

[研究論文]

ハッセ図構造を持つ実体関連モデルと正規形データベース Entity-Relationship-Model having Hasse-structure and Normalized Relational Database

金田 重郎[†], 井田 明男[†]
Shigeo KANEDA, Akio IDA

[†] 同志社大学理工学部

[†] Faculty of Science and Engineering, Doshisha University

要旨

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。本論文では、(1) 任意の ERM は、1 対多関連のみで構成される ERM に等価変換可能であること、(2) 1 対多関連は、関係づけられたエンティティ間に、時間的前後関係を含意すること、を提起し、ERM が持つべき構造である EADD（エンティティ・アクション従属ダイアグラム）を用いたモデリング手法を提案する。EADD は、ハッセ図構造を有し、EADD をそのまま写し取って ERM を作成し、エンティティの属性部分を第 3 正規形にしておけば、当該 ERM 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、データベースは、ハッセ図構造を持つ ERM になる。本論文は、(1) 任意の業務処理は、入力に該当するエンティティを時間的順序に従い加工してゆく「工場モデル」で表現できること、(2) 既約性を持つ EADD としてアクション群がモデル化できていれば、生成されるデータベースは第 3 正規性が保証されること、を示している。

Abstract

Conceptual data modeling extracts a data structure from a target domain by creating an entity-relationship model (ERM), which is converted into a normalized relational database model. Such normalization is a process after extracting the data structure. On the other hand, we change this perspective and clarify the required ERM structure for generating a database model with a third normal form (3NF). An ERM should be composed of (1) associations with at least one multiplicity of “1” and (2) entities whose non-key attributes are in the 3NF. Additionally, the ERM structure should be a Hasse diagram: an irreducible directed acyclic graph. We also propose an Entity Action Dependency Diagram (EADD), which is a template for simply designing our proposed ERM. Our proposed EADD/ERM structure is a necessary and sufficient condition to generate 3NF databases.

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、実体関連モデル（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。その原因は種々あるが、その一つは、実体関連モデル（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、誰が書いても同じ様な ERM が得られる、概念データ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、本論文では、ERM が利用する関連として 1 対多関連のみ¹を利用し、しかも、ERM がハッセ図構造を持つ新しいモデリング法を提案する[1][2]²。ここで 1 対多の「1」は「1..1」の意味である。特に、本論文では、1 対多関連は、関連付けられたクラス間に時間的前後関係を含意することを提起する。また、任意の ERM は、1 対多関連のみの ERM に容易に等価変換できる。この結果、任意の ERM は、時間的前後関係を半順序とするハッセ図構造（既約な有向非巡回グラフ構造）を持つこととなる。このハッセ図構造を、本論文では EADD（エンティティ・アクション従属ダイアグラム, Entity Action

[研究論文]

2020 年 4 月 1 日受付, 2020 年 6 月 8 日改訂, 2020 年 7 月 1 日受理

© 情報システム学会

¹ 正確には、片方の多重度が 1 であれば、他方は何でも良い。1 対多以外は実務では希と思われるので、本論文では、この 1 対 ANY の意味で、1 対多と呼ぶ。当然、これには 0..1 対多の関連は含まれない。

² 文献[1][2]は、著者らの文献[3][4][5]の内容をリファインしたものである。本論文は、[1][2]をジャーナル論文化したものである。

Dependency Diagram) と呼ぶ³。EADD 構造を有する ERM を関係モデルに変換すると、得られた関係モデルは第 3 正規形を満たす。以上の関係は逆も成り立つ。第 3 正規形 (Third Normal Form, 以下「3NF」) を構成している関係モデルデータベースから変換した ERM は EADD 構造となる。即ち、EADD で構造を与えた ERM は、生成される正規化データベースの制約とはならない。

本論文は、(1)入力値を次々と加工してゆく「工場モデル」を持つ EADD で任意のドメインビジネスを表現できること、(2)EADD から ERM を経由して生成されるデータベースは第 3 正規性が担保されること、を示す。

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アクションを定義し、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、本論文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 レイヤーに区分する。Entity Layer は、アプリケーションプログラム (以下、「AP」) において、データベース化されるべきエンティティ群である。本論文では、概念データモデリングに注目しているため、Entity Layer のエンティティ群がモデリング対象となる。

但し、本論文では、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⁴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。また、ERM を記述する際に、実用上は、多対多関連が出現する割合は小さく、この関連を表現するエンティティの利用によるエンティティ数増加は小さいと考えている。

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、本論文における図示ではクラス図を用いる。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[6]) 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。

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とする。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、その属性値 (FK) として格納する。

本論文の議論では、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、本論文の議論では、ひとつのクラスから他クラスに延

³ 文献[1][2][3][4][5]では EADD を RPDD と呼んでいる。より適切な表現として、本論文では EADD を用いる。

⁴ 関連のクラス化は、UML で言う「関連クラス」ではないことに注意されたい。UML 関連クラスは、関係した 2 つのオブジェクト間に唯一のインスタンスしか持たせることはできない。

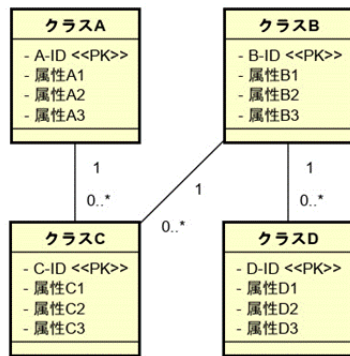


図1 クラス図の例

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 関係モデルへの変換後

びる関連の数は最大2としている。この NOT NULL 条件は、本論文では重要である。

但し、ERM では、関連の両端の多重度が、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（モジュールリティ）

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、本論文では、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、AP のモジュール化と理解する。AP の中で、エンティティ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 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 F の追加の可否に影響を与える様では困る。クラス図では、文脈自由文法の様子、各部品（本論文では、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。

3. 提案手法の概要

3.1. 1 対多関連と時間的前後関係

本論文では、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

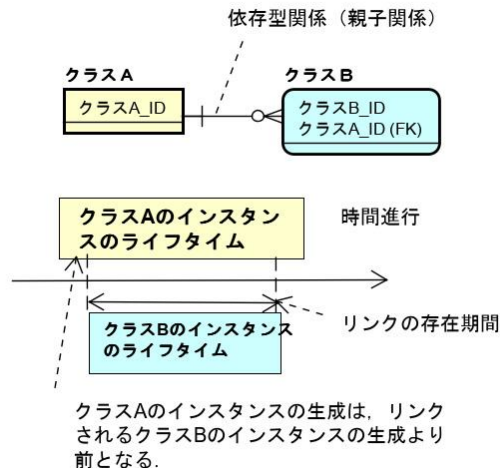


図3 1対多関連 (依存型)

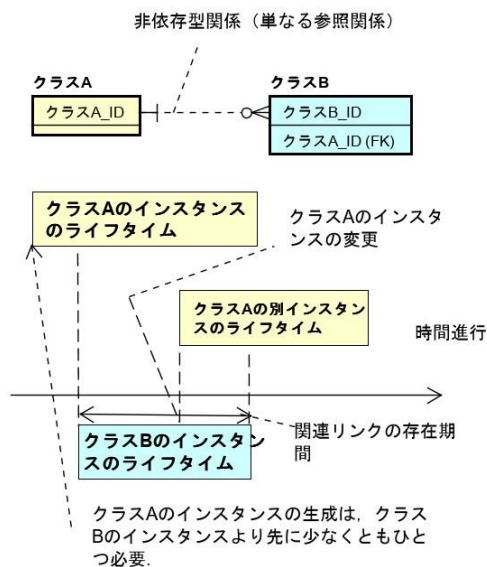


図4 1対多関連 (非依存型)

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、現状のコンピュータは、シーケンシャルに動くので、その様な場合は考えない。

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例示する。これは、「選手Aは球団Bに2018年4月から2019年3月まで在籍した」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(図5)。この図5は誤っている。なぜなら、2.3節で論じたように、「選手-球団」関係と「球団-在籍」関係との間には、本来、何の関係もないのがクラス図の大原則であるからである。このため、図5では、選手のインスタンスが1個なら問題ないが、選手を2人以上登録すると、選手と在籍の間の対応関係が分からなくなるのである。しかし、学生は「関連は意味的關係のある所に引く」程度の方法論しか知らないなので、この様に、ムード的にクラス図を描く事になる。

図6の左側は正しいクラス図である。本論文では、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であ

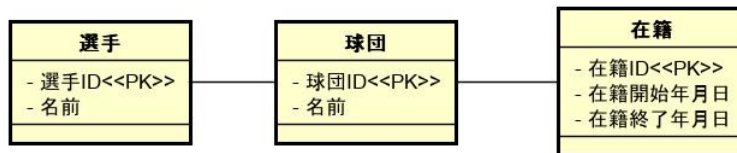


図5 誤ったクラス図の例

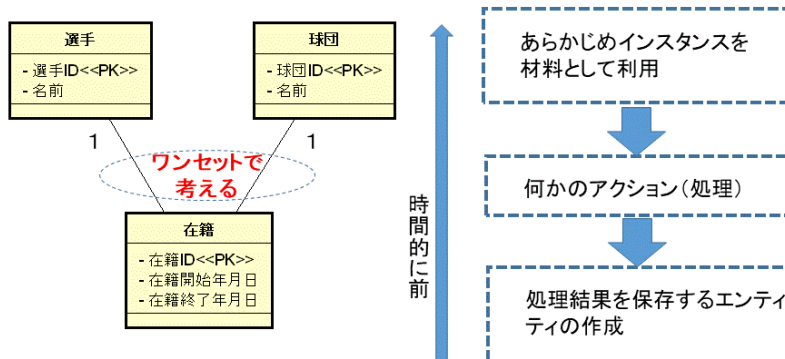


図6 新しいクラス図の理解

る。

・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。このようなアクションは、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、ドメインモデルとしてその様にERMを記述すれば良いはずである。

ここで、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、「在籍」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が事前に存在している必要がある。しかも、「選手」と「球団」クラス側の多重度が「1」であることは、2つの関連「選手-在籍」と「球団-在籍」が相互に独立ではなく、強いワンセットの関係を持つことを組み込んでいる。多重度「1」ならではの機能である。

3.3. 有向非巡回グラフ (DAG) と ERM の等価性

以上の議論から、以下の点が明らかとなる。

(1) どの様なドメインを表現したERMであっても、1対多(正確には1対ANY)関連のみのERMに容易に変換可能である。この場合、多対多関連を併用した場合に比べてエンティティ数は増加する。しかし、ERMでは多対多の利用は少なく、エンティティ数の増加は大きくないと想定される。また、多対多を利用しないことは、ERMを関係モデルに変換する上でも都合が良い。

(2) 1対多の関連は、「1」側を上流とする大小関係と見なすことができる。つまり、ERM全体は、エンティティ間に半順序関係を持つ(ノードとリンクを持つ)ネットワークと見なし得る。この大小関係は、時間的前後関係であるので、「1」側へつぎつぎと関連を辿って行った時に、この関連群が元のエンティティに戻りループを成すことは無い。もし、ループになると、どこかのリンクで、未来から過去へ飛ぶと解釈できる。これは不合理である。よってループの存在は、モデリングの誤りを示唆する。

以上から1対多関連のみで構成されるERMは、任意のドメインモデルを表現でき、有向非巡回グラフ(Directed Acyclic Graph, DAG)を構成する。DAGは、ノードとリンクで構成されるが、本論文では、ノードをエンティティ、リンクを1対多関連による時間的前後関係と考える。DAGを構成するリンク(1対多関連)は、「1」の方向に、次々と手繰ってゆけるが、やがてそれ以上、手繰って行けないノード(独立したエンティティ)に到達する。つまり、DAGは独立したエンティティが加工され、やがて、それ以上加工されないエンティティに到達して終わる様な全体構造(以下、「工場モデル」)を持つ。この構造は、1対多関連のみからなるERMにそのまま引き継がれる。即ち、どの様なドメインモデルのERMでも、工場モデルで表現できる。

3.4. 提案手法の概要

提案手法の概要フローを図7に示す。どの様なドメインモデルでも、DAG形式のERMで表現できるなら、最初からDAG形式でモデルを描き、その構造をそのままERMに写し取れば良い。そこで、本論

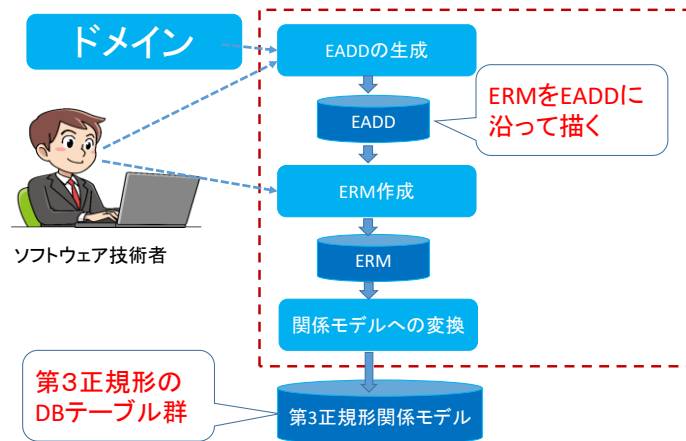


図7 提案手法の概要

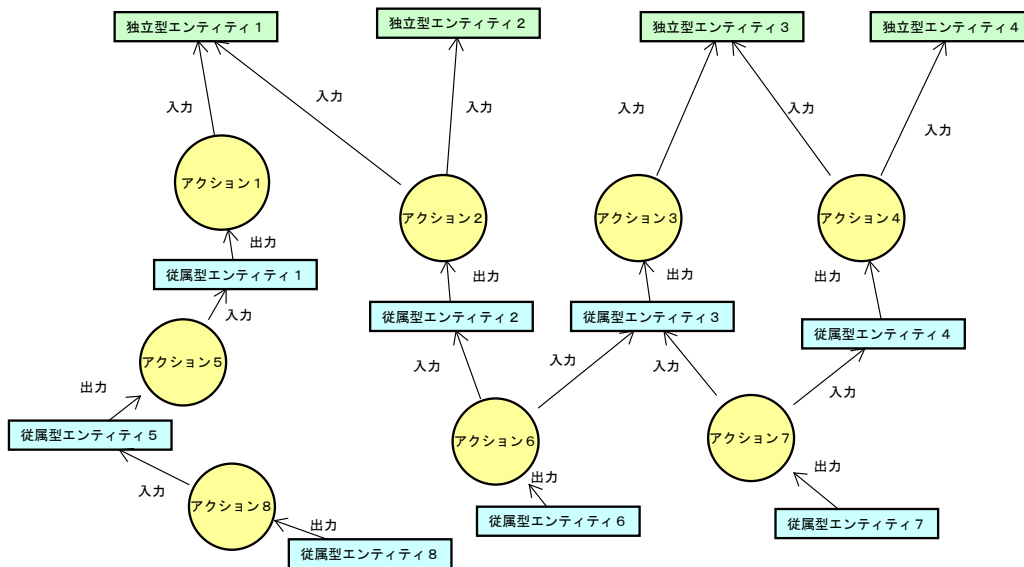


図8 EADD (エンティティ・アクション従属ダイヤグラム)

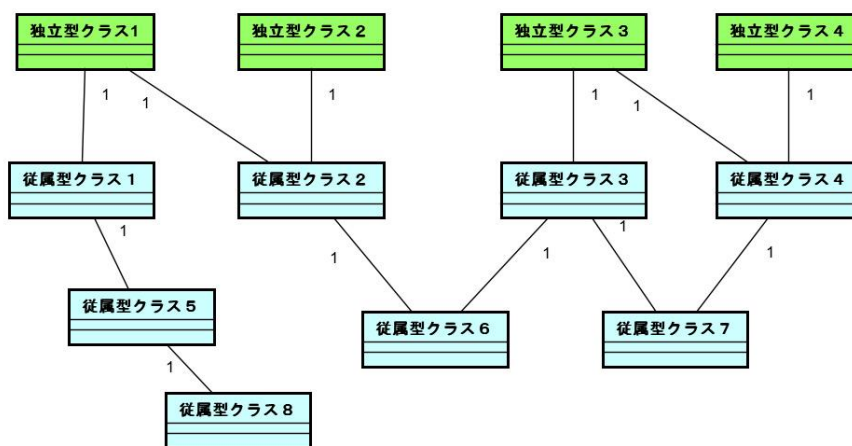


図9 生成されたクラス図

文では、モデラーは、ドメイン知識に従って、次章で説明する EADD (エンティティ・アクション従属ダイヤグラム) を記述する手法を提案する。EADD は少なくとも DAG 構造を持ち、生成される ERM の構造を定める。

次に、EADD に従って、その構造を写し取り、エンティティの属性を追加して、ERM を完成する。更に、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。ただし、単に DAG 形式でモデルを描いても、生成される関係データベースは、3NF とはならない。EADD は、ハッセ図 (既約な DAG)

でなければならない。理由は次章で示す。

以下、次の4章では、EADDについて説明し、その次の5章では、EADD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. エンティティ・アクション従属ダイヤグラム

本論文では、ERMが持つべき構造として、EADD(エンティティ・アクション従属ダイヤグラム, Entity Action Dependency Diagram)を提案する[1][2][3][4][5]。図8には、EADDのイメージを示す。エンティティとアクションから構成される。アクションは、工場における加工あるいは中間製品の製造である。本論文では、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」側を矢印の鎌(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。各アクション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、中間製品からアクション側を向いている。

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. EADDが持つべき条件

EADD・ERMは、既約有向非巡回グラフ(ハッセ図)でなければならない。これは、最終的に生成される関係モデルの正規化を保証するために必要である。以下の条件1, 条件2を満たす必要がある。

【条件1】EADDにおけるアクションの既約性

EADDにおいては、関連に該当する「アクション」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】EADDにおけるアクションの入力数は最大2

アクションの入力数が3以上となる場合でも、2入力のアクションを連鎖することにより、アクションを実現できる⁵。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入力アクション)、決まった科目と時間割の関係を決め(2入力アクション)、最後に教室を割り当てている(2入力アクション)。□

上記の条件1は、関係モデルにおける非推移的な関数従属と類似している。EADDが示す「工場モデル」は、対象ビジネスを構成するプリミティブな処理(アクション)の前後関係が半順序として与えられる限りにおいて、EADDが成立することを示唆する。たとえば、エンティティA, エンティティB, エンティティCを考える。エンティティCからは、1対多関連のリンクが、 $C \rightarrow A$ というルートと、 $C \rightarrow B \rightarrow A$ という2つのルートがモデル化されていたとする。この場合、関係モデルの非推移的関数従属と同様に、2つのルートを同時にモデル化するとモデルに冗長性が生じる。また、 $C \rightarrow A$ のみモデルに採用すると、 $C \rightarrow B \rightarrow A$ のルートが持つ情報が喪失する。この場合には、 $C \rightarrow B \rightarrow A$ を採用して、 $C \rightarrow A$ の関連を描かないことが必要である。この既約性は、DAGがハッセ図(既約なDAG)である条件そのものである。従って、EADDは、ハッセ図でなければならない。また、この既約性は、後述の様に、生成される関係モデルの第3正規性を保証する。

ただし、EADDの段階では、エンティティは属性を持たない。このためEADDに基づいて、以下の条件を守って、ERMを生成する。尚、この条件1~4で、データベースの正規性が保証されることは、後述する。

【条件3】PKはオブジェクトIDのみ

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。□

⁵ 3以上でも本提案手法が適用できる可能性はあるが、本論文では、議論を簡明化するため、2に限定する。3以上の処理についても、2個ずつの処理の組み合わせで表現できるからである。

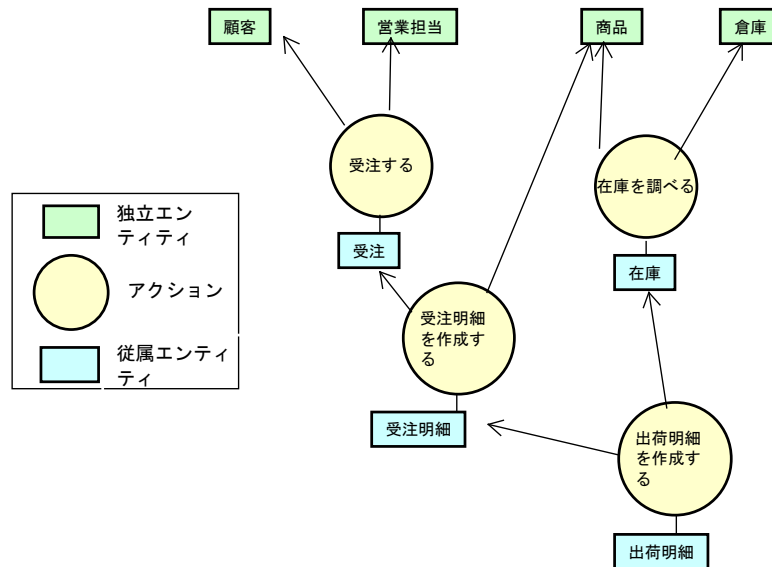


図 10 EADD の作成例

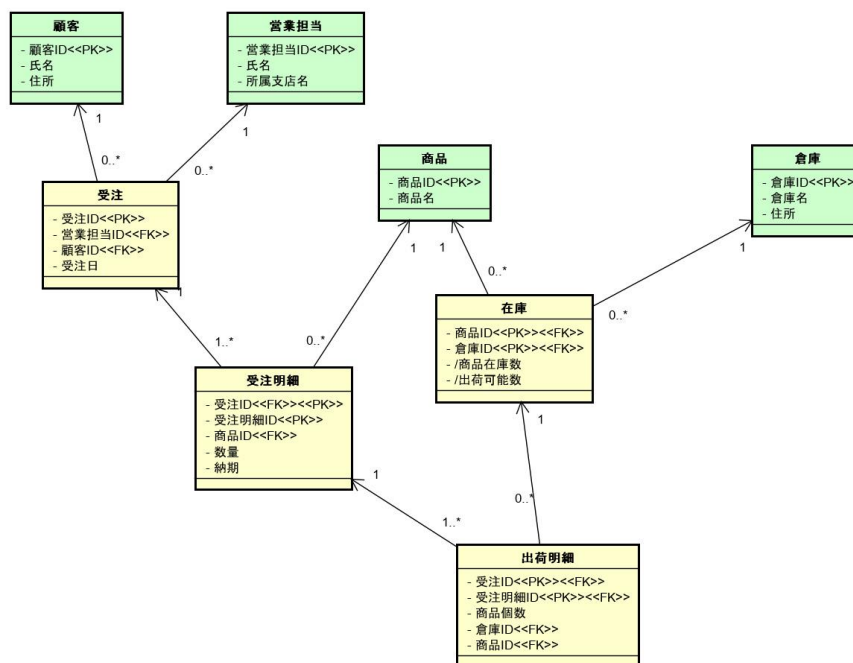


図 11 ERM の作成例

4.3. EADD と ERM の例

図 10 には、EADD 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[7]。業務の時間的流れが、そのまま、EADD 上に実現されている。また、図 11 には、EADD から写し取った ERM の構成例を示した。EADD 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

EADD 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、本論文の議論では、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、結果 1 を確認する。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 を満たす。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、構造を持っている。このため、FK はアトミックではないとの立場があり得るのかも知れない。しかし、FK でポイントしたい対象はオブジェクト全体であり、内部構造ではない。このため、本論文では、FK を入れる属性についてもアトミック性が成立しているとする。

次に、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 値は許されないこととしている。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、EADD 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。つまり、EADD から出発した ERM 作成は、ERM を変換したデータベースの正規性を保証する。

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。それが成立すれば、ERM がハッセ図となることと、3NF データベースであることは等価である。等価であるならば、EADD でモデル化することは、正規化データベースを設計する上で網羅性を持っている。任意の 3NF 正規化データベースを生成する能力を EADD が持つからである。

上記を証明するには、上記の結果 1 から 4.2 節で述べた条件 1~4 を導く必要がある。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⁶。

(C1) PK は単一属性である。

(C2) 関連は、関連を表す属性である FK 値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。また、この関連を表す属性は、「NOT NULL」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、本論文では NULL は許されない。このため、関連リンクの多重度は「1」となる。

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先に存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、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を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、

⁶ ここに示した、1) PK は単一属性、2) FK 属性の数が最大 2、とした制限は実際には必須ではないと考えている。

EADD 側を修正して複数属性を PK に許容し、3 以上の入力数を許せば、C1,C2,C3 側でもこの制限を除去できると思われる。しかし、本論文では、議論を簡明化するために、この条件を設定する。尚、C1,C2,C3 では、FK を入れる属性に対しても非推移的関数従属性を要求している。本論文の立場として、直接的な因果関係を有するテーブル間に対するのみ、FK によるリンクを設定可能であると理解している。従って、例えば、アクセス高速化等を目的とした、推移的な関数従属先であるテーブルへのジャンプは設定できない。

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、EADD は生成される関係データベースの制約にはなり得ない。任意の 3NF 正規化データベースに適用できる。

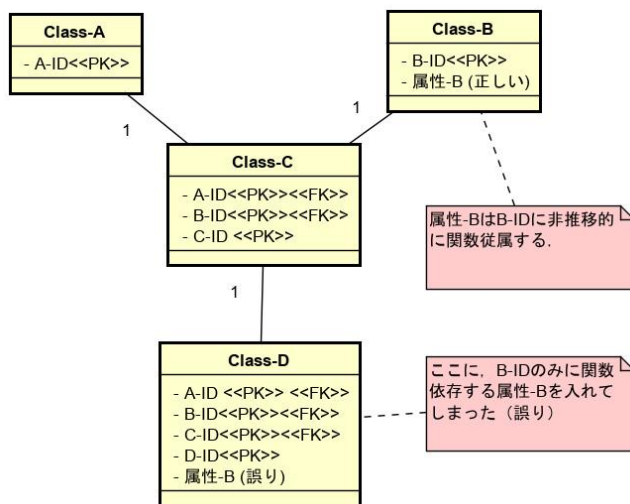


図 12 BCNF の説明図 1

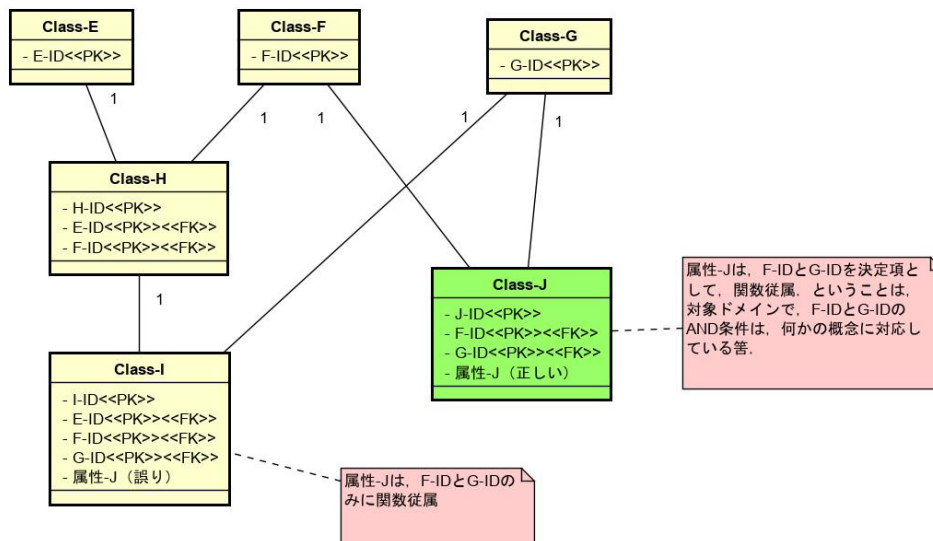


図 13 BCNF の説明図 2

7. 議論

前章まででは、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、EADD が正しく作られていると、本論文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ことが本来は望まれる。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。EADD 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。ここでの興味は、このような属性群を PK として持

⁷ ERM はクラスとオブジェクトの 2 つの側面を持つので、正確に言えば、グラフ理論というハッセ図の規約性とは異なる部分がある。詳細は付録を参照されたい。

つ時に、ボイスコード正規形，第4正規形，第5正規形が成立するかどうかである。

7.1. ボイスコード正規形(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 PK 属性を有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。よって、真部分集合が PK 中の 1 属性であるときには、BCNF の条件を満たす。

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-J 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位であるから、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、EADD 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、EADD 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 の様な証明はできない。しかし、本論文の EADD では、アクションの処理を 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. 既存手法との関係

本論文で提案した EADD 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」[8]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[7]。この一致も、偶然であろうか。本論文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様に見える。

英語圏では、クラス図を作成するに際して、デザインパターン、アナリシスパターン[9][10]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、本論文のアプローチは、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、処理アクション=動詞に注目している。日本語は、主語は不要であり、動詞中心の言語である[11][12]。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[13]。本論文を含めて、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

概念データモデリングにおける ERM (実体関連モデル) が持つべき構造として、エンティティ・アクション従属ダイアグラム (Entity Action Dependency Diagram, EADD) を提案した。EADD は、グラフ理論で言うハッセ図の構造を持ち、任意の ERM を表現できる。EADD の構造をそのまま写し取って作成した ERM は、1 対多関連とエンティティから構成される。ここで、EADD のアクションが、対象ドメイ

ンにおけるプリミティブなアクションであるなら、生成された ERM からは、第 3 正規形に正規化されたデータベースが得られる。

本手法で提案した EADD は、工場モデルに相当する。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。ただし、生成されるデータベースの正規性を担保するため、EADD 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アクションに充分注意深く分割する必要がある。その中で、どのアクションが非推移的かを見出す必要がある。

本論文の結果を見ると、クラス図・ER 図を用いた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行為である。その意味では、関係モデルにおける正規化作業が、モデリングでは必要になる。これが、初学者にモデリングが難しく感じられるひとつの原因かも知れない。

参考文献

- [1] 金田 重郎, 井田 明男, “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,” 第 15 回情報システム学会全国大会・研究発表大会, 2019 年 11 月.
- [2] 金田 重郎, 井田 明男, “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,” 情報処理学会, 情報システムと社会環境研究会, 2019 年 11 月.
- [3] 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.
- [4] 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.
- [5] 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 一要のもの・こと間のハッセ図としてのクラス図一,” 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.
- [6] 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,” トップラン, 1992 年 7 月.
- [7] 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.
- [8] 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.
- [9] マーチン・ファウラー(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.
- [10] エリック・ガンマ(著), ラルフ・ジョンソン(著), リチャード・ヘルム他(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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] マーチン・ファウラー(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.

著者略歴

金田 重郎 (かねだ しげお)

1974 年 3 月京都大学工学部電気第二学科卒業, 1976 年 3 月京都大学大学院工学研究科電子工学専攻修士課程修了. 1976 年 4 月日本電信電話公社・武蔵野電気通信研究所入所. 1997 年 4 月同志社大学大学院総合政策科学研究科教授・同工学部教授. 2020 年 3 月同志社大学退職. 現在, 同志社大学名誉教授, 同志社大学理工学部講師.

井田 明男 (いだ あきお)

1984 年同志社大学文学部文化学科(心理学専攻)卒業. 銀行員, 大手メーカー系 SE を経て, 2012 年に独立. 有限会社井田代表取締役, 現在に至る. 同志社大学理工学部講師.

付録

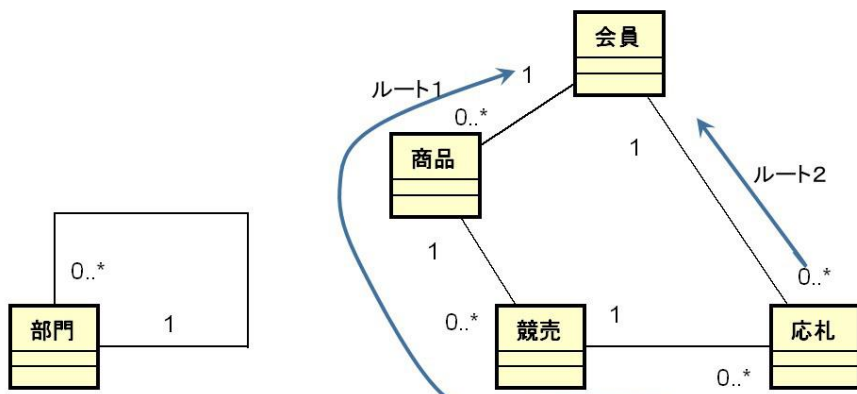
本付録では、EADDの既約性について、若干の補足を行う。本論文では、EADDの構造がハッセ図になると説明している。しかし、エンティティ間の関連は、実際には、エンティティではなく、オブジェクト間に張られている。このため、作成されたERMは、グラフ理論的な意味でのハッセ図の規約性とは多少異なる場合がある。その2つのケースについて説明する。

(1) 自己回帰型エンティティ

付図1は、エンティティが自分自身に戻ってくる関連をもつ場合である。この場合は、関連の両端に存在するオブジェクトは、異なるものである。オブジェクトは同一のエンティティ（クラス）に所属するので、自己回帰型の関連を含めて、ハッセ図上のノードとして扱うことができる。

(2) 多重ルートが存在

付図2は、あるエンティティ（「応札」）から、会員を求めるのに、2つのルートが存在する場合である。通常のERMでは、明らかにルート2の関連は、3つのプリミティブな関連に分解できるので、ルート2は、ERMから削除するべきである。しかし、付図2の場合、応札からルート1で手続られる会員（競売に商品を出した会員）と、ルート2で手続られる会員（応札した会員）は、ターゲットとなるインスタンスが異なっている。その意味で、ルート2が表現する関数従属性と、ルート1が表現する関数従属性は異なるものであり、2つのルートはERMに残すべきである。



付図1 自己回帰型関連の場合

付図2 多重ルートの存在

以上