Vol.2019-IS-150 No.13
2019/11/24

ハッセ図構造を持つ実体関連モデルと第3正規形データベース

金田 重郎[1,a)]　井田 明男[1]

概要：UML クラス図・ER 図を用いて，実体関連モデル（以下，ERM）を作成する概念データモデリングの重要性については論を待たない．著者らは，ERM 作成のガイドラインとして，「1 対多関連」のみを利用して，1 対多関連を時間的前後関係 (半順序) と見なした場合に，ハッセ図構造で ERM を描くことを既に提案している．これに対して，本稿では，同様に 1 対多関連を時間的前後関係と見なしてハッセ図構造を持つ ERM を出発点として，これを自明な方法で関係モデルのテーブル群に変換する．この際，ERM のエンティティの属性部分が第 3 正規形に正規化され，しかも，ハッセ図構造を持つ ERM でモデルを表現できれば，そこから生成される関係モデルのテーブルは第 3 正規形を満たす．この条件は逆も成り立つ．即ち，第 3 正規形の関係モデルデータベースが与えられた場合，それに相当する ERM は必ずハッセ図構造を持つ．本稿の内容は，(1) ERM によるモデリングは，対象ドメインのデータ構造を写し取るだけではなく，正規化を伴う作業であること，(2) ERM を作成する際には，対象ドメインの処理ステップを直接的な因果関係を示すプリミティブなプロセス群に分解する既約化が重要であること，を示唆する．

1. はじめに

概念データモデリングの重要性については論を待たない．しかし，我が国では，実体関連図（クラス図・ER 図）による概念データモデリングは広く実施されているとは言えない．その原因は種々あろうが，その一つは，実体関連図 (Entity-Relationship Model，以下 **ERM**) をどう描いたら良いかについて，多くの方に合意の取れた方法論が無い点にあると考える．つまり，現状では，各々のモデラーが，自分の経験・スキルに基づいた方法論で ERM を描いている側面がある．

著者らは，この問題を回避する一つのアプローチとして，容易に描くことができて，誰が書いても同じ様な実体関連図が得られる，モデリング方法論が望まれると考えている．

上記目的のため，著者らは，関連として 1 対多関連のみを利用し，ハッセ図構造（既約な有向非巡回グラフ構造）を持った ERM を描くことを提案している [1][2][3]．ただし，ここでは，ハッセ図構造は，実体関連図を描くためのガイドラインに過ぎなかった．

本稿では，従来研究のハッセ図構造について，従来研究とは異なる視点から分析する．即ち，提案手法では，(1) 関連の少なくとも片方の多重度が「1..1」（以下，「1」で表現）であり，エンティティの持つ属性は少なくとも第 3 正規化されているハッセ図構造の ERM を描き，次に，(2) 自明な方法で ERM を関係モデルのテーブル群に変換することを考える．本論文では，得られた関係モデルのテーブルが第 3 正規形を満たすことを示す．この条件は逆も成り立つ．即ち，生成された関係モデルが第 3 正規形（以下，**3NF**）を構成しているためには，元の ERM は必ずハッセ図構造を持たねばならない．

以下，第 2 章では，分析の前提を確認する．第 3 章は，提案手法のアウトラインを示す．第 4 章では，ハッセ図構造について説明する．第 5 章では，ハッセ図から出発して，関係モデルに変換するプロセスを定義し，理論的に分析する．第 6 章は，逆に，関係モデルから，元となる ERM が持つべき構造を分析する．第 7 章では，若干の議論を行う．第 8 章はまとめである．

2. 議論の前提

2.1 モデリング対象

最初に，本稿が対象とするエンティティ（クラス）について確認する．ロバストネス分析では，オブジェクト群（エンティティ群）を，Boundary Layer，Control Layer，Entity Layer の 3 レイヤーに区分される．特に，Entity Layer は，アプリケーションプログラム（以下，**AP** と呼ぶ）におい

[1] 同志社大学大学院理工学研究科情報工学専攻
Doshisha University, 1–3, Miyakodani, Tatara, Kyotanabe-city, Kyoto-pref., 610–0321,Japan
[a)] shigeokaneda@gmail.com

表 1　ERM の関連モデルへの変換方法

ERM の要素	関係モデルへの変換方法
エンティティ	多対多関連は無いので，エンティティがテーブルに 1 対 1 に対応する．エンティティ名はそのままテーブル名へ．エンティティが持つ属性も，そのままテーブルの属性とする．
関連	多重度が「1」ではない側のエンティティに該当するテーブル中のタプルに関連を表現する属性を作り，多重度「1」で指定された側のエンティティに相当するテーブル中のタプルの PK を，そこの属性値（FK）として格納する．

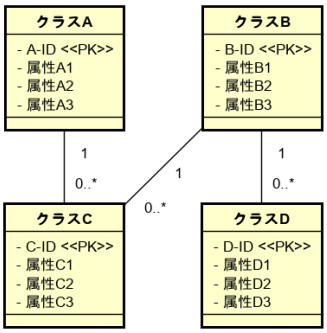

図 1　クラス図の例

テーブルA

A-ID <<PK>>	属性A1	属性A2	属性A3
A-1	(値)	(値)	(値)
A-2	(値)	(値)	(値)
A-3	(値)	(値)	(値)

テーブルB

B-ID <<PK>>	属性B1	属性B2	属性B3
B-1	(値)	(値)	(値)
B-2	(値)	(値)	(値)

テーブルC

C-ID <<PK>>	属性C1	属性C2	属性C3	関連1(FK)	関連2(FK)
C-1	(値)	(値)	(値)	A-1	B-1
C-2	(値)	(値)	(値)	A-1	B-2
C-3	(値)	(値)	(値)	A-1	B-2
C-4	(値)	(値)	(値)	A-2	B-1
C-5	(値)	(値)	(値)	A-3	B-2

テーブルD

D-ID <<PK>>	属性D1	属性D2	属性D3	関連1(FK)
D-1	(値)	(値)	(値)	B-1
D-2	(値)	(値)	(値)	B-2

図 2　変換後のテーブル群

て，データベース化されるべきエンティティ群である．本稿では，概念データモデリングに注目しているので，Entity Layer のエンティティ群がモデリング対象となる．

また，ERM には，多対多関連は利用しない．多対多関連が存在すると，関係モデルのテーブルに，エンティティを 1 対 1 変換できないからである．多対多関連は，関連を表現するエンティティを追加すれば，2 つの 1 対多関連に容易に変換できる．多対多関連を用いないからと言っても，モデルの表現能力が低下することはない．

ERM の表記方法としては，クラス図と ER 図のどちらでも良い．ただし，本稿における図示では，エンティティに便宜的にクラス図を用いる．また，本論文の議論では，エンティティのプライマリーキーは 1 属性（オブジェクト ID[4]）に限定する．以後，PK，FK をそれぞれ，プライマリーキー (Primary Key)，外部キー (Foreign Key) を示すものとする．

2.2　ERM から関係モデルへ

次に，多対多関連を持たない ERM をどのように関係モデルに変換するのか，自明の方法であるが，確認しておく．表 1 に示すように，エンティティはそのままテーブルに相当させる．エンティティが持つ各属性も，そのままテーブルの属性として引き継ぐ．

本稿の議論では，関連の持つ 2 つの多重度の内，少なくとも一方は「1」である．そこで，多重度が「1」ではない側のエンティティを変換して得られるテーブル中のタプル（レコード）に，リンクを表現する属性を設けることになる．そして，多重度が「1」のエンティティに相当するテーブル中のリンク先となるタプルの PK を，そのリンクを表現する属性に FK (NOT NULL) として格納する．図 1，図 2 は変換前と変換後のサンプルイメージを示す．また，本稿の議論では，ひとつのクラスから他クラスに延びる関連の数は最大 2 としている．

但し，本来の ERM では，多重度が双方ともに「1」ではないケースが想定される．例えば，「0..1」と「0..*」を多重度として持つ関連のケースである．この場合には，「1」を含まないので，この関連をクラス化して，2 つの 1 対 ANY の関連で表現する．この場合，クラス数増加が想定されるが，そもそも，このような（双方が「1」ではない）関連は少ないと思われるので大きな問題ではないと考える．また，双方が「1」の関連もあり得るが，これは，後述のハッセ図構造（既約な有向非巡回グラフ）の半順序で「等値（差なし）」と見なすことができる．

2.3　3NF と AP 処理の独立性（モジュラリティ）

前節の様にして，ERM は関係モデルのデータベースに変換される．AP の要請から，データベースは 3NF 以上でなければならない．データベースの教科書では，正規化はデータベースの更新異常を避けるためと説明されることが多い．無論，これは間違いではない．しかし，本稿では，自明なもう一つの解釈に注目する．正規化は，「テーブル毎に独立して処理を実行できること」である．正規化を，アプリケーションプログラム（以下，AP）のモジュール化と理解する．AP の中で，クラス毎に閉じて処理が終わる

からである．

そもそも，ERM においては，既存の ERM に，後から，別のエンティティとの間に，関連を自由に追加できなくてはならない．例えば，図 1 に例示したクラス図において，「新たなクラス（クラス F)」及び「クラス F から既存クラス C へ至る関連」をワンセットで追加することを考える．この時，クラス C から他クラス（クラス A,B）に至る関連がクラス A の追加の可否に影響を与える様では困る．クラス図では，文脈自由文法の様に，各部品（本稿では，クラスとそれから延びた関連）において処理が完結する．ERM 自体が，エンティティ毎の正規化の要請のもとに設計されたツールであるとも言える．

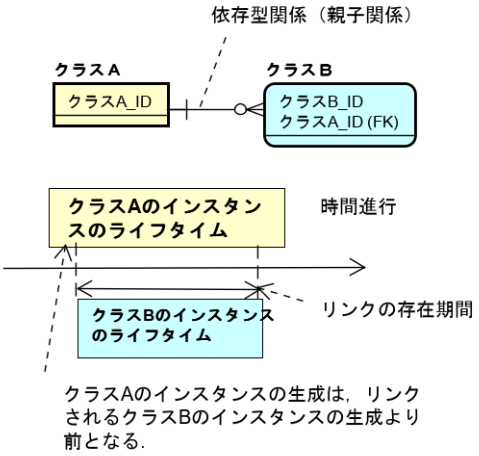

図 3　1 対多関連（依存型）

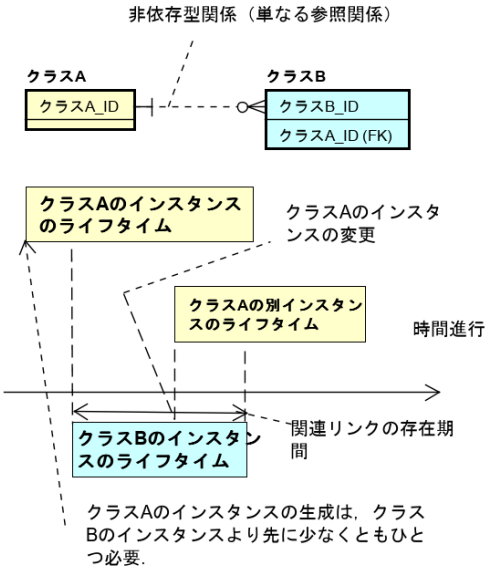

図 4　1 対多関連（非依存型）

3.　提案手法の概要

3.1　1 対多関連と時間的前後関係

本稿では，関連としては 1 対多（正確には一方の多重度が「1」であり，他方の多重度は何でも良い）の関連を用いる．最初に，1 対多関連は，時間的前後関係を含意することを示す．ただし，1 対多関連には，依存型と非依存型がある．依存型では，多重度が「1」ではない側のインスタンスが一旦作成されると，そのインスタンスが削除されるまで，多重度「1」側のインスタンスの取り換えができない．一方，非依存型では，多重度「1」ではない側のインスタンスが一旦作成されると，多重度「1」側のインスタンスは必ず必要であるが，インスタンス自体は取り換えが可能である．

図 3 は，依存型の 1 対多関連とインスタンスのライフタイムとの関係をを示す．クラス A 側のインスタンスが先に存在していないと，クラス B 側のインスタンスは生成できない．そして，クラス B 側のインスタンスが存在する限り，クラス A 側のインスタンスは存在せねばならず，取り換えは効かない．結局，クラス A はクラス B に比べて時間的に先行している．

図 4 は，非依存型の 1 対多の関連のライフタイム分析結果である．この場合も，クラス A 側のインスタンスが先に存在していないと，クラス B 側のインスタンスは生成できない．結局，クラス A はクラス B に比べて時間的に先行している

以上から 1 対多関連は，結び付けられた 2 つのクラスに時間的前後関係を含意することが分かる．尚，クラス A 側のインスタンスは，クラス B 側のインスタンスと同時に生成されてもよいが，現状の計算機は，シーケーンシャルに動くので，その様な場合は考えない．

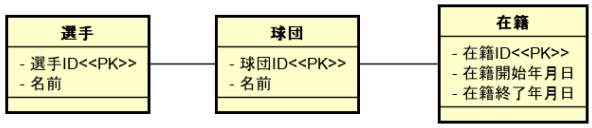

図 5　学生の犯しやすい誤り

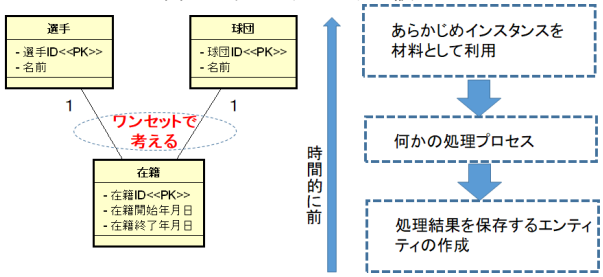

図 6　新しいクラス図の理解

3.2　時間的前後関係としてのクラス図の理解

クラス図による概念データモデリングが必ずしも普及しないのは，クラス図をどう描くかの方法論が存在しないためである．ここで，初学者の陥り易い誤りを図 5 に示す．これは，「選手 A は球団 B に 2018 年 4 月から 2019 年 3 月

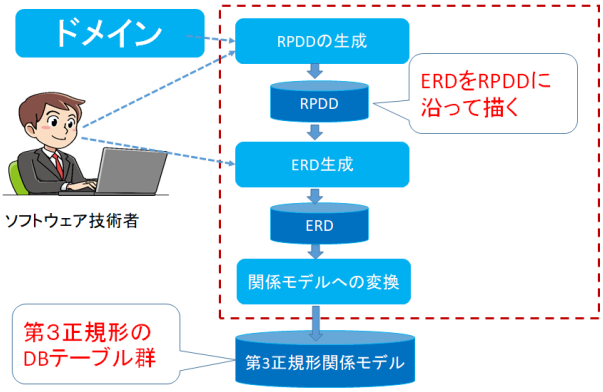

図 7　提案方式の概要

まで在籍した」という課題文に対して，学生がしばしば解答するクラス図の構造である．この図 5 は誤っている．しかし，学生は「関連は意味的関係のある所に引く」程度の方法論しか知らないので，この様に，ムード的にクラス図を描く事になる．

図 6 左側は正しいクラス図である．本稿では，このクラス図を以下の様に理解する．

- 2 つのクラス「選手」「球団」は，所属するというアクションが行われるための「材料」と見なし得る．
- 一方，クラス「在籍」は，選手が在籍したという行為（アクション）の記録であり，「選手」「球団」から見ると，「アクションの結果の記録」である．
- 2 つの関連は，アクションと対応づけられている．つまり，2 つの関連はワンセットで考えるべき．
- 上のクラスと下のクラスの間には，時間的前後関係がある．

この時間的前後関係は，前節で論じた，1 対多関連における時間的前後関係と対応している．この様はアクションは，因果関係の連鎖として，時間的につぎつぎと連鎖してゆく．そうであるなら，その様に ERM を記述すれば良いはずである．

尚，図 6 において，上方向の関連の多重度が「1」であることは重要な意味を持っている．多重度「1」であるから，「所属」クラスのインスタンスを生成する際には「選手」クラスのインスタンスと，「球団」クラスのインスタンスを必要とする．上側の 2 つのクラスの多重度を「1」と「0..1」にすると，あるプロセスの原料として，1 つのインスタンスでよい場合と，2 つのインスタンスを必要とする場合が混在してしまう．プロセスの意味が曖昧になる．

3.3　提案手法の概要

図 7 は，提案手法の概要である．モデラーは，ドメイン知識に従って，次章で説明する RPDD(リソースプロセス従属ダイヤグラム）を記述する．時間的前後関係で対象を記述するハッセ図構造を持っている．原料がつぎつぎと加工される「工場モデル」と見なせるものである．次に，RPDD に従って，その構造を写し取り，クラスの属性を追加して，ERM を完成する．

次に，ERM を自明な方法で，関係データベースのテーブルに変換する．本稿の主旨により，関係データベースは，3NF となる．以下，次の 4 章では，RPDD について説明し，その次の 5 章では，RPDD から 3NF のデータベースが生成されることを示す．6 章では，逆に，3NF のテーブル群が，ハッセ図構造をもつ ERM に変換できることを示す．

4.　ハッセ図構造を持つ ERM

4.1　RPDD(リソースプロセス従属グラフ

著者らは既に，ERM 作成のガイドラインとして，RPDD(リソースプロセス従属グラフ) を提案している [1][2][3]．図 8 には，RPDD のイメージを示す．エンティティとプロセスから構成される．プロセスは，工場における加工・中間製品の製造である．本稿では，少なくとも関連の一方の多重度は「1」である．この多重度「1」側を矢印の鏃（やじり）で示しているので，原料が流れてゆく方向と矢印の方向は逆になっている．以後，「上流側」とは，矢印の流れの方向を意味するものとする．各プロセスからは中間製品がそれぞれ出力側の矢印により指定されている．ただし，矢印は，中間製品からプロセス側を向いている．

エンティティには 2 種類ある．矢印がどこにも出てゆかない**独立型エンティティ**と，中間生成物を出力する先である**従属型エンティティ**である．独立型エンティティでは，他のエンティティのインスタンスのデータ状態に無関係に，インスタンスを追加できる．従属型エンティティは，他のエンティティのインスタンスの存在があって初めて作成できる．

4.2　RPDD が持つべき条件

RPDD・ERM は，既約有向非巡回グラフ（DAG) でなければならない．以下の条件 1，条件 2 を満たす必要があ

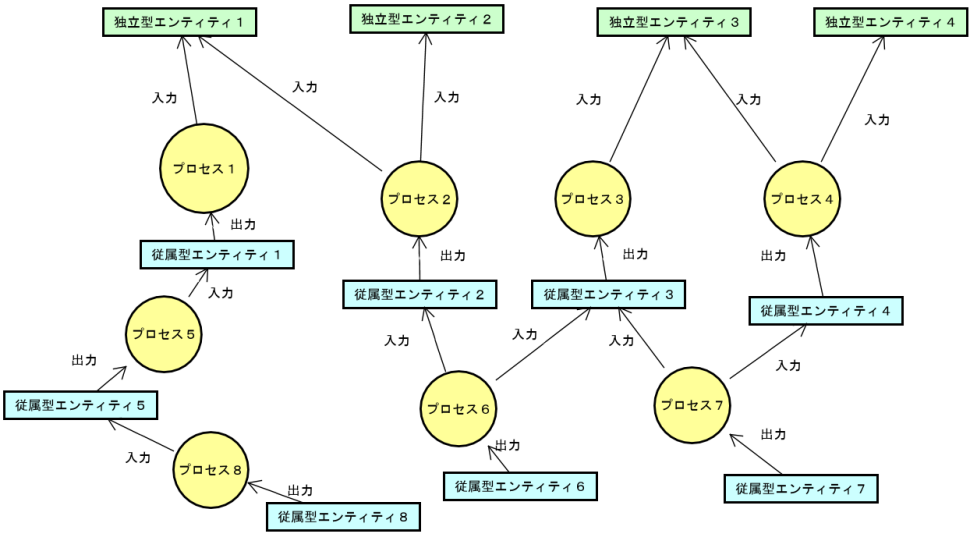

図 8　RPDD(リソースプロセス従属ダイヤグラム

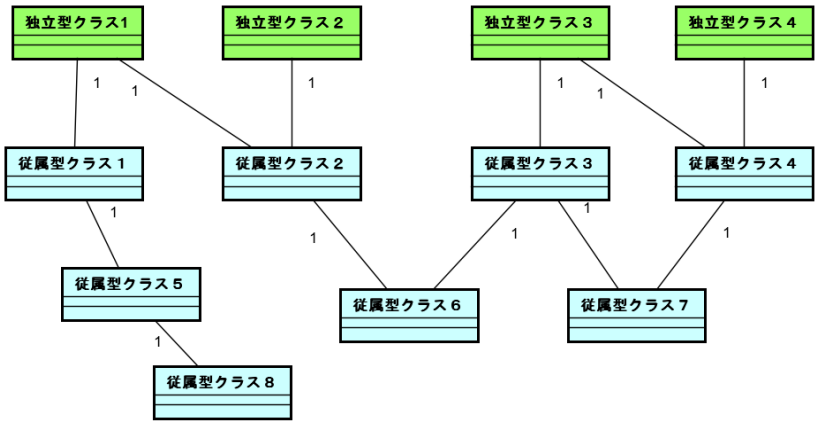

図 9　生成されたクラス図

る．

【条件 1】RPDD におけるプロセスの既約性

RPDD においては，関連に該当する「プロセス」は，対象ドメインにおいて，それ以上，細分化できない基本的なものでなければならない．即ち，入力と出力の因果関係は直接的であることが要求される．

【条件 2】RPDD におけるプロセスの入力数は最大 2

プロセスの入力数が 3 以上となる場合でも，2 入力のプロセスを連鎖することにより，プロセスを実現できる．例えば，学校の時間割作成では，まず，科目を決めて教員との関係を決め（2 入力プロセス），決まった科目と時間割の関係を決め（2 入力プロセス），最後に教室を割り当てている（1 入力プロセス）．

RPDD の構造は，図 9 の様に，そのまま ERM に写し取る．本稿のねらいは，この ERM から生成された関係モデルの正規性である．ERM の作成では，以下の条件が成立する様に設計する．

【条件 3】PK はオブジェクト ID のみ

本稿では，エンティティの PK は，唯一の属性であるオブジェクト ID のみを考える．

【条件 4】エンティティは 3NF に正規化

エンティティの属性は 3NF に正規化されていることを前提とする．

4.3　RPDD と ERM の例

図 10 には，RPDD の作成例を示した．この事例は，渡辺幸三による著作に記載されていた ER 図を参考にしている [5]．業務の時間的流れが，そのまま，RPDD 上に実現されている．

また，図 11 には，RPDD から写し取った ERM の構成例を示した．RPDD の構造は，そのまま直接的に ERM に変換できる．

5.　ハッセ図から関係モデルへ

RPDD をそのまま写し取り作成した，上記条件 1~4 を満足する ERM を，表 1 に示した方法で，関係モデルのテーブル群に変換する．図 9 の ERM を見ると，関連の一方の多重度は「1」であるので，図 2 の様な，関連を FK で表

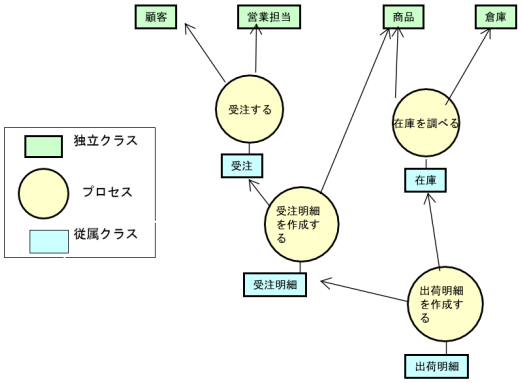

図 10 RPDD の例

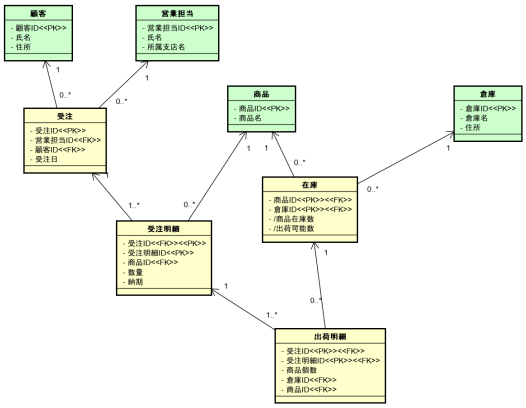

図 11 生成されたクラス図

現したテーブルに変換できる．問題は，この変換後のテーブルが 3NF を満たすかどうかである．3NF であるためには，変換後のテーブルは，以下の 3 つの要件を満たす必要がある．

(R1) すべてのテーブルが，スカラ値のみを持ち，

(R2) すべての非キー属性が，すべての候補キーに対して（部分従属ではなく）完全従属し，

(R3) 非キー属性の全てが候補キーに非推移的に関数従属する．

ただし，本稿の議論では，候補キーは，オブジェクト ID のみである．このため，上記 R2 の部分従属と完全従属の区別はない．以下の結果 1 が得られる．

【結果 1】関連を表現する FK を属性値とする属性を含めて，生成されたテーブルは 3NF を構成する．

以下，これを証明する．最初に，関連を表現する属性を除いたテーブルについて考える．ここでは，エンティティの属性をそのままテーブルにコピーしている．エンティティの属性は，上記の R1，R2，R3 満たす．よって関連を表現した属性を除くテーブルは 3NF である．

次に関連を表現している属性について考える．この場合，R1 は満たされている．FK はひとつの数値に過ぎないからである．ただし，実際には FK はポインタであり，ポイントしている先にはオブジェクトがあり，構造を持つので，アトミックではないとの議論はあり得ると思われるが，本稿では，その立場は取らない．

関連を表現する属性における，R2 の成立を調べる．関連を表現する属性では，PK が一つ決まると FK がひとつ決まる．PK が無い（タプルを消した）時には，FK は存在しない．テーブル上では，FK を入れる属性において NULL 値は許されない．PK から FK への関数従属性は明らかである．PK に相当するインスタンスが決まると，FK に相当するインスタンスが決まるが，RPDD の主旨から，この関係は，直接的因果関係である．これは，非推移的とみなしてよい．よって，R2,R3 が満たされる．

以上から，生成される関係モデルのテーブルは，関連を表現する属性を含めて，3NF を構成する．

6. 逆変換の妥当性

次に，関係モデルの 3NF のテーブル群を逆変換すると，ハッセ図になるか否かを分析する．上記の結果 1 から条件 1~4 を導く問題である．但し，与えられるテーブル群は以下の条件を満たしているものとする．

（C1）PK は単一属性である．

（C2）関連は，関連を表す属性の FK として表現されるが，関連を表す属性の数は，最大 2 個とする．

（C3）テーブルは関連を表す属性を含めて，3NF である．

ERM への変換は，図 1，図 2 に示した自明な方法を逆に利用する．テーブルの関連を表す属性以外の部分はそのまま ERM のエンティティに変換されるので，エンティティは 3NF となる．

問題は，関連を表す属性である． FK で表現されるリンク先は，関連 1 つに対して，1 個が限定であり，NULL は許されない．このため，関連リンクの多重度は「1」でなければならない．

更に，関連で結ばれたエンティティ間には，半順序の時間的前後関係がある．FK を格納するためには，FK を PK として有するエンティティのインスタンスが実在している必要がある．これは時間的前後関係を構成する．ただし，時間的前後関係であるので，半順序がループとなることは物理的にあり得ない．よって，エンティティと関連は，有向非巡回グラフ（DAG）を構成する．

加えて，テーブルにおいて，関連を表現する属性が持つべき条件として，PK で表現されるインスタンスから FK で表現されるインスタンスは，直接的で推移性の無いものでなければならない．その意味で，PK で表現されるインスタンスから FK で表現されるインスタンスへは直接的な関係でなければ，テーブルの 3NF とはならない．よって，DAG は既約となり，逆生成される ERM は，ハッセ図でなければならない．以上から，RPDD により作成された ERM と 3NF を満たすテーブル群は等価である．

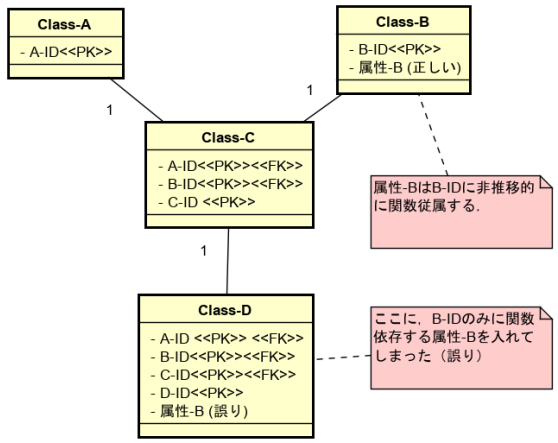

図 **12**　BCNF の説明図 1

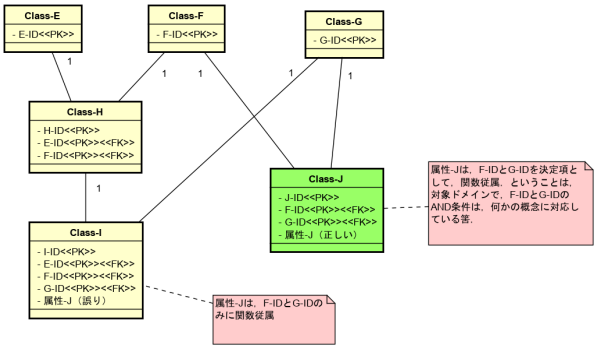

図 **13**　BCNF の説明図 2

7. 議論

前章までは，正規形は 3NF とした．エンティティの PK はオブジェクト ID であるので，3NF を越える正規性は，そのままでは議論にならない．しかし，RPDD が正しく作られていると，本稿の手法により生成された関係モデルは，結果的に，第 5 正規形までもその主旨としては満足していると考えられる．以下，その説明を行う．

以上の議論では，各エンティティは，PK としてオブジェクト ID を持つ．しかし，高度な正規形を議論するには，より詳細にエンティティの特性を表す PK が必要である．RPDD の構造から，各エンティティのインスタンスを作るための材料は明確である．着目しているエンティティから見て上流に存在するすべてのエンティティのインスタンスが材料である．そこで，各エンティティの PK は，上流に存在する全エンティティのインスタンスが持つオブジェクト ID（群）及び自分自身のオブジェクト ID から成る属性集合（属性群）から構成されると考える．問題は，このような属性群を PK として持つ時に，ボイスコッド正規形，第 4 正規形，第 5 正規形が成立するかどうかである．

7.1 ボイスコッド正規形 (BCNF)

生成されたテーブルが BCNF であることを確認する．あるエンティティの非キー属性が，当該エンティティの PK を構成する属性群の真部分集合に関数従属したとする．

まず，真部分集合が PK 中の 1 属性であるときを考える．図 12 を見ても分かる様に，クラス D の PK 属性の 1 つ 1 つについては，必ず上流側に相当するエンティティが存在する．従って，「クラス D の属性-B」は，その上流側のエンティティ (クラス B) で属性として表現されているはずで，エンティティの 3NF 設計が誤っていることになるので，矛盾である．

次に，非キー属性が，PK を構成する属性群の 2 個以上の属性群に関数従属した時を考える．図 13 において，左下に示した「クラス-I の属性-J」が，PK 属性群の真部分集合に関数従属したとする．ただし，「I-ID<<PK>>」は，クラス I で新規に追加した PK 属性なので，分析には含めない．

例えば，図 13 に示した様に，クラス I における「I-ID<<PK>>」を除く属性の 2 個以上の組み合わせに対して，「属性-J（クラス I に間違っておかれたものとする）」が従属したとする．例えば，F-ID と G-ID の組み合わせに対して，関数従属したとする．これは，クラス I の上流側にはない組み合わせである（組み合わせがあれば，それに相当するエンティティがあるので，そちらに属性を付加すべきである）．

しかし，クラス F とクラス G の組み合わせは，関数従属の決定項となるくらいであるから，ひとつの概念として，対象ドメインでは重要なはずである．そうであるなら，その概念はクラス化をどこかでされており，そのクラスの属性として，クラス I の属性-J を置けば良いことになる．従って，そのようなクラスが存在しないのは，RPDD のモデル化がおかしい．以上から，提案手法によるテーブルは，RPDD が正しく作成されていれば BCNF の主旨を満たす．

7.2 第 4 正規形，第 5 正規形

第 4 正規形，第 5 正規形については，BCNF の様な証明には合致しない．しかし，RPDD では，プロセスの処理を 2 入力に制限している．このため，3 者以上のインスタンス間に一段階で関係を生成することはない．このため，3 者の関係は，まず 2 者の関係づけを行った後に行うことになる．従って，第 4 正規形，第 5 正規形を生じにくくする仕組みを持つと考えているが，この点は，今後の検証が必要である．

7.3 日本的なクラス図の描き方

本稿で提案した RPDD における独立クラスと従属クラスの区分は，椿正明による「リソースエンティティ」「イベントエンティティ」[6] の妥当性に傍証を与える．ただし，椿は 1 対多関連には言及していない．また，渡辺幸三は，ER 図を用いたモデリングにおいて，1 対多関連を多用し

ており，ER 図も，上部からつぎつぎと降りてくる様な段階的な方法を例示している [5]．この一致も，偶然であろうか．本稿の分析結果を見る限り，椿や渡辺も，生成されるデータベースの正規化を前提とする ERM 構造を指向している可能性がある．

英語圏では，クラス図を作成するに際して，デザインパタン，アナリシスパタン [7][8] のように，「対象から一般性を抽出する」ことを追究して来た．これは，現実を離れて，高い立場で対象を眺める，抽象化の世界であり，西洋文明の姿そのものである．

これに対して，本稿のアプローチは，データベースに変換すると 3NF にそのままなるような，職人的な作り込みを ERM に施している様にも見える．加えて，椿が「イベントエンティティ」と言っている様に，椿のアプローチは，オブジェクトに相当する名詞ではなく，処理プロセス＝動詞に注目している．

これらは偶然とは思えない．日本語は，主語は不要であり，動詞中心の言語である [9][10]．一方，英語は名詞が無いと始まらない言語である．英語の認知構造とクラス図の関係は，すでに著者らが論じている [11]．本稿を含めて，椿，渡辺のアプローチは，日本語に適した，ERM 構築法になっているのではないかとの印象が強い．

8. 終わりに

UML クラス図，あるいは ER 図で記述される ERM（多対多関連は含まれないものとする）を関係モデルのデータベースのテーブル群に変換する時に，テーブルが 3NF を満たすための，ERM が持つべき性質について論じた．結論的には，(1) ERM の関連としては，一方の多重度が「1」である関連のみとし，エンティティの属性が 3NF を満たす様に構成し，(2) 一方の多重度が「1」である関連を半順序とするハッセ図構造となる様に ERM を描けば，結果的に，3NF を満たすテーブル群を得られる．この様な ERM の構造と，RPDD(リソースプロセス従属グラフ) と呼ぶ．逆に，3NF を満たすデータベースのテーブル群から ERM を描けば，一方の多重度「1」である関連を半順序としてハッセ図を構成する．即ち，データベースの正規性を重要視するなら，ハッセ図構造で，ERM を作成すべきである．

本手法で作成した ERM では，エンティティとそこから出ている関連をワンセットとして，作成・削除が可能であり，アプリケーションのモジュラリティが向上する．しかも，ハッセ図構造を持つ ERM は，工場モデルであり，独立型エンティティを最初の材料・入力として，つぎつぎと加工してゆき，中間生成物は従属型エンティティとして保存される．その意味で，工場モデルは分かりやすい．その様な正規性を担保するため，ERM のひな型である RPDD を描く際には，対象ドメインの業務処理を，プリミティブな処理プロセスに充分注意深く分割する必要がある．それが，ERM から生成されたデータベーステーブルの第 3 正規形を保証する．

本稿の結果を見ると，概念データモデリングは，「対象ドメインが持っているデータ構造をそのまま写し取るもの」とは，言い難い．概念データモデリングは，対象業務を正規化分割するプロセスである．

参考文献

[1] 金田重郎，井田明男，森本悠介，「ハッセ図としてのクラス・ER 図：クラス図作成のためのガイドライン」，電子情報通信学会・知能ソフトウェア研究会，2019 年 3 月
[2] 金田重郎，井田明男，森本悠介，「ハッセ図としてのクラス図・ER 図について」，第 14 回情報システム学会全国大会・研究発表大会，2018 年 12 月
[3] 金田重郎，井田明男，森本悠介，「正規化クラス図（ER 図）作成のガイドライン －要のもの・こと間のハッセ図としてのクラス図－」，ソフトウェアエンジニアリングシンポジウム 2018 論文集,2018,pp.93-102，2018 年 9 月
[4] ジェームズ・ランボー (著)，ウィリアム・プレメラニ (著)，ウィリアム・ローレンセン (著)，マイケル・プラハ (著)，フレデリック・エディ (著)，「オブジェクト指向方法論 OMT―モデル化と設計」，トッパン，1992 年 7 月
[5] 渡辺幸三，「業務別データベース設計のためのデータモデリング入門」，日本実業出版社，2001 年 7 月
[6] 椿正明，「名人椿正明が教えるデータモデリングの “技”」，翔泳社，2005 年 11 月
[7] マーチン・ファウラー (原著)，堀内一，友野晶夫 他 (訳)，「アナリシスパターン―再利用可能なオブジェクトモデル」，ピアソンエデュケーション，2002 年 4 月
[8] エリック・ガンマ (原著)，ラルフ・ジョンソン (原著)，リチャード・ヘルム 他 (原著)，「オブジェクト指向における再利用のためのデザインパターン」，ソフトバンククリエイティブ，1999 年 10 月
[9] 金谷武洋，「日本語に主語はいらない」，講談社選書メチエ，2002 年 1 月
[10] 三上章，「象は鼻が長い―日本文法入門」，くろしお出版，1960 年 10 月
[11] 金田重郎，井田明男，酒井孝真，熊谷聡志，「日本語仕様文からの概念モデリングガイドライン－行為文と関数従属性に基づくクラス図の作成-」，電子情報通信学会論文誌 D，Vol.J98-D，No.7，pp.1068-1082，2015 年 7 月，DOI: 10.14923/transinfj.2014JDP7103
[12] 増永良文，「リレーショナルデータベース入門―データモデル・SQL・管理システム・NoSQL」，サイエンス社，2017 年 3 月
[13] マーチン・ファウラー (原著), 羽生田 栄一 (訳)，「 UML モデリングのエッセンス 第 3 版」，翔泳社，2005 年 6 月