

ERモデル+制約を用いた対象世界の記述による 高次部品化について

7S-2

岡本克己, 橋本正明, 門田充弘
ATR通信システム研究所

1. はじめに

ソフトウェア自動作成の問題は, 1)ソフトウェア化したい対象世界の認識の方法と, 2)その認識結果に基づく計算機の動かし方, に大きく分けられる. そこで筆者らは, 1)については認識結果を記述するためのプログラム仕様記述法PSDM(Program Specification Description Method)¹⁾と, プログラム仕様を再利用するための高次部品化を研究し, 2)についてはプログラム構造設計の自動化を含めたプログラム生成法を研究中である.

PSDMは, プログラムの入出力データに表わされている情報についての枠組みと, 入出力データの形式に着目した宣言的な仕様記述法である. この情報の枠組みを記述するのにER(Entity Relationship)モデルと制約を用いているが, プログラム生成を容易にするため, 現在は制約を計算式で記述している. しかし, 対象世界の記述を容易にするには, 制約指向プログラミングと同じように, 条件式による制約の記述も許容するのが望ましい.

従来, 制約指向プログラミングは対象指向言語や論理型言語の上で研究されているが, ERモデル上の例は見られない. そこで筆者らはデータベースの観点から見た制約も参考にして, ERモデルと制約を用いた対象世界の記述による高次部品化について考察したので報告する.

2. 制約指向プログラミング

制約指向プログラミングは”制約”をプログラムの基本要素とするプログラミングであり, 制約は条件式で宣言的に記述される. このため, プログラムのコントロール・フローを考慮する必要がないので, プログラムの部品化に適している.

条件式で記述された制約を解消していくには以下の2つの方法が上げられる.

(1)局所的伝播(Local Propagation)²⁾

条件式を満たす計算式が以下の例に示すように準備されており, あるデータの値が決まると, そのデータを用いて実行可能となった計算式が実行される. このように計算がデータドリブンで局所的に進む.

(条件式) $a+b=c$

(計算式) $c \leftarrow a+b, a \leftarrow c-b, b \leftarrow c-a$

(2)制約伝播(Constraint Propagation)

制約として与えられた複数の条件式が簡約されて, 制約が解消される. このため, 以下の例に示すように, 制約を解消するのに十分な条件式が与えられなくても, 簡約された式が得られる.

(制約) $a=c+d, b=2*c+4*d, b=14$

(簡約式) $c=2*a-7, d=-a+7$

3. ERモデル+制約

ERモデルと制約を説明し, ERモデルと制約を用いた対象世界の記述について特徴を述べる.

3.1 ERモデル

ソフトウェア化したい対象世界に存在している”もの”や”こと”をエンティティといい, その”もの”や”こと”相互のつながりをリレーションシップという. また, エンティティがその種類に応じて集まった集合をエンティティ・タイプといい, リレーションシップがその種類に応じて集まった集合をリレーションシップ・タイプという.

個々のエンティティの性質は属性の値で表す. 同じエンティティ・タイプに属するエンティティは同じ属性の値を持つ. エンティティ・タイプの中で個々のエンティティを識別するための属性をプライマリ・キー属性という.

上記のうち, 仕様にはエンティティ・タイプと属性, プライマリ・キー属性, リレーションシップ・タイプを記述される.

3.2 制約

前節で述べたように, ERモデルにおいて対象世界はエンティティとその属性値, リレーションシップで認識されるので, PSDMでは, それぞれに応じた以下の3種の制約を用いている.

(1)エンティティ存在従属性制約

あるエンティティの属性値に基づいて, 他に存在するエンティティのプライマリ・キー属性値が計算式で決められる. 同時にその両エンティティの間にリレーションシップも存在する.

(2)属性値従属性制約

リレーションシップで相互につながれた複数のエンティティの上で, ある属性の値が, 他の属性の値に基づいて計算式で決められる.

(3)リレーションシップ存在従属性制約

複数のエンティティの属性値に基づいて, それらのエンティティの間にリレーションシップが存在するか否かが条件式で決められる.

ここで, (3)項の制約は条件式で記述されているが, (1)と(2)項は計算式で記述されている. 高次部品化においてはその計算式の代わりに条件式の記述も許容する. 例えば, (2)項の例として図1に示すような条件式を記述する. なお, 制約は仕様を含めて記述される.

3.3 特徴

我々が対象世界を認識する時, ”もの”や”こと”, 及び, ”もの”や”こと”相互のつながりがベースにな

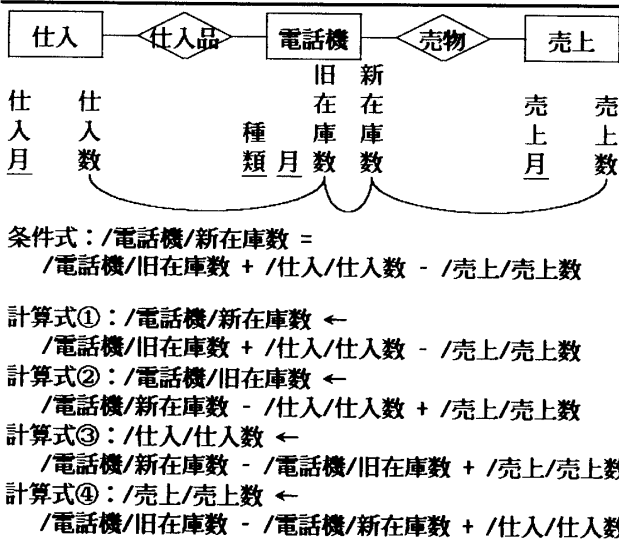


図1 属性値の条件

っている。このため、認識結果を記述する枠組みとして、エンティティ・タイプとリレーションシップ・タイプの区別は重要である。一方、この区別は、制約指向プログラミングが研究されてきた対象指向言語や論理型言語では定式化されていない。

4. 高次部品化

高次部品化の構想とその特徴、今後の研究課題について述べる。

4.1 構想

高次部品化は以下に示す方法で実現される。

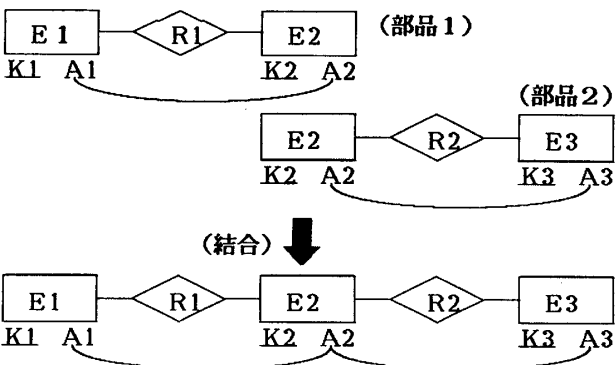
(1) 対象世界の記述と部品の蓄積

ある対象世界、例えば電話の料金計算等の対象世界を、前章に述べた方法で認識のうえ記述し、その記述されたものを部品として蓄積する。部品は図2の上段に示すようにリレーションシップ・タイプを単位として蓄積できる。その中にはリレーションシップ・タイプが対応づけているエンティティ・タイプとその属性、制約も含まれる。

(2) 部品の結合

部品は、同じ名前を持つエンティティ・タイプ、例えば図2に示すE2を重ね合わせることによって相互に結合できる。その時に不足部品の追加や、部品の修正も行ってよい。

(3) プログラム仕様の抽出



結合された部品群のうち、プログラムの入力データから値が与えられる属性と、出力データへ値が渡される属性を決めると、属性値のデータ・フローの方向が決まる。そこで、例えば図1に示した計算式の中から方向に合ったものを選んで条件式と置き換える。この制約解消は2章で述べた局所的伝播に当る。なお、入出力データと関係しない属性や制約はプログラム仕様に含めない。

(4) プログラムの生成

抽出されたプログラム仕様からプログラムを生成する。生成については別途研究中である。

4.2 特徴

高次部品の主な特徴を以下に上げる。

(1) 対象世界の知識のみに依存する部品

前節までの説明からわかるように、高次部品の中に現れているのは対象世界の知識のみであり、プログラムの構造等、プログラムの実現方法に基づいた知識は現れていない。このため、高次部品の取扱いにはプログラミングの専門家の必要性は小さい。

(2) 概念としての部品

部品の中に現れるエンティティ・タイプや属性、リレーションシップ・タイプ、制約は、我々の思考の中に現れる概念と同じものである。このため、高次部品化の研究は概念に関する研究となり、AIの研究や自然言語の研究とつながりが取り易く、将来の発展が期待される。

4.3 今後の研究課題

高次部品には様々な形態が予想されるので、例えば以下のような項目を研究する予定である。

(1) 複合部品とマクロな認識

上記4.1(1)で部品の単位について述べたが、部品の幾つも結合したものが定形的に現れれば、それを複合部品として取り扱う。この複合部品の認識の枠組み、すなわちマクロな認識の枠組みについて研究する。

(2) 属性や制約等の継承と抽象的な認識

類似部品については、その抽象的な記述を部品として蓄積し、属性や制約等を継承することによって個々の部品を得る方法もある。ここで属性はエンティティ・タイプに付随し、制約はリレーションシップ・タイプに付随するので、それぞれのタイプについて抽象化が必要である。この抽象的な認識の枠組みについて研究する。

5. おわりに

ERモデルと制約を用いて記述された高次部品は概念を表しているので、AIや自然言語処理と結び付いた発展が予想される。今後は高次部品の様々な形態を明らかにするため、概念の取扱いに関する課題を研究の予定である。

参考文献

1) 橋本正明：E A R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P S D M, 情報処理学会論文誌, Vol.27, No.7 (1986).

2) G.L.Steel JR.: The Definition and Implementation of a Computer Programming Language Based on Constraint, Ph.D. Dissertation, MIT-AI-TR 595 (1980).