となる. フォーム生成機構の中核にあるパス歩行行列は, 特定の実装DBを想定せず, 単にノード同士の結合関係をグラフ理論の隣接行列で表現したものに過ぎない. 従って結合可能な個物として特定されるものならば何でも良い. ということは, 結合対象となるノード群は, ヘテロなDB環境のもの同士でも構わないし, 実装DBそのものが存在せず「そのようなデータ実体として仮想定されたノード」でも構わないことになる. また, ノードは, 結合可能であれば, どのようなメディア形式であっても構わないし, 正規形であるかどうかは問わない.

このようなフォーム構造の解析は, それぞれのノードやパスにスキーマ定義が与えられなくとも行える. ただしパス歩行の結果を, 実際のフォームイメージに展開するときには注意が必要である. 対象ノードにインスタンス登録のためのスキーマ定義があれば, ビュー構造行列を使ってスキーマ操作言語を生成することにより, 格納されたインスタンス内容をマップしたフォームが生成可能だが, そうでない場合は, 別の方法によってフォーム生成イメージを作り上げる必要がある.

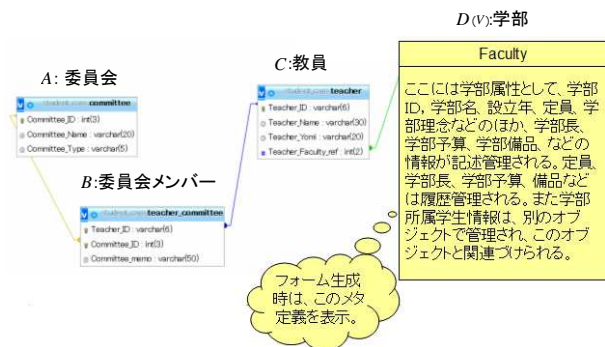


図-4 実データノードと仮想定ノードの組合せ

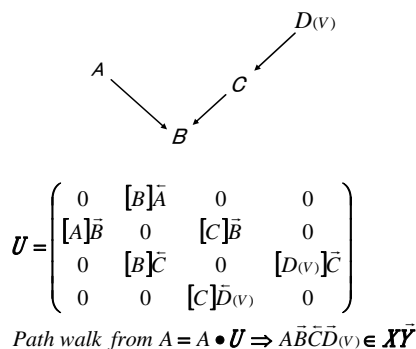


図-5 図-4のデータソース構造とパス歩行行列

例示しよう. 図-4は4つのノード (A: 委員会, B: 委員会メンバー, C: 教員, D: 学部) からな

るデータソースである。ただし D は仮想定 [ノード名と若干のノード説明しか決まっておらずスキーマ未定義: $D(w)$ と表記] のまま、スキーマが定義済みの C とパス結合している。両者のパス認識は漠然としており結合仕様は未定義である。この状態での A をエントリとするパス歩行行列 U に対するパス歩行結果は $A \bullet U$ の演算で得る概念フォーム式 (図-5) から直ちに 2 階層構造と判明するが、このフォームを実際に生成しようとするとき、通常スキーマ操作言語の処理だけでは C から $D(w)$ への結合の実現方法で躓くことになる。 C と $D(w)$ を媒介するスキーマ定義が存在しないからである。

概念フォームといえども実データを取り扱う箇所は実装制約を受ける。この課題を克服するため本研究では「Real/Virtual スキン識別」概念と、これに基づく「サブビュー」概念の新たな導入を提案する。インスタンス登録のための属性が定義された場合のノードのスキンを Real, そうでない場合を Virtual と定義する。Real/Virtual は便宜的な呼称である。またノード間のインスタンスを連結するためのパス属性^脚が定義された場合のパスのスキンを Real, そうでない場合を Virtual と定義する。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間であってもインスタンス連結を定義する属性がないパスのスキンは Virtual である。上の例ではノード A, B, C のスキンは Real, $D(w)$ のスキンは Virtual, パス($C, D(w)$)のスキンは Virtual である。

ノードの属性に、以下の 3 つの属性を与える。

- (i) 「ノード・スキン識別」属性: スキーマ有無によって Real/Virtual のスキン識別と定義する。
- (ii) 「ノード説明」属性: ノードの特徴および運営にまつわる諸事項を記載する。記載の表現とメディアは問わないので、文言、ピクチャ、音声、動作イメージその他何でも登録して構わない。
- (iii) 「スキーマ操作言語」属性: スキーマ操作言語 (SQL, Xquery など) を記載する。独自の操作言語を作る場合は、その言語の呼び出し名を記載する。

またパスの属性についても、同様に、「パス・スキン識別」、「パス説明」の属性を与える。

上のスキン識別に基づく「サブビュー」概念は以下のものである。

フォーム生成ビューにあつて、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」とする。また、Virtual スキンの単体ノードを以て「Virtual サブビュー」とする。サブビュー同士は共通ノードを持たないため、サブビュー群とフォーム生成ビューは直和分解の関係にある。

上の基準によりサブビューは自動認識される。Real ノード同士であっても Virtual パスで結合する場合は、上の定義の帰結として、単体の Real ノードがそのまま Real サブビューとなることもある。また互いにヘテロなスキーマ定義のノード同士ということが障ってノード結合に必要な操作命令の生成が困難な場合も— 当座の措置として— 単体の Real ノードをもって Real サブビューとすることを認める。Real サブビューのインスタンス内容は、ノードごとに定義されたスキーマ操作言語で取り込むことができる。

以上を準備として、Real/Virtual スキンを含む場合のフォーム生成方法を整理すると、表-1 のスキン組合せへの対応が必要となる。ただし後件ノードの表示方法に限れば、後件ノードが Real スキンのときに後件側の結合インスタンス集合が ①特定できる場合と ②特定できない場合、および後件ノードが Virtual スキンのときに後件側の ③結合インスタンス自体が存在しない場合の 3 つに類別される。

①の場合はスキーマ操作でインスタンス表示する。②の場合の原因事態は、 α, β, γ の 3 つに分かれる。

α : 前件ノードの確定インスタンスと結合するパスが未定義で後件ノード・インスタンスが決まらない、

β : 前々件ノードからのパスや前々件ノードのスキンが Virtual のため前件ノードのインスタンスを確定できず、その結果、後件ノード・インスタンスが特定できない、

γ : 前件ノードが Virtual スキンのため後件ノード・インスタンスが特定できない。

これらに対し本稿では結合候補の後件ノード・インスタンス集合をユーザに提示し状況説明を添え書きする方法を薦める。③の場合は後件ノードが Virtual スキンであるためノード・インスタンス自体が存在しない事態だが、これについてもインスタンス不在の状況説明を添え書きする方法が適切であろう。

^脚 RDB の外部キー属性や XML の子要素定義など、ノード群をパス連結する属性に使える物理仕様が存在すること。

表-1 Real/Virtualスキンの組合せとフォーム表示の対応

前件ノードのスキン	前件ノードの状態	パス結合のスキーマ定義有無	後件ノードのスキン	
			Realノード	Virtualノード
Realノード	インスタンスが存在 インスタンスの確定有無⇒	○	① 結合インスタンス集合が特定できる	③ 結合インスタンス自体が存在しない
		×	② 結合インスタンス集合が特定できない	
Virtualノード	インスタンス不在	γ: ×		

前々件ノードからのパスや前々件ノードのスキンがVirtual

結合候補となる後件ノード・インスタンス集合をユーザの目に晒した上で、状況説明を添え書きする。

スキーマ操作でインスタンス表示する。

ノード説明等に、状況説明を添え書きする。

上のA: 委員会をエントリとするパス歩行 (図-6) の場合, CからDへのパスはaかつ③のケースに対応するため, この箇所はノード名やノード説明にインスタンス状況を添えたフォームイメージとなる (図-7)。

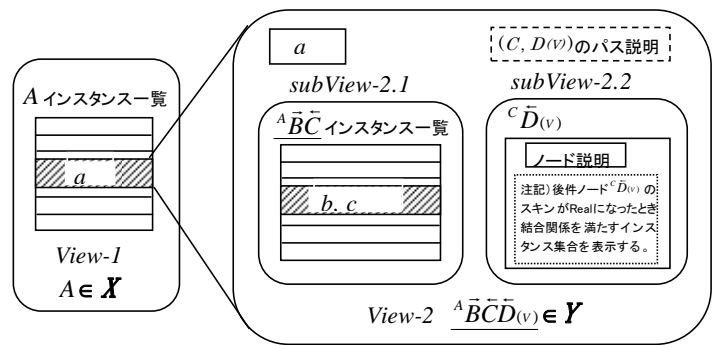
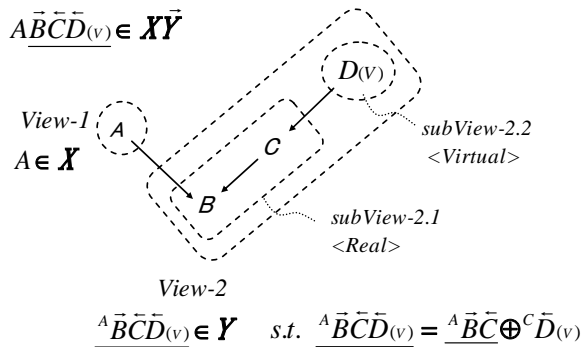


図-6 図-5のデータソースへのパス歩行

図-7 パス歩行の結果, 得られた $\overline{A\overline{BCD}}_{(v)} \in \overline{XY}$ のフォーム展開

3.2. 仮想定パス歩行行列への拡張と応用

このようなスキン組み合わせ対応のフォーム生成原理は, 仮想定ノードやパスが自由かつ多様に議論されるべき要求分析プロセスで使われる。仮想定ノードやパスがスキーマ定義により Real スキンに変わるとサブビュー構成が自動的に変わりノード・インスタンスがユーザの目の前に現れるようになり実践レベルでの要件検討が促進される。Real ノードの内容に不満があれば Virtual スキンのノードやパスに差し替えて議論しなおせばよい。この活動の繰り返しはユーザ自身による要求分析の促進に繋がる。

こうした活動を実現するため, 以下のように「仮想定パス歩行行列」への拡張を行う。

所与のパス歩行行列 U に対し, 仮想定ノードに対応する行列成分の付加や, 特定ノードに対応する行列成分の隠蔽^{注1}を施した行列を N_{θ} とし「ノード投射行列」と呼ぶ。 N_{θ} に対し仮想定パスの付加や特定パスの隠蔽を施すための上書き行列^{注2}を $L_{i,j}$ とし「パス上書き行列」と呼ぶ。 i, j は版識別の添え字である。 L による N の上書きを $overwrite(N, L)$ で表す。 $M_{i,j} = overwrite(N_{\theta}, L_{i,j})$ となる $M_{i,j}$ を「仮想定パス歩行行列」と呼ぶ。 仮想定ノードとパスを前提とするパス歩行によるフォーム生成は $M_{i,j}$ 上で行う。

注1) パス歩行対象外の成分を $\overline{h} : \text{hbar}$ で表す。 特定ノードの隠蔽は当該ノードに対応する行及び列の全成分の \overline{h} による上書きで, また特定パスの隠蔽は当該パスに対応する行列成分及び対称成分の \overline{h} による上書きでそれぞれ表す。 隠蔽ノードと結ばれる全てのパスは無効となるため上の処理において自動的に隠蔽される。 行列を構成する成分が0または \overline{h} である行列を H 行列と呼ぶことにする。 仮想定パス歩行行列におけるパス歩行の完了判定基準は O 行列もしくは H 行列への到達となる。

注2) L の非上書き成分はヌル値だが, これを $-$ で表す。

上の例で示そう。 ノードA:委員会を隠蔽する代わりに仮想定ノード $\overline{E}v$: チームを加え, B:委員会メンバーをチームメンバーとみなすノード投射行列 N を作る。 付加成分は [まだどのノードともパス結合が定義されていないため] O で, 隠蔽成分は \overline{h} で表す。 この N に対し図-7 下段右のように新ノード $\overline{E}v$ から B への

1 対多パスを付加するパス上書き行列 L を与える (図-6) と, 仮想定パス步行行列 $M = \text{overwrite}(N, L)$ が求まる. こうした繰り返しによってユーザは新たなフォーム視点からの要求分析を進めることができる.

Given $[A, B, C, D(v)]$,

$$U = \begin{pmatrix} 0 & [B]A & 0 & 0 \\ [A]B & 0 & [C]B & 0 \\ 0 & [B]C & 0 & [D(v)]C \\ 0 & 0 & [C]D(v) & 0 \end{pmatrix} \Rightarrow N = \begin{pmatrix} h & h & h & h & h \\ h & 0 & [C]B & 0 & 0 \\ h & 0 & [C]D(v) & 0 & 0 \\ h & 0 & 0 & 0 & 0 \end{pmatrix}$$
 Given $[A, B, C, D(v), E(v)]$ s.t. A is hidden,

$$L = \begin{pmatrix} - & - & - & - \\ - & - & - & [E(v)]B \\ - & - & - & - \\ - & - & - & - \\ - & [B]E(v) & - & - \end{pmatrix} \therefore M = \text{overwrite}(N, L) = \begin{pmatrix} h & h & h & h & h \\ h & 0 & [C]B & 0 & [E(v)]B \\ h & 0 & [C]D(v) & 0 & 0 \\ h & 0 & 0 & 0 & 0 \\ h & [B]E(v) & 0 & 0 & 0 \end{pmatrix}$$

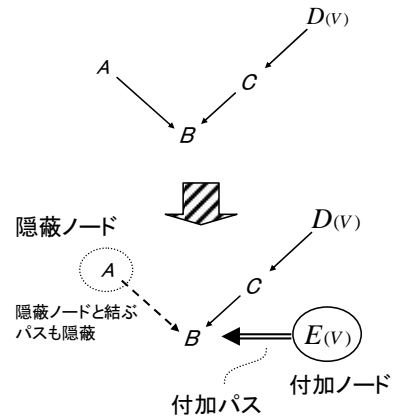


図-6 所与のパス步行行列を上書きして仮想定パス步行行列を作る

図-7 データソースの上書き

ちなみに〔実務有効性の議論と別に〕最も粗い要求レベルとして, パス步行行列を構成するノードの洗い出しも未着手のときは所与 U を O から始めて構わない. このときは仮想定ノードのみから構成された N_{θ} に対し仮定定のパスを与えた $L_{i,j}$ から得た $M_{i,j}$ を用いてパス步行検討することが可能である.

4. 議論と展望

本稿では実装と仮想定を問わずノードとパスをパス步行行列で扱う方法として, 新たな2概念〔Real/Virtual スキン識別とサブビュー〕を導入した仮想定パス步行行列への拡張と応用を提案した.

仮想定パス步行行列を実装したツールに関する実験結果と動作イメージ (図-8) は, 本学会の「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) 例会で報告したが, 要求分析に勤しむユーザの探究活動を促すためのユーザインタフェースの検討は今後の課題である.

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い. Virtual スキンのノードやパスの追加変更や隠蔽は実装の制約を受けず容易に行えるので, このモードでのデータソースの形態変化やパス步行はユーザの自由な発想を促す可能性を秘めている. この論点は本研究の射程にあるが, それを生かした要求分析への活用については稿を改めて議論したい.

選択された実体ノード一覧 selected list of real/virtual node(s)				
ID	NAME	NARRATIVE	SKIN	DESCRIPTION
11527	Form_sales_registration	販売登録	REAL	販売案件を個別に識別管理する。販売を識別する番号を発行する。
11437	Form_customer_registration	顧客マスター登録	REAL	種々の取引相手である行を別区分し、取引の
11301	Form_organization_registration	部署マスター登録	REAL	部署をマスター管理する
NCT-1	JouTokui	版ノード1: 上得意別帳	VIRTUAL	上得意の管理情報は通業務管理が可能になる
NCT-3	Manuhi	版ノード4: 営業マール秘モ	VIRTUAL	営業担当の手元で理蔵源にあるので共有化の

パス步行の対象となる妥当なリレーションシップ Relationships valid for path walking matrix									
■ 選択された妥当な実リレーションシップ ■ 選択された妥当な仮想リレーションシップ ■ 非妥当もしくは非選択により無効となったリレーションシップ									
Fixed	Select all rels	FORM_OID	FIELD_OID	REFERENCED_FORM_OID (Many-side)	REFERENCED_FIELD_OID (One-side)	SKIN	link_type	child_name	child_narrative
1	1	11527	11541	11437	11446	REAL	Ref_link	Form_sales_registration	
1	1	11527	11570	11353	11358	REAL	Ref_link	Form_sales_detail_BAG	
1	1	11527		NCT-1		VIRTUAL	NCT	Form_sales_registration	販売登録

図-8 実装実験における仮想ノードとパスの追加変更や隠蔽を行う画面の例

参考文献

- [1] Wieringa, R. J., *Requirements Engineering*, John Wiley & Sons, 1996.
- [2] Lauesen, S., *User Interface Design*, Peason Addison Wesley, 2005.
- [3] Halpin, T., *Information Modeling and Relational Databases*, Morgan Kaufmann Publishers, 2001.
- [4] 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, Vol.6, No.4., 1998.5.
- [5] 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, Vol.11, No.1, 2002.6.
- [6] Nakanishi, M. “Office Form Generator Supported with Path Reading Matrix”, *Economic and Business Review*, Vol.15, No.1, Nagoya Keizai University, 2007.9.
- [7] 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.
- [8] 中西昌武, “パス步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.