

パス歩行行列を媒介した帳票検討プロセス

Form study process through path walking matrix

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

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。本研究では帳票・フォームの論理構造を、DBアクセスの歩行経路構造とみなし、パス歩行行列を使って表現する。パス歩行行列は、データノード間の1: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・フォーム生成のための最も基本的で有用な情報を管理する。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、生成可能な帳票・フォームパターン群が洗い出せ、種々の帳票・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。このアプローチは、DBの正規化が不十分の場合も、実装DBが存在しない場合も適用可能であるので、最上流の曖昧状況での活用方法についても応用が期待できる。

1. はじめに

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。これらを要求仕様3者と仮称することにする。エンドユーザにしるシステム設計者にしる、要求分析するに当たっては、実際に触れるであろう帳票・フォームの使用局面を思い描き、ときに感触を味わい、ときに思いもよらない使用方法を試して、確かな手ごたえが得られる要求仕様3者の解明へと向かう。

この刹那の感覚について、筆者は、認知科学の泰斗である佐伯胖[1]の「視点の活動論」に引き寄せられる。彼は「重要なのは範例ではなく、範例化し、それらをつなぐ合成や変形の操作を、自分の目で、自分の体感で実感できるものの中から発見することである。それができたとき、わたしたちは真底から、ナルホド、ナルホド、ナルホド、と実感する」と述べるのだが、実はそのようなとき、実感する当人は思考作業を支える基底ベクトルのようなものを手にしている、というのが佐伯の主張である。

話は戻って、では要求分析プロセスにおいて、帳票・フォームを眺め回すときに、分析担当者の中に何か基底となるものはあるのだろうか。筆者は、要求分析には、基底となるイメージを媒介とする認識過程が存在すると見ている。すなわち要求分析は、基底イメージの持つ仕様に対し、あるときは整合性の確保を、別のあるときは利害対立を超越する新仕様の確保を目指しつつ、あれこれ点検する過程であると見ている。このとき帳票・フォーム構造とDB構造と業務構造の3者は基底イメージを共有しつつ互いに牽制しあっている。実際に触れるであろう帳票・フォームを眺めるときは、業務仕様との違背は許されなく、たとえ姿は見えなくとも標的DBの論理構造の影を眺めているはずである。

見方を変えてみよう。標的DBから導出可能な帳票・フォームの論理構造が、理論的に網羅可能であれば、それは要求分析に寄与する基底イメージのひとつとなりうる。それを可能とするのが本稿で説明する概念帳票モデルである。モデル記述に隣接行列（「パス歩行行列」と呼ぶ）を用いることで数理科学的に扱えるようにし、それと同時に、コンピュータ可読性を確保することで自動生成のプログラミングに道を開く。本稿は、概念帳票モデルの活用による帳票・フォーム検討の小さな提案である。

2. 概念帳票とパス歩行行列

2.1. 概念帳票モデル

著者が提唱する「概念帳票」は、帳票・フォームから物理的成分を除去して得た帳票の概念的な構造であり、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として理解することもできる。また「概念帳票式」は、概念帳票の構造を数式で表現したものである。

標的とする概念データモデルの構造が特定できれば、それにアクセスする概念帳票の構造は有限個の

パターン群に特定でき、しかもその構造は概念データモデルの辿り方によって〔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、というのがこれまで判明した知見[2]である。

例で示そう。図-1は、5個のデータノードから構成されるデータソースの簡易的表現である。各ノードのシンボルはデータノードを、矢線はデータノード間の1:多の結合を表す。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。

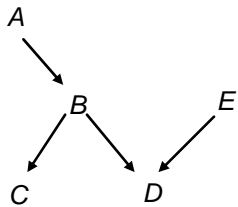


図-1 サンプル・データソース

★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: 商品, B: 商品現品, C: 状態履歴, D: 販売, E: 顧客

図-1のデータノード E をエントリノードとし、指定された全てのデータノードを渡り歩くDBアクセスはどのような帳票を作るだろうか？ 歩行対象が木構造であるとき、特定ノードを出発点とするパス歩行経路パターンが常に1個だけ得られることは、グラフ理論を待つまでもなく自明だが、その点に留意しつつ、帳票構造を表す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。

概念帳票式では、以下のような記号を使用する。

アルファベットで表記されたデータノードの上の矢線記号

- 1:多の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:多の結合を1の側に向けてパス歩行するアクセス〔下の例示みよ〕

\overline{DB} (下線記号) データノード D に対し B は非正規化により 1:1 の行結合を生む。

${}^B\overline{C}$ (左肩記号) データノード C は、B から 1:多の結合のアクセスで得られる。

この帳票の概念的な展開イメージは、次のとおりである。E から D へのアクセスは 1→多のパス歩行であるから E をヘッダ、D をボディとする 2階層構造 $E\overline{D} \in XY$ をひとまず作る。D から B へのアクセスは、多→1のパス歩行であり、D とは 1:1 の行結合 \overline{DB} を作るから、帳票の 2階層構造は維持されて、 $E\overline{DB} \in XY$ となる。B から A へのアクセスも、多→1のパス歩行であるから同様 1:1 の行結合により \overline{DB} は \overline{DBA} へと成長する。帳票の 2階層構造も維持されて、 $E\overline{DBA} \in XY$ となる。残るデータノード C へは、B からアクセスし、それは 1→多のパス歩行であるから B の成分を内含する \overline{DBA} をヘッダ、C をボディとする 2階層構造 $\overline{DBA} \cdot {}^B\overline{C} \in YZ$ をひとまず作る。ところが \overline{DBA} は、ヘッダである E に対するボディであるから、帳票の全体構造は 3階層の $E\overline{DBA} \cdot {}^B\overline{C} \in XYZ$ を作る。

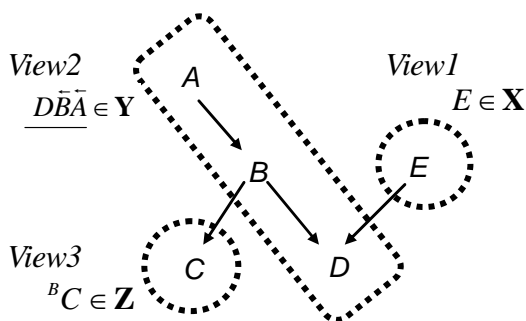


図-2 サンプル・データソースのパス歩行結果

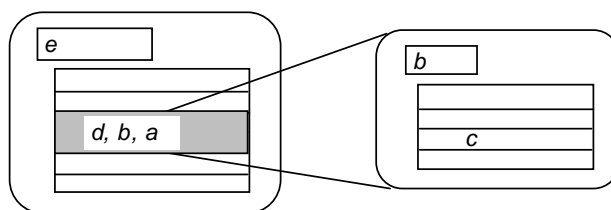


図-3 パス歩行の結果、得られた $E\overline{DBA} \cdot {}^B\overline{C} \in XYZ$ の帳票展開

ここで $E \in X$, $\overset{E}{D}\bar{B}\bar{A} \in Y$, $\overset{B}{C} \in Z$ は, それぞれ概念帳票に内在して 3 階層構造の各階層を形作る View 成分である (図-2).

なお, $\overset{B}{C}$ の B から C への展開部分は, 実際には, $\overset{E}{D}\bar{B}\bar{A}$ のひとつの行選択に対しポップアップに開く子画面などの形で実装されることになる (図-3). これは現在の実装技術と認知能力の制約によるものであって, 将来は抜本的な変革が起こるかもしれない. しかしデータの概念的な構造は変わらない.

詳細は文献[2]に譲るが, 図-1 のグラフ形状 [H_3 型] からは, 合計 10 種の帳票が導出できる.

2.2. パス歩行行列

概念帳票式の導出について筆者は, データ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を提案[3]しており, コンピュータ可読の方式として有用である.

パス歩行行列を以下のように定義する.

パス歩行の前件側データノード (P) を列, 後件側データノード (Q) を行とし, 行列成分として, データ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 O で表す隣接行列を, 「パス歩行行列」と呼ぶ.

図-1 のデータソースのパス歩行行列は図-4 のようになる.

アクセスされたデータノードはパス歩行行列から抜き出されて概念帳票式の成分に順次組み込まれてゆく. たとえばデータノード E をエントリとする 図-1 へのアクセスは, 図-5 のごとく展開して概念帳票式を成長させ, いっぽうパス歩行行列は O 行列に向かって縮退する. すなわち前件のデータノードに対応する列の中の非 O 成分を見つけたら, 概念帳票式に加えるとともに該当成分とその対称成分を O 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, それとともに概念帳票式

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

を得る. 図-5 の最終行にあるこの式は, まだデータノード間の多→1 関係がもたらす 1 : 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帳票式を得る.

$$E\bar{D}\bar{B}(\bar{A}, \bar{C}) \Rightarrow \overset{E}{D}\bar{B}\bar{A} \cdot \overset{B}{C} \in XY\bar{Z} \tag{2}$$

2.3. パス足跡小行列を活用した View 構造の導出

筆者は, 式(2)の $\overset{E}{D}\bar{B}\bar{A} \in Y$ のような View 成分へのコンピュータ可読の行列表現をすでに提案[4]している. 図-6 のように, 抽出された View を, それと等価のパス歩行部分行列 [式(4)] で表すと同時に, 生成された帳票の View 構造を表現する行列 [式(3)] も行列演算によって自動的に得ることができる. 前者を「パス足跡小行列」, 後者を「View 構造行列」と呼ぶ. これは以下のメリットをもたらす.

- ① 概念帳票式の字句解析が不要となる.
- ② SQL や Xquery は, パス足跡小行列から直接, 自動プログラミングできるようになる.
- ③ パス足跡小行列から等価の概念帳票式をいつでも生成することができる.

パス足跡小行列を生成するに当たり, 2つの規則が適用される.

- (1) 後件に対し 1 : 多のパス歩行をする場合は, 新たなパス足跡小行列を作り, 結合する.
- (2) それ以外のパス歩行をする場合は, 現在のパス足跡小行列の成分に, 新たなパス歩行によって得た成分を加筆してパス足跡小行列を成長させる.

パス足跡小行列同士の結合記号としては, DB 理論で使われている結合記号の $\triangleright\triangleleft$ を用い, 必要に応

じて結合記号の下に前件データノードのシンボルを記す. [自明の結合の場合は●もしくは無記号.]

View 同士の結合関係は, データノード同士の結合関係の要約概念であるから, 同じ理由によって, パス歩行列で表現できる. 逆に言えば, データノードが正規化されていなくとも, この処理は可能であるということだ. **View** 構造行列の各列がサブ帳票となり, 列成分の **View** が各サブ帳票の構成 **View** [子の列挙] となる. パス歩行で認識した **View** は直ちに **View** 構造行列に反映する. 図-2 [ないし式(2)] の **View** 構造は図-6 のように表し, **View** を結合して帳票を自動生成する情報として用いられる.

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

図-4 図-1 のデータソースのパス歩行列

$$View1 \triangleright \triangleleft_E \begin{bmatrix} 0 & 0 & 0 \\ [View1]^E View2 & 0 & 0 \\ 0 & [View2]^B View3 & 0 \end{bmatrix} \quad (3)$$

$$st. \begin{cases} View1: E \\ View2: \bar{E} \bar{D} \bar{B} \bar{A} \\ View3: \bar{B} \bar{C}. \end{cases}$$

$$View2 = D \bullet \begin{bmatrix} 0 & [B]\bar{A} & 0 \\ 0 & 0 & [D]\bar{B} \\ 0 & 0 & 0 \end{bmatrix} \quad (4)$$

図-6 生成された **View** 構造行列と, **View2** のパス足跡小行列

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

$$E\bar{D} \bullet \begin{bmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & [D]\bar{B} & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & [B]\bar{D} & 0 & 0 & \mathbf{0} \\ 0 & 0 & 0 & \mathbf{0} & 0 \end{bmatrix} \Rightarrow$$

$$E\bar{D}\bar{B} \bullet \begin{bmatrix} 0 & [B]\bar{A} & 0 & 0 & 0 \\ [A]\bar{B} & 0 & [C]\bar{B} & \mathbf{0} & 0 \\ 0 & [B]\bar{C} & 0 & 0 & 0 \\ 0 & \mathbf{0} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow$$

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

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

図-5 図-2 の例へのパス歩行列の適用

2.4. パス足跡小行列からのSQLプログラム生成

ここでは概念帳票が実装DBにアクセスする場合の帳票・フォーム生成の方法について述べる. 標的DBがRDBの場合, 各テーブルがデータノードとなる. **View** を構成するデータノード群に対応するテーブルを 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 の場合は以下のようなになる.

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

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を生成し, 実行することができる.

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, データノードと結合要件さえ分かれば, 生成されたパス歩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.

2.5. 概念帳票エンジンの機構

本稿で提案したパス歩行行列の利用による帳票・フォームの自動生成エンジンを「概念帳票エンジン」と呼ぶ。筆者はすでに、以下の実験環境で実装技術を検証済みである[5] [図-7のサービスを想定]。

環境: Windows Server + XAMPP
 DB: MySQL
 概念帳票エンジン実装言語: PHP

このエンジンは以下の手順でデータソースの解析から帳票・フォームの生成までを行う。必要箇所ではエンジンはユーザに対し選択対象を列挙表示し、ユーザ選択を受けて、エンジンは次の工程に進む。

- ① DBから解析対象スキーマを列挙表示する。→ ユーザは任意選択。
 - ② スキーマの実装情報を解析して、パス歩行行列を PHP 配列として生成する。
 - ③ アクセス対象テーブル〔データノードに該当〕を列挙表示する。→ ユーザは任意選択。
 - ④ アクセス・エントリ〔となるデータノード〕を列挙表示する。→ ユーザは任意選択。
 - ⑤ パス歩行行列を解析して生成対象となる View 構造を解明し、各 View 成分のパス足跡小行列と View 構造行列を生成する。
 - ⑥ パス足跡小行列に従って SQL を生成してDBを呼び出し、実 View を生成する。
 - ⑦ View 構造行列に従って、View を組み合わせてプロトタイプ帳票・フォームを生成する[†]。
- [†] Web フォーム (HTML + JavaScript + CSS) およびそのデータマート (.csv または .js ファイル) の生成。ただし当実験では標準フォームを設定しての生成 [カスタマイズは別途実験中]。
- ⑧ さらに、商用帳票パッケージ(FormEditor&DocCreator by HOS 社)の帳票パラメータの生成^{††}。
- ^{††} 帳票の物理仕様を実装設計するための雛形パラメータが生成・提供される。
- ★ ③~⑦の任意の時点で、③もしくは④に戻り、仕様変更して再試行することができる。
 - ★ 上記工程で、ユーザもしくはデザイナーがDB構造を変更した場合は、①に戻って再試行する。
- 実験の結果、概念帳票式の構造の帳票を行列演算により自動生成することができた (図-8)。

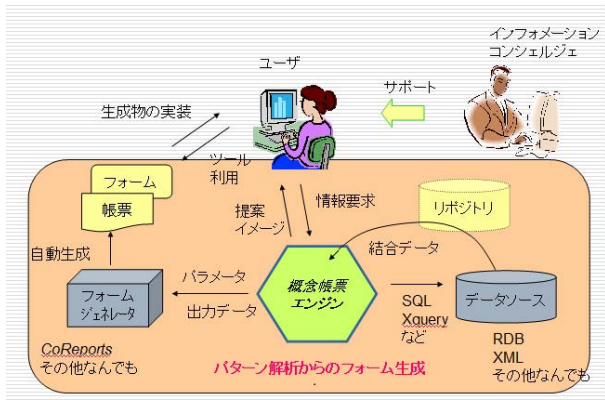


図-7 概念帳票エンジンの機構イメージ

注)

ユーザが、概念帳票エンジンを媒介して、自身の業務関心である標的DBのデータノード群を選択すると、エンジンは当該データノード群が基底となって生成可能な帳票・フォームのパターン群をユーザに提示する。ユーザは、これを見て、自身の要求仕様を点検することが出来る。ユーザが不慣れで作業に不安がある場合は、インフォメーション・コンシェルジェがサポートすればよい。

図-8 生成された Web 画面の展開例

3. パス歩行行列を媒介した帳票・フォーム検討プロセス

以上の技術整備を前提として、本稿の主題であるパス歩行行列を媒介した帳票・フォーム検討プロセスの理論的射程について述べる。

概念帳票エンジンの中核にあるパス歩行行列は、特定の実装DBを想定せず、単にデータ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環境のもの同士であっても構わないし、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データノード」であっても構わないことになる。また、データノードは、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。

「仮想定データノード」を、結合対象に含める表現方法としては、当該データノードのメタデータをスキーマとして「ワーク支援用リポジトリ」に登録して呼び出せばよい。このリポジトリは情報資源管理用のセンター・リポジトリとは別物である。

図-9は、実装されたRDBのレガシー・テーブルに、新たな仮想定データノードを加えた組合せのイメージである。図-10は、新たな要求仕様を満たすべき実装データノードが全く存在しない場合に生じるオール仮想定データノードの組合せイメージである。いずれの場合も、結合対象となるデータノードは、実装と仮想定を問わずパス歩行行列で表現可能である。帳票・フォームのプロトタイプ生成において、実データが存在する場合は実データを表示〔図-8参照〕すればよいし、実データが無いデータノードについてはメタデータ（文言、ピクチャ、音声なんでも可）を帳票・フォームイメージ内に埋め込み表示すればよい。概念帳票エンジンは、こうした曖昧な仕様のデータ構造が与えられた場合でも、そのデータ構造から導出されるべき帳票・フォームのパターン群を網羅的にユーザに提示する。ユーザは、このような基底イメージを元にしつつ、次第に自らの要求仕様イメージの点検を深めることが出来る。最上流の曖昧状況での活用方法についても応用が期待できる方法である。

上で述べたことは、まだ構想段階であり、実証実験はこれからの予定のため、本稿では、この目論見が理論的に可能であると主張するにとどめる。

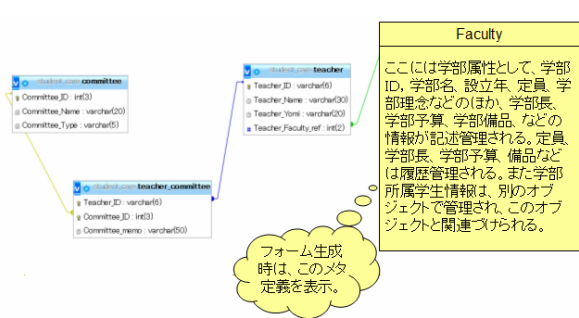


図-9 実RDBと仮想定データノードの組合せ

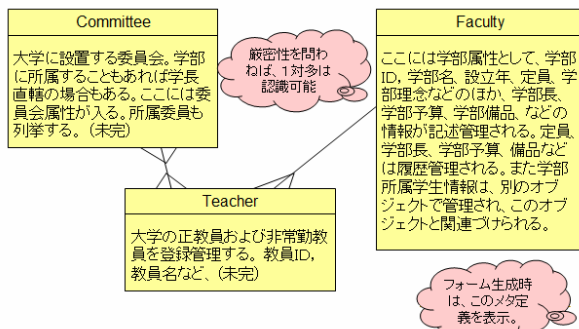


図-10 全ての標的が仮想定データノードの場合

参考文献

- [1] 佐伯胖, イメージ化による知識と学習, 東洋館出版社, 1978, pp.264-275.
- [2] 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.
- [3] Nakanishi, M. “Office Form Generator Supported with Path Reading Matrix”, *Economic and Business Review*, Vol.15, No.1, Nagoya Keizai University, 2007.9.
- [4] 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.
- [5] 中西昌武, “概念帳票エンジンのプロトタイプ”, 経営情報学会 2010 年秋季全国研究発表大会予稿集, 2010.11.