

プログラム仕様記述のための計算指向 EAR モデル†

橋 本 正 明††

プログラム仕様記述法へ適用するための計算指向 EAR モデルを提案する。本モデルは、対象世界の情報を主体、属性および関連の枠組で捕えている EAR モデルに基づいている。EAR モデルは当初、データベース構造設計へ適用するために提案された。そのため、主体、属性および関連等の枠組を定める構造に関しては、定式化が進んでいる。その後、たとえば、トランザクション設計へも適用するための ER 動態モデル、および、データ・ディクショナリへ適用するための論理指向 ER モデル等が提案された。それに伴って、構造に課せられた条件を定める制約に関しても、定式化が進んできている。本論文では、プログラム仕様のうち、計算に関する部分を EAR モデルで表すために、以下の制約を設けた。①ある主体の属性値に基づいて、他の主体の属性値を得るための属性値従属性制約。②いくつかの主体の存在に基づいて、その主体間に存在する関連を得るための関連存在従属性制約。③ある主体の存在に基づいて、他に存在する主体を得るための主体存在従属性制約。計算指向 EAR モデルの特徴は、以上の制約を定式化したことである。本モデルはプログラム仕様記述法 PSDM へ適用した。

1. はじめに

関係モデルや階層モデル等のデータ・モデルは、データ上に対象世界の情報を表す方法を与えている。そのなかで、EAR (Entity-Attribute-Relationship) モデル^{1), 2)} は比較的自然的に対象世界の情報を表すことができる。このモデルでは、対象世界に存在する事物、および事物相互の関係をおのおの、主体 (Entity) および関連 (Relationship) という。主体および関連の集合を定めた型をおのおの、主体型および関連型という。主体の性質は属性 (Attribute) の値で表す。主体を代表する属性を主キーという。主体型、関連型、属性および主キーを合わせて構造という。構造に課せられた条件を制約という。本論文では構造と制約を合わせて情報構造と呼ぶ。情報構造は情報の枠組を定めている。

Chen は当初、データベース構造設計へ適用するために EAR モデルを提案した¹⁾。データベース構造はおもに、本モデルの構造に基づいて設計できる。そのため、本モデルの構造に関する定式化は進んでいる。一方、制約に関しては、関連で関係付けた主体間の 1 対 n や m 対 n の個数の対応関係等、データベース構造設計に必要なものを定式化している。

その後、データベース構造設計以外へも本モデルの

適用が始まった。それに伴って、制約に関する定式化が進んできている。たとえば、酒井らはデータベース構造設計のほか、データベースを更新するトランザクションの設計へも適用するために、ER 動態モデルを提案した³⁾。その制約では主体や関連の発生、変化、消滅を表す状態、および、状態を遷移させるトランザクションを用いて、主体や関連の動態を定式化した。中野はデータ・ディクショナリへ適用するために、論理指向 ER モデルを提案した⁴⁾。その制約では主体型や関連型を述語と見なし、おもに一階述語論理を用いてディクショナリ中のデータのチェック条件を定式化した。

筆者は、プログラム仕様記述法へ適用するために、以下の制約に関する特徴を持った計算指向 EAR モデルを考案した。プログラム仕様には、プログラムの入力データが表している主体や関連に基づいて、出力データが表す主体や関連を得るための計算の仕様が含まれている。本論文では、与えられた主体、その属性値、および、関連に基づいて、与えられたものとは異なる主体、属性値または関連を得ることを計算という。その計算の仕様を表すために、以下の 3 種の制約を定式化した。①ある主体の属性値に基づいて、他の主体の非主キー属性の値を得るための属性値従属性制約。②いくつかの主体の存在に基づいて、その主体間に存在する関連を得るための関連存在従属性制約。③ある主体の存在に基づいて、他に存在する主体の主キー値を得るための主体存在従属性制約。以上の制約を組合せて計算の仕様を表す。そして、与えられた主体、

† A Computation-Oriented EAR Model for Program Specification Descriptions by MASAOKI HASHIMOTO (NTT Electrical Communications Laboratories, Nippon Telegraph and Telephone Corporation).

†† 日本電信電話(株)NTT 電気通信研究所

その属性値、および、関連へ適用できるものから制約の適用を始め、制約の適用により得られた主体、属性値または関連へ適用できる制約も順次適用して、計算の目的とする主体、属性値または関連を得る。与えられたものとは異なる属性値、関連および主体を得るために、おのおの、①、②および③の制約が必要である。

本論文の目的は計算指向 EAR モデルを提案することである。本モデルはプログラム仕様記述法 PSDM⁵⁾へ適用した。以下、第2章にプログラム入出力データの情報構造、第3章に本モデルの内容、第4章に従来との比較を述べる。

2. プログラム入出力データの情報構造

売上データと顧客データを入力し、請求データを出力する簡単な請求書作成プログラムを例に取り、プログラム入出力データの情報構造の考え方を述べる。

(1) 構造

図1左下部の売上データは、同図上部に示すように、1) 主体型“製品”に属する主体“テレビ”や“ラジオ”等の属性“品名”や“単価”の値、2) 主体型“売上”に属する主体“85番売上”や“86番売上”等の属性“売上番号”や“売上数”の値、3) 主体型“顧客”に属する主体“田中”や“鈴木”等の属性“氏名”の値を表している。さらに、4) 関連型“売物”に属する関連“85番売上の売物はテレビであること”や

“86番売上の売物はラジオであること”等、5) 関連型“売先”に属する関連“85番売上の売先は田中であること”や“86番売上の売先は鈴木であること”等も表している。図1から、顧客データおよび請求データが表している主体や関連も理解できる。

入出力データが表している主体や関連は、入出力データの内容が変われば、それに応じて変化する。一方、主体型、関連型、属性および主キーからなる構造は、入出力データの内容で変わるものではなく、プログラム仕様に相当するものである。

(2) 制約

出力側の請求データが表す情報のうち、主体“顧客”は入力側の顧客データが直接表している。しかし、主体“請求”および関連“請求先”は、入力データが直接に表してはいない。このため、入力側の売上データと顧客データが表している主体、その属性値、および、関連から、主体“請求”および関連“請求先”を得るための計算が必要である。

計算方法を表すものとして、図1のように、以下の制約を設けることができる。6) 主体“売上”の売上額は、同じ主体の売上数と、その主体と関連“売物”を持つ主体“製品”の単価の積である。また、主体“顧客”の合計は、その主体と関連“売先”を持ついくつかの主体“売上”の売上額の総和である。主体“請求”の請求額は、その主体と関連“請求先”を持

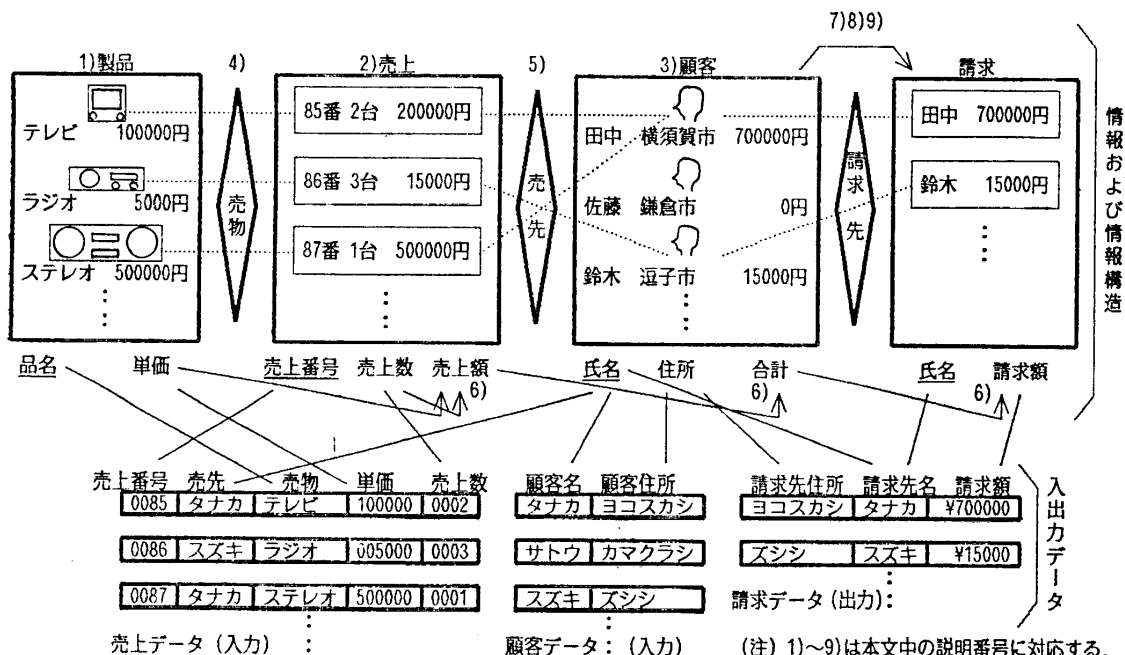


図1 プログラム入出力データの情報構造

Fig. 1 Information-structure on input-output-data of program.

つ主体“顧客”の合計に等しい。以上の条件は属性値従属性制約で表す。7) 主体“顧客”の主キー“氏名”の値と、主体“請求”の主キー“氏名”の値が等しければ、両主体間に関連“請求先”が存在する。この条件は関連存在従属性制約で表す。8) 主体“顧客”の合計が正の値ならば、その主体と関連“請求先”を持つ主体“請求”が存在する。請求の主キー“氏名”には、関連“請求先”の関連存在従属性制約に基づいて関連が存在するような値、つまり、顧客の氏名と同じ値を与える。この条件は主体存在従属性制約で表す。9) 一つの主体“請求”は必ず、一つの主体“顧客”と対応している。このため、顧客の主キー“氏名”を請求の主キーとしても使用できる。この条件は識別従属性制約で表す。本制約は計算を表すものではないが、自然な主キーを定めるのに必要である。

制約は構造に課せられた条件を表しており、構造と同じように、プログラム仕様に相当するものである。したがって、構造と制約からなる情報構造も、プログラム仕様に相当するものである。なお、売上の売上額や顧客の合計は、入出力データと直接には対応していない。しかし、計算方法を表すのに必要であれば、入出力データと直接には対応していない属性、主体型および関連型を情報構造に含めることができる。

3. 計算指向 EAR モデル

本モデルを構造と制約に分けて述べる。

3.1 構造

構造は以下のように定式化する。

(1) 主体型

主体型を E_j 、主体を e_{jt} と書く。主体型は集合であり、 e_{jt} が E_j に属することを $e_{jt} \in E_j$ と書く。

(2) 関連型

関連型は n ($n \geq 2$) 個の主体型 E_{ij} ($j=1, \dots, n$) 間の関係 (Relation) である。関連は関連型の集合の要素 ($e_{i1h}, \dots, e_{inh}$) ($e_{ijk} \in E_{ij}, j=1, \dots, n$) である。関連型を R_i 、関連を r_{ih} 、 r_{ih} が R_i に属することを $r_{ih} \in R_i$ と書く。 E_{ij} ($j=1, \dots, n$) の中に同じ主体型があってもよい。たとえば、親子関係は同じ主体型“人”の主体相互の関連である。関連“親子関係”を持つ二つの主体はおのおの、親と子の役割を持つ。役割は L_{ij} 、役割 L_{ij} を持つ E_{ij} の主体の集合は L_{ij}/E_{ij} と書く。ここで、“/”は区切り記号である。

(3) 属性

属性値の組で主体の性質を表す。各属性値は定めら

れた定義域に属する。定義域を V_a 、属性値を v_{ap} と書く。属性は主体型から定義域への関数 A_{ijk} で表す。たとえば、顧客の属性“氏名”は主体型“顧客”から、文字列の集合である定義域への関数である。 A_{ijk} が一価関数の場合、 $A_{ijk}(e_{ijt})$ は定義域中の一つの値を表す。 A_{ijk} が多価関数の場合、属性は多値であるといい、 $A_{ijk}(e_{ijt})$ は定義域の部分集合を表す。同じ主体型に属する主体は同じ属性を持つ。 A_{ijk} が E_{ij} の主体の属性であることは E_{ij}/A_{ijk} と書く。ある関連型 R_i において、役割 L_{ij} を持つ E_{ij} の主体の属性 A_{ijk} は $L_{ij}/E_{ij}/A_{ijk}$ と書く。いくつかの属性をまとめて、 $L_{ij}/E_{ij}/(A_{ij1}, \dots, A_{ijm})$ と書くこともできる。本論文では、関連は属性を持たない²⁾。

(4) 主キー

主体型の集合中で、各主体の識別に用いる属性を主体のキーという。属性の組をキーとして用いることもある。一つの主体にいくつかのキーが存在する場合、一つを選んで主キーとする。主キーは主体を代表する属性である。主体型に、その主体の属性のみからなる主キーがある場合、正規主体型という。たとえば、製品、売上および顧客にはおのおの、その主体の属性のみからなる主キー“品名”、“売上番号”、“氏名”があるので、正規主体型である。

主体型に、その主体の属性のみからなる主キーがない場合、その主体は、他の主体型に属する主体を識別の上、この識別した主体に関連で関係付けられているものとして識別する。この方法でのみ識別できる主体が属する主体型を、弱主体型という。識別の起点となる主体に関連で関係付けられている主体が複数存在する場合は、さらに、弱主体型の属性を用いて個々の主体を識別する。弱主体型においては、識別の起点となる主体の主キーの属性へ、上記のように、その弱主体型の属性を識別に用いる場合はその属性も加えたものを、主キーという。たとえば、請求の主キー“氏名”は、その意味を考えると、本来、請求の属性ではなく、請求先の顧客の主キー属性である。したがって、この例の請求は、請求の属性のみからなる主キーを持っていないので、弱主体型である。主体“請求”は、主体“顧客”を識別の上、この顧客に関連“請求先”で関係付けられているものとして識別する。

関連の主キーは、関連に含まれている主体の主キーの組からなる。たとえば、関連“売上先”の主キーは、売上の売上番号と顧客の氏名の組からなる。関連に含まれているすべての主体が正規主体型の場合、そ

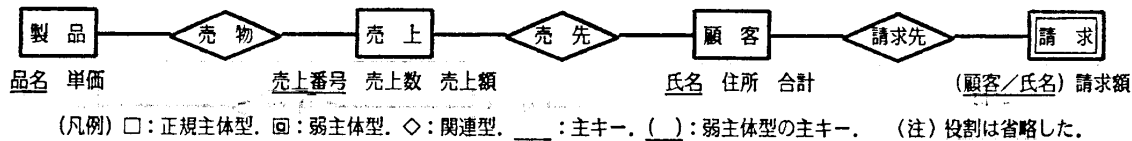


図 2 構造
Fig. 2 Structure.

の関連型を正規関連型という。弱主体型が少なくとも一つあれば、弱関連型という。たとえば、売物および売先は正規関連型である。請求先は弱関連型である。

図 1 の構造の部分を図 2 の ER (Entity-Relationship)¹⁾ 図で図示する。正規主体型や正規関連型に属する主体や関連の属性値は、図 3 の表を用いて図示する。その表には入力データが表している属性値のみを記入しているので、売上の売上額は空欄である。なお、その表は ER 図に含めない。弱主体型と弱関連型の表については、次節に述べる。

品名	単価
テレビ	100000
ラジオ	5000
ステレオ	500000
⋮	⋮

製品 品名	売上 売上番号
テレビ	85
ラジオ	86
ステレオ	87
⋮	⋮

売上番号	売上数	売上額
85	2	
86	3	
87	1	
⋮	⋮	⋮

(注1)主体型の表の各欄は属性を表す。各行は個々の主体を表す。(注2)関連型の表の各欄は主体の主キーを表す。各行は個々の関連を表す。(注3)表内の値は属性値を表す。

図 3 正規主体型と正規関連型の表

Fig. 3 Tables of regular entity type and regular relationship type.

3.2 制 約

制約は以下のように定式化する。

(1) 識別従属性制約

相異なる n ($n \geq 2$) 個の L_{ij}/E_{ij} ($j=1, \dots, n$) 間の弱関連型を R_i とする。 L_{ij}/E_{ij} ($j=1, \dots, n$) のうち、弱主体型からなるものを L_{iw}/E_{iw} とし、 E_{iw} とは異なる主体型からなるものの組を $L'_{ij'}/E'_{ij'}$ ($j'=1, \dots, n', n' \leq n-1$) とする。各 $L'_{ij'}/E'_{ij'}$ ($j'=1, \dots, n'$) に関して、その主体を一つずつ識別の上、その n' 個の主体に、 R_i の関連で関係付けられている L_{iw}/E_{iw} の主体を識別する場合、 L_{iw}/E_{iw} は R_i に基づいて、 $L'_{ij'}/E'_{ij'}$ ($j'=1, \dots, n'$) に識別従属するといひ、以下の制約式で表す。

$$L_{iw}/E_{iw} \xrightarrow{ID} R_i L'_{i1}/E'_{i1}, \dots, L'_{in'}/E'_{in'} \quad (1)$$

ここで、ID は識別従属性制約を表す。本制約は弱関連型 R_i の性質である。一つの弱主体型に対して、本制約を一つ設ける。 E_{ij} ($j=1, \dots, n$) 内の相異なる主体型に関しては、役割を省略しても本制約はあいまいにはならないので、役割を省略してもよい。次項以降の制約についても同じである。

L_{iw}/E_{iw} の主キーを $L'_{ij'}/E'_{ij'}/A_{ij'k_{j'}}$ ($j'=1, \dots, n'$), $L_{iw}/E_{iw}/A_{iwk_w}$ とし、主キー値が $v_{ij'k_{j'}}$ ($j'=1, \dots, n'$), v_{iwk_w} の主体 e_{iwI_w} は、 L_{iw}/E_{iw} の部分集

合 $\{e_{iwI_w} \mid (e_{i1I_1}, \dots, e_{in'I_{n'}}, e_{iwI_w}) \in R'_i \text{ かつ } e_{ij'I_{j'}} \in L'_{ij'}/E'_{ij'} (j'=1, \dots, n') \text{ かつ } A_{ij'k_{j'}}(e_{ij'I_{j'}}) = v_{ij'k_{j'}} (j'=1, \dots, n')\}$ の中で、 $A_{iwk_w}(e_{iwI_w}) = v_{iwk_w}$ となる主体として識別する。ここで、 R_i から $L'_{ij'}/E'_{ij'}$ ($j'=1, \dots, n'$), L_{iw}/E_{iw} のみを残す射影 (projection) 演算で R'_i が得られるものとする。

前節で述べたように、弱主体型の主キーには、その弱主体型が識別従属する先の主体型の主キー属性を含めている。したがって、属性の書き方に一貫性を持たせるため、弱主体型 E_{iw} が識別従属する先の主体型の主キー属性を、 $L_{iw}/E_{iw}/(A_{iw1}, \dots, A_{iwnw})$ に含めて書くことができ、 A_{iwk_w} の代りに $(E'_{ij'}/A_{ij'k_{j'}})$ の形で書くものとする。たとえば、主キー“(顧客/氏名)”を持つ弱主体型“請求”は、弱関連型“請求先”に基づいて、正規主体型“顧客”へ識別従属する。弱主体型と弱関連型に属する主体や関連の属性値は、図 4 の表を用いて表現する。

(2) 属性値従属性制約

相異なる n ($n \geq 2$) 個の L_{ij}/E_{ij} ($j=1, \dots, n$) 間の関連型を R_i とする。 L_{ij}/E_{ij} ($j=1, \dots, n$) のうち、ある一つを L_{iv}/E_{iv} とし、ある組を $L'_{ij'}/E'_{ij'}$ ($j'=1, \dots, n', n' \leq n$) とする。 R_i から L_{iv}/E_{iv} , $L'_{ij'}/E'_{ij'}$ ($j'=1, \dots, n'$) のみを残す射影演算で R'_i が得られるものとする。 L_{iv}/E_{iv} の任意の主体を e_{ivI_v} とし、

e_{ivl_v} を含む R_i の任意の関連を $(e_{ivl_v}, e_{i1l_1}, \dots, e_{in'l_{n'}})$ ($e_{ij'l_{j'}} \in L'_{ij'}/E'_{ij'}, j'=1, \dots, n'$) とする。 e_{ivl_v} の主キーを除くある属性を $L_{iv}/E_{iv}/A_{ivk_v}$ とし、各 $e_{ij'l_{j'}}$ の属性のある組を $L'_{ij'}/E'_{ij'}/(A_{ij'1}, \dots, A_{ij'm_{j'}})$ とする。要素 $(e_{i1l_1}, \dots, e_{in'l_{n'}})$ からなる集合を変域とし、各 $e_{ij'l_{j'}}$ の属性 $L'_{ij'}/E'_{ij'}/(A_{ij'1}, \dots, A_{ij'm_{j'}})$ の値のみを演算に用いるある関数 F_v に対して、 $A_{ivk_v}(e_{ivl_v}) = F_v(\{(e_{i1l_1}, \dots, e_{in'l_{n'}}) | (e_{ivl_v}, e_{i1l_1}, \dots, e_{in'l_{n'}}) \in R_i\})$ となる場合、 $L_{iv}/E_{iv}/A_{ivk_v}$ は R_i および F_v に基づいて、 $L'_{ij'}/E'_{ij'}/(A_{ij'1}, \dots, A_{ij'm_{j'}})$ ($j'=1, \dots, n'$) に属性値従属するといひ、以下の制約式で表す。

$$L_{iv}/E_{iv}/A_{ivk_v} \xrightarrow[\frac{VD}{R_i}]{F_v} L'_{i1}/E'_{i1}/(A_{i11}, \dots, A_{i1m_1}), \dots, L'_{in'}/E'_{in'}/(A_{in'1}, \dots, A_{in'm_{n'}}) \quad (2)$$

ここで、VD は属性値従属性制約を表す。本制約は関連型 R_i の性質である。 e_{ivl_v} の属性 $L_{iv}/E_{iv}/A_{ivk_v}$ が、他の主体の属性のほか、同じ主体の他の属性にも属性値従属する場合、 L_{iv}/E_{iv} を $L'_{ij'}/E'_{ij'}$ ($j'=1, \dots, n'$) に含めて、(2) 式を書く。

売上の属性“売上額”は、関連型“売物”，および、積を取る関数に基づいて、同じ主体の属性“売上数”と製品の属性“単価”に属性値従属する。顧客の属性“合計”は、関連型“売先”，および、総和を取る関数に基づいて、売上の属性“売上額”に属性値従属する。前者の制約では図5のように、関連“売物”により一対一に関係付けられた売上と製品の二つの主体に関して、売上数と単価の積を取り、売上額の値とする。後者の制約では図6のように、関連“売先”により主体“顧客”と1対 n に関係付けられた n 個の主体“売上”に関して、売上額の総和を取り、合計の値とする。このように、主体間の個数の対応関係を考慮して、関数 F_v を定めなければならない。

属性値従属性制約が、同じ主体の属性のみの間で成り立つ場合、以下の制約式で表す。

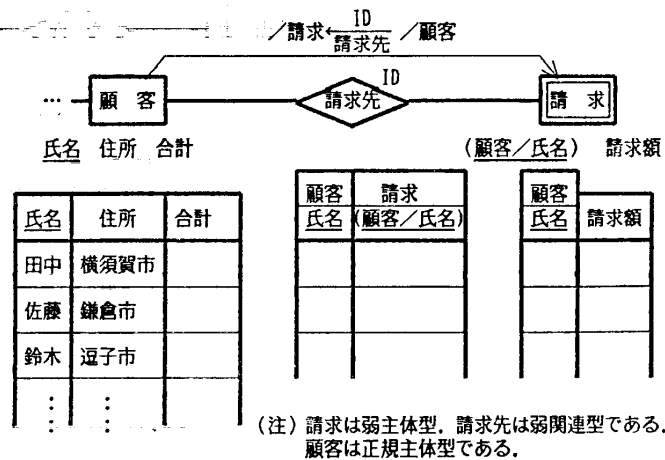


図4 識別従属性制約および弱主体型と弱関連型の表
Fig. 4 Identification dependency constraint, tables of weak entity type and weak relationship type.

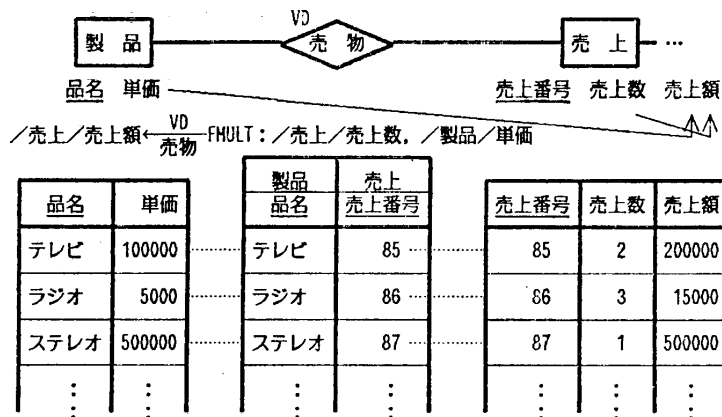


図5 属性値従属性制約 (一対一対応)
Fig. 5 Attribute value dependency constraint (1:1 mapping).

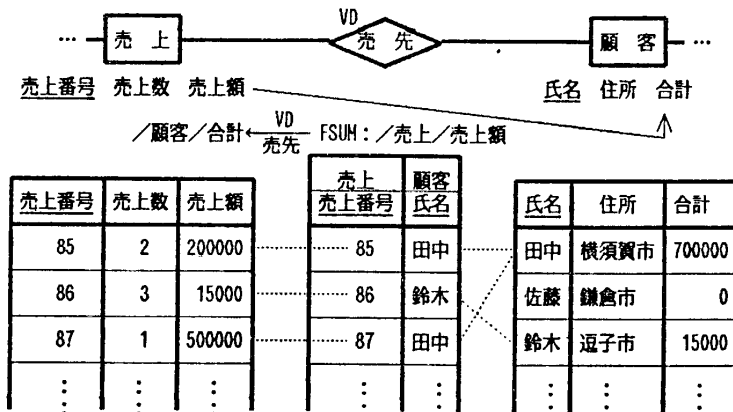


図6 属性値従属性制約 (一対 n 対応)
Fig. 6 Attribute value dependency constraint (1:n mapping).

$$A_{jv} \xrightarrow{VD} F_v : A_{j1}, \dots, A_{jm} \quad (3)$$

ここで, $A_{jv}, A_{jk} (k=1, \dots, m, k \neq v)$ は E_j の属性である. 本制約は主体型 E_j の性質である. 一つの属性に対して, (2)式または(3)式の属性値従属性制約をただだか一つ設ける.

(3) 関連存在従属性制約

相異なる $n (n \geq 2)$ 個の $L_{ij}/E_{ij} (j=1, \dots, n)$ 間の関連型を R_i とする. R_i は(4)項に述べる主体存在従属性制約を持たないものとする. 各 L_{ij}/E_{ij} の主体 e_{ij1j} に関して, 少なくとも主キーを含む属性のある組を $L_{ij}/E_{ij}/(A_{ij1}, \dots, A_{ijm_j}) (m_j \geq 1)$ とし, その値を $v_{ij1}, \dots, v_{ijm_j}$ とする. 各 e_{ij1j} の属性の他のある組を $L_{ij}/E_{ij}/(A'_{ij1}, \dots, A'_{ijm'_j}) (m'_j \geq 0)$ とし, その値を $v'_{ij1}, \dots, v'_{ijm'_j}$ とする. 関連に含まれる主体相互の対応条件を示すある述語 $P_e(v_{i11}, \dots, v_{i1m_1}, \dots, v_{in1}, \dots, v_{inm_n})$, および, 関連の存在条件を示すある述語 $P_e(v'_{i11}, \dots, v'_{i1m'_1}, \dots, v'_{in1}, \dots, v'_{inm'_n})$ がともに真ならば, $(e_{i11}, \dots, e_{in1}) \in R_i$ となる場合, R_i は P_e と P_e に基づいて, $L_{ij}/E_{ij} (j=1, \dots, n)$ に存在従属するといひ, 以下の制約式で表す.

$$R_i \xrightarrow{RD} P_e : L_{i1}/E_{i1}/(A_{i11}, \dots, A_{i1m_1}), \dots, \\ L_{in}/E_{in}/(A_{in1}, \dots, A_{inm_n}) \\ \text{if } P_e : L_{i1}/E_{i1}/(A'_{i11}, \dots, A'_{i1m'_1}), \dots, \\ L_{in}/E_{in}/(A'_{in1}, \dots, A'_{inm'_n}) \quad (4)$$

ここで, RD は関連存在従属性制約を表す. 本制約は関連型 R_i の性質である. 一つの関連型に対して, 本制約をただだか一つ設ける. $m'_j=0$ の場合, P_e は e_{ij1j} の属性値を反映しない. P_e が恒真ならば, if 以降を書かない. たとえば, 関連型“請求先”は, 顧客の主キー“氏名”と請求の主キー“(顧客/氏名)”が等しいという対応条件に基づいて, 主体型“顧客”と“請求”に存在従属する. 主体“人”が主キー“氏名コード”と非主キー属性“年齢”を持ち, 年齢が等しければ, 人相互間に関連“同年”が存在する場合, 年齢が等しいことは関連の存在条件である.

(4) 主体存在従属性制約

相異なる $n (n \geq 2)$ 個の $L_{ij}/E_{ij} (j=1, \dots, n)$ 間の関連型を R_i とする. R_i は関連存在従属性制約を持つものとする. $L_{ij}/E_{ij} (j=1, \dots, n)$ を二つに分けた組を $L'_{ij'}/E'_{ij'} (j'=1, \dots, n', n' \geq 1), L''_{ij''}/E''_{ij''} (j''=1, \dots, n'', n'' \geq 1, n''=n-n')$ とする. 両者に同じものは含まない. 各 $L'_{ij'}/E'_{ij'}$ の主体 $e'_{ij'h}$ の属性のある組を $L'_{ij'}/E'_{ij'}/(A_{ij'1}, \dots, A_{ij'm_{j'}}) (m_{j'} \geq 0)$ とし, そ

の値を $v_{ij'1}, \dots, v_{ij'm_{j'}}$ とする. $e'_{ij'h} \in L'_{ij'}/E'_{ij'}$ ($j'=1, \dots, n'$) に対して, 主体の存在条件を示すある述語 $P_e(v_{i11}, \dots, v_{i1m_1}, \dots, v_{in'1}, \dots, v_{in'm_{n'}})$ が真ならば, R_i の関連存在従属性制約における主体相互の対応条件と, 関連の存在条件を示す両述語が真となることを方程式と見なし, この方程式へ $e'_{ij'h} \in L'_{ij'}/E'_{ij'}$ ($j'=1, \dots, n'$) の属性値を代入する. そして, $e''_{ij''h} \in L''_{ij''}/E''_{ij''}$ ($j''=1, \dots, n''$) の属性に関して方程式を解いて得られた属性値を持つ主体 $e''_{ij''h}$ ($j''=1, \dots, n''$) がおのおの, $L''_{ij''}/E''_{ij''}$ ($j''=1, \dots, n''$) に属し, かつ, $e'_{ij'h} \in L'_{ij'}/E'_{ij'}$ ($j'=1, \dots, n'$), $e''_{ij''h} \in L''_{ij''}/E''_{ij''}$ ($j''=1, \dots, n''$) を含む関連 r_{ih} が, R_i に属する場合, $L'_{ij'}/E'_{ij'} (j'=1, \dots, n')$ は R_i と P_e に基づいて, $L'_{ij'}/E'_{ij'} (j'=1, \dots, n')$ に存在従属するといひ, 以上の制約式で表す.

$$L'_{i1}/E'_{i1}, \dots, L'_{in'}/E'_{in'} \xrightarrow{ED} L'_{i1}/E'_{i1}, \\ \dots, L'_{in'}/E'_{in'} \\ \text{if } P_e : L'_{i1}/E'_{i1}/(A_{i11}, \dots, A_{i1m_1}), \\ \dots, L'_{in'}/E'_{in'}/(A_{in'1}, \dots, A_{in'm_{n'}}) \quad (5)$$

ここで, ED は主体存在従属性制約を表す. 関連存在従属性制約の条件が成立することを方程式と見なすのは, その制約により, 関連を存在させるためである. 方程式において, どの主体型の属性値を代入し, どの主体型の属性に関して方程式を解くかは, (5)式の if 以前の項で示す. 方程式は解くことができ, その解法は与えられているものとする. 関連存在従属性制約における主体相互の対応条件を示す述語は, $L''_{ij''}/E''_{ij''} (j''=1, \dots, n'')$ の主キー属性を含んでいるので, 方程式から, 少なくとも, $e''_{ij''h}$ の主キー値を得ることができる. 主キーのある属性に関して, 複数の値が得られる場合, その主キーをもつ主体が複数存在することになる. 方程式から得ることのできない $e''_{ij''h}$ の属性に関しては, 属性値従属性制約を設ける. 主体存在従属性制約は関連型 R_i の性質である. 一つの関連型に対しては, 本制約をただだか一つ設ける. 一つの主体型に対しては, 異なる関連型上の本制約を複数設けてもよい. (5)式において, $m_{j'}=0$ の場合, P_e は $e'_{ij'h}$ の属性値を反映しない. P_e が恒真ならば, if 以降を書かない. $L'_{ij'}/E'_{ij'} (j'=1, \dots, n')$ 内に弱主体型があれば, ①弱主体型の主体が識別従属する先の主体, および, ②識別従属先の主体との間の関連も, とともに存在するものとする. なお, 弱主体型の主体の主キー値が定まれば, ①の主体

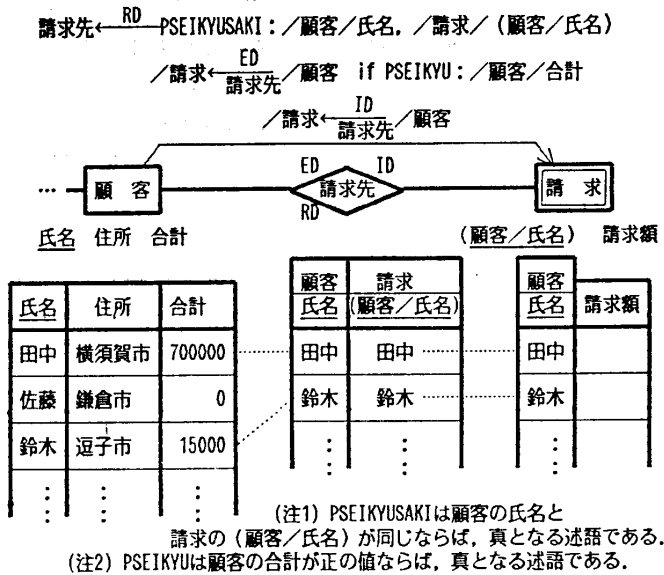


図 7 関連存在従属性制約および主体存在従属性制約
 Fig. 7 Relationship existence dependency constraint and entity existence dependency constraint.

の主キー値も定まる。①の主体と②の関連は、弱主体型の主体の識別に必要である。

主体型“請求”は、関連型“請求先”，および、顧客の属性“合計”が正の値であるという存在条件に基づいて、主体型“顧客”に存在従属する。本制約では図7のように、合計が正の値である主体“顧客”に対応して主体“請求”が存在する。その主キー値は、関連型“請求先”の関連存在従属性制約に基づいて関連が存在するような値、つまり、顧客の氏名と同じ値になる。関連“請求先”もともに存在する。

(5) 主体存在制約

E_j に関して、少なくとも主キーを含む属性のある組を $A_{jk} (k=1, \dots, m)$ とする。主体の存在条件を示すある述語 P_e があり、 $\{e_{ji} | A_{j1}(e_{ji})=v_{j1}, \dots, A_{jm}(e_{ji})$

$=v_{jm}$ かつ $P_e(v_{j1}, \dots, v_{jm})$ が真} $\subset E_j$ となる場合、 P_e に基づく主体が E_j に存在するといひ、以下の制約式で表す。

$$E_j \leftarrow P_e : A_{j1}, \dots, A_{jm} \quad (6)$$

ここで、EC は主体存在制約を表す。本制約は主体型 E_j の性質である。

図8のように、ER 図に制約を付記する。制約は広くても、一つの主体と関連を持っている主体群をスコープに持つ計算の仕様しか、定めることができない。スコープ外にある主体の属性を参照したい場合は以下の2方法のうち、自然な方を用いる。①参照される属性を持つ主体と、スコープ内にある主体との間に、制約を持つ関連を設ける。そして、参照される属性の値へ作用させた関数の値を、スコープ内にある主体の属性値とする。②参照される属性を持つ主体も関連の中に含める。制約式が多値属性を扱う場合、定義域の部分集合からなる集合を変域または値域とする関数や述語を用いる。制約式の適用法や解法に関しては、本モデルを適用する中で定める。

種々の制限のもとで、本モデルを適用したプログラム仕様記述法 PSDM, 仕様記述言語 PSDL および COBOL プログラム・ジェネレータ PSD-CG を作成した。そして、生成した COBOL プログラムの実行により、属性値従属性制約、関連存在従属性制約および主体存在従属性制約に基づいて、与えられたものとは異なる属性値、関連および主体が得られることを確認した⁵⁾。

4. 従来との比較

データベースまたはディクショナリを想定している

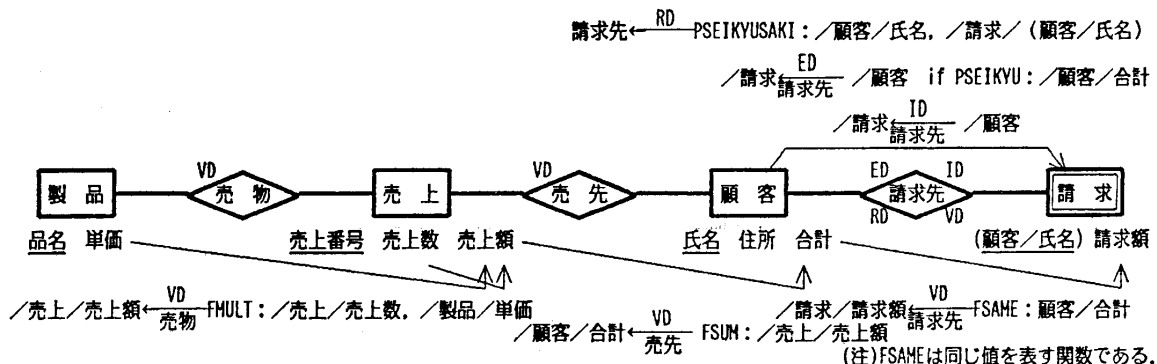
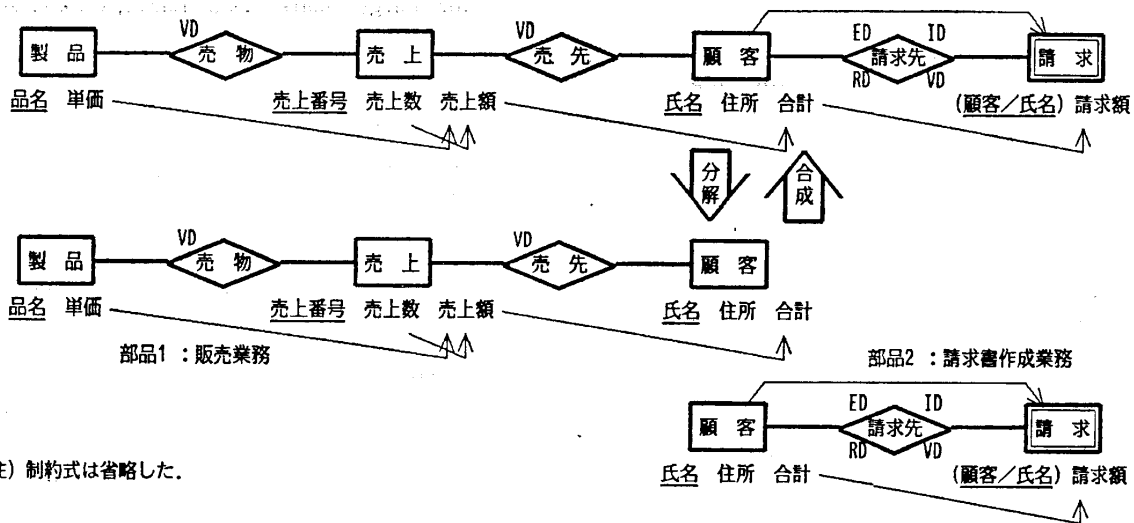


図 8 構造および制約
 Fig. 8 Structure and constraint.



(注) 制約式は省略した。

図 9 情報構造の分解・合成

Fig. 9 Decomposition and composition of information structure.

従来の EAR モデル^{1), 3), 4)}と、プログラムを想定している計算指向 EAR モデルとの主な相違点を述べる。

①従来の EAR モデルは構造、制約および操作からなる。操作によるデータベースまたはディクショナリの更新に伴い、主体や関連は発生、変化、消滅する。したがって、対象世界の中で時間とともに発生、変化、消滅する一つの事物を、一つの主体で動的に表すことができる。一方、計算指向 EAR モデルは構造と制約からなり、操作はない。主体や関連、および、その集合である主体型や関連型は変化せず、対象世界の事物を静的に表している。したがって、時間とともに変化する一つの事物は、おのおのの変化時点の事物に対応したいくつかの主体で表す。②従来の EAR モデルの制約は、データベースやディクショナリの格納データが表している主体や関連が満たす条件、または、操作を用いた主体や関連の更新が満たす条件を、おもに定めている。一方、計算指向 EAR モデルの制約は、プログラムの入力データが表している主体や関連から、出力データが表す主体や関連をえるための計算を、おもに定めている。③図 9 のような情報構造の分解・合成を手段とするプログラム部品化技術へ、計算指向 EAR モデルを適用の予定である。このためには局所性のよい情報構造が必要である。個々の制約式を一つの関連型または主体型に閉じた性質として定義し、制約を構造へ一体化した。

関係モデル⁶⁾に基づいたデータベース言語 QBE⁷⁾は、リレーション群上の計算の仕様を表している。リレーションは主体または関連に相当する情報を表している⁸⁾が、リレーションには主体型と関連型に相当する

区別はない。また、QBE のイグザンプル・エレメントを用いたリレーション相互の結合も、関連に相当する情報を表すことができる。このことから、QBE は情報構造を間接的に表していると考えられる。一方、計算指向 EAR モデルは情報構造を直接表している。

5. おわりに

対象世界の情報構造を直接表すこと、および、プログラム仕様中の計算の仕様を表すことを特徴とした計算指向 EAR モデルを報告した。従来の EAR モデルとのおもな相違点は、あらたに定式化した属性値従属性制約、関連存在従属性制約および主体存在従属性制約を用いて、計算の仕様を表すことである。QBE とのおもな相違点は、情報構造を直接表すことである。種々の制限のもとで、計算指向 EAR モデルを適用したプログラム仕様記述法 PSDM、仕様記述言語 PSDL およびプログラム・ジェネレータ PSD-CG を作成し、本モデルが計算の仕様を表すのに有効であることを確認した。PSDM に関しては、別途報告する。

計算指向 EAR モデルに関しては、今後、以下を研究する。①制約式に現れる関数や述語の定義方法、②表現能力の検証、③PSDL から、効率のよいプログラムを生成するのに有効な制約、④情報構造の分解・合成に必要な制約。

謝辞 本研究をご指導、ご支援いただいた伊吹公夫 特別研究室長、高村真司前データ処理研究部長、橋本昭洋知能処理研究部長、花田収悦ソフトウェア生産技術研究所長、徳山五郎図書館情報大学教授に厚くお礼申し上げる。

参考文献

- 1) Chen, P. P.: The Entity-Relationship Model—Toward a Unified View of Data, *ACM Trans. Database Syst.*, Vol. 1, No. 1, pp. 9-36 (1976).
- 2) Van Griethuysen, J. J. et al. (ed.): Appendix D. The Entity-Attribute-Relationship Approaches, Concepts and Terminology for the Conceptual Schema and the Information Base, ISO TC97/SC5-N695 (1982).
- 3) 酒井博敏, 堀内 一: データベース論理設計におけるトランザクションモデリングについて, 情報処理学会データベース・シンポジウム論文集, pp. 101-117 (1983).
- 4) Nakano, R.: *Integrity Checking in A Logic-Oriented ER Model, Entity-Relationship Approach to Software Engineering*, pp. 551-564, North-Holland, Amsterdam (1983).
- 5) 橋本正明: EARモデルに基づく情報構造記述を用いたプログラム仕様記述法 PSDM, 情報処理学会ソフトウェア工学研究会資料, 40-11(1985).
- 6) Codd, E. F.: A Relational Model of Data for Large Shared Data Banks, *Comm. ACM*, Vol. 13, No. 6, pp. 377-387 (1970).
- 7) Zloof, M. M.: Query by Example: A Data Base Language, *IBM Syst. J.*, No. 4, pp. 324-343 (1977).
- 8) Schmid, H. A. and Swenson, J. R.: On the Semantics of the Relational Data Model, *Proc. ACM SIGMOD*, pp. 211-223 (1975).

(昭和60年5月7日受付)

(昭和60年11月21日採録)



橋本 正明 (正会員)

昭和43年九州大学工学部電子工学科卒業。昭和45年同大学院修士課程修了。同年日本電信電話公社電気通信研究所入社。現在 NTT 情報通信処理研究所勤務。これまでおもにオペレーティング・システム, 中間言語マシン, データベース設計支援ツール, プログラム仕様記述法の研究実用化に従事。電子通信学会会員。