

スキーマ変換システム TRICS における関係型データベース による概念構造の記述

3U-8

梶山民人 伊藤秀昭 飯田三郎 福村晃夫
中京大学大学院情報科学研究科

1 はじめに

関係型データベース [1] の概念構造は、関係で表される概念対象と、関係の属性値の参照によって表される概念対象間の関連とで構成される。既存のデータベースを再利用するには、そのデータベースがどのような概念構造を表わしているかを知る必要がある。しかし、関係が必ずしも一つの概念対象を表現しておらず、概念対象間の関連が混乱していることがある。そのため、データベースの概念構造に関する知識が失われている場合、概念構造を直感的に把握することが困難である。この問題を解決するために、我々は関係型データベースの概念構造を関係インスタンスに基づいて求めて、得られた概念構造をフレームで記述するスキーマ変換システム TRICS の開発を進めている [2, 3]。

本システムは、関係インスタンスから関数従属と包含従属の二種類のデータ従属 [1] を求めて概念構造を明らかにする。関数従属は、一つの関係が一つの概念対象を表現するように関係スキーマを再構成するため、および関係の候補キーを求めるために用いられる。一方、包含従属は、候補キーと共に概念対象間の関連を求めるために用いられる。得られた概念対象はビューで表現される。さらに、これらの関係はフレームに変換されて、概念構造を記述したフレーム型知識ベースが構成される。

上記の過程で、概念対象を表現するために定義した関係、各関係が満たす関数従属と候補キー、および関係間に成り立つ包含従属が得られる。これらはデータベーススキーマなので、本システムはデータベース管理システムによって定義されるデータ辞書と同様に、これらのスキーマ情報を関係としてデータベースに保持する。また、関係で記述されたスキーマ情報をフレーム型知識システムから参照するために、SQL 文で表された参照手続きをフレームに付加する。

データベースの概念構造に関する記述をデータベース中に保持して、その記述を参照するための付加手続きをフレームに付与することにより、関係インスタンスに基づいて概念構造を再構成するだけでなく、既に得られている概念構造の記述に基づいて概念構造を再構成することが可能になる。

本論文では、TRICS における関係による概念構造の記述方法、およびフレームに付加される参照手続きについて述べる。

2 関係による概念構造の記述

本システムは、データベースの概念構造を求める過程で得られるスキーマ情報を記述するために、以下の4つの関係をデータベースに定義する。

1. TRICS_COBJECT ... 概念対象
2. TRICS_FD ... 関数従属
3. TRICS_CK ... 候補キー
4. TRICS_ID ... 包含従属

これらの関係の利用方法を示すために、以下の4つの関係から成るデータベースを例として用いる。なお、下線は関係

Describing Conceptual Structures in Relational Databases in TRICS. Tamito KAJIYAMA, Hideaki ITO, Saburo IIDA, Teruo FUKUMURA. Graduate School of Computer and Cognitive Sciences, Chukyo University.
E-mail: kajiyama@grad.sccs.chukyo-u.ac.jp

の候補キーを表す。

学生 (学籍番号, 氏名)
教員 (教員番号, 氏名)
講義 (講義番号, 講義名, 教員番号)
TA (講義番号, 学籍番号)

2.1 TRICS_COBJECT

関係 TRICS_COBJECT は、概念対象を表現するために定義した関係の一覧を格納する。この関係のスキーマは以下の通りである。

```
TRICS_COBJECT ( relation char(18) )
```

属性 relation の値は関係名を表す文字列である¹。この関係の例を図 1(a) に示す。

2.2 TRICS_FD

関係 TRICS_FD は、各関係が満たす関数従属を記述する。この関係のスキーマは以下の通りである。

```
TRICS_FD ( id integer,
            relation char(18),
            attribute char(18),
            side integer )
```

関数従属は $R: \{A_1, \dots, A_{n-1}\} \rightarrow \{A_n\}$ と記される。ここで、 R は関係名、 A_i は R の属性である。関数従属は、関係 TRICS_FD において $(id, R, A_i, side)$ の形式の n 個のタプルで表現される。ここで、 id は関数従属の識別番号であり、 $side$ の値は属性 A_i が左辺の要素ならば 0、右辺の要素ならば 1 である。例として、以下の二つの関数従属を表現する関係を図 1(b) に示す。

学生: { 学籍番号 } \rightarrow { 氏名 }
講義: { 講義番号 } \rightarrow { 講義名 }

2.3 TRICS_CK

関係 TRICS_CK は関係の候補キーを記述する。この関係のスキーマは以下の通りである。

```
TRICS_CK ( id integer,
            relation char(18),
            attribute char(18) )
```

例えば、関係講義の候補キー { 講義番号 }, { 講義名 }, および関係 TA の候補キー { 講義番号, 学籍番号 } は、図 1(c) に示す関係で表現される。

2.4 TRICS_ID

関係 TRICS_ID は、関係間に成り立つ包含従属を表現する。この関係のスキーマは以下の通りである。

```
TRICS_ID ( id integer,
            lhs_relation char(18),
            lhs_attribute char(18),
            rhs_relation char(18),
            rhs_attribute char(18),
            type integer )
```

¹スキーマ中の定数 18 は、本システムが利用する関係型データベース管理システム Informix での識別子(関係名, 属性名など)の最大長を表す [4]。

(a) TRICS_OBJECT

relation
学生
教員
講義
TA

(b) TRICS_FD

id	relation	attribute	side
0	学生	学籍番号	0
0	学生	氏名	1
1	講義	講義番号	0
1	講義	講義名	1

(c) TRICS_CK

id	relation	attribute
0	講義	講義番号
1	講義	講義名
2	TA	講義番号
2	TA	学籍番号

(d) TRICS_ID

id	lhs_relation	lhs_attribute	rhs_relation	rhs_attribute	type
0	教員	教員番号	講義	教員番号	0
1	学生	学籍番号	TA	学籍番号	0
2	講義	講義番号	TA	講義番号	1

図 1: 関係による概念構造の記述例

包含従属は $R[A_1, \dots, A_n] \gg S[B_1, \dots, B_n]$ と記される。ここで、 A_i, B_i は関係 R, S の属性である。包含従属は、関係 TRICS_ID において $(id, R, A_i, S, B_i, type)$ の形式の n 個のタプルで表される。ここで、 $type$ は $S[B_1, \dots, B_n]$ が $R[A_1, \dots, A_n]$ の部分集合ならば 0, 等しい集合ならば 1 である。例として、以下の 3 つの包含従属を表現する関係を図 1(d) に示す。

教員[教員番号] \gg 講義[教員番号]
 学生[学籍番号] \gg TA[学籍番号]
 講義[講義番号] = TA[講義番号]

3 フレームへの記述参照手続きの付加

3.1 個別概念対象に対する付加手続き

前節で述べた概念構造を記述する 4 つの関係のうち、関係 TRICS_FD (関数従属) および TRICS_CK (候補キー) を参照する手続きは、個々の概念対象を記述するフレームに付加される。以下、フレームに付加されるスロット名と記入される SQL 文を示す。

fd-number-retrieval-sql スロット このスロットは、関係 R が満たす関数従属の識別番号の集合を求める付加手続きである。記入される SQL 文は以下の通りである。

```
select distinct id from TRICS_FD
where relation = R
```

fd-lhs-retrieval-sql スロット, **fd-rhs-retrieval-sql** スロット これらのスロットは、識別番号 n の関数従属の左辺および右辺を求める付加手続きである。以下に示す SQL 文が記入される。

```
select attribute from TRICS_FD
where side = 0 and id = $ID (左辺)
select attribute from TRICS_FD
where side = 1 and id = $ID (右辺)
```

ここで、 $\$ID$ は識別番号 n に置換される埋め込み SQL の変数である [4]。

ck-number-retrieval-sql スロット 関係 R の候補キーの識別番号の集合を求める付加手続きであり、以下に示す SQL 文が記入される。

```
select distinct id from TRICS_CK
where relation = R
```

ck-retrieval-sql スロット 識別番号 n の候補キーを求める付加手続きであり、以下に示す SQL 文が記入される。

```
select attribute from TRICS_CK
where id = $ID
```

3.2 概念構造全体に対する付加手続き

関係 TRICS_OBJECT (概念対象) および TRICS_ID (包含従属) は、個々の概念対象だけではなく概念構造全体に関するスキーマ情報を記述する関係である。そのため、これらの関係を参照するための手続きは、フレームの親フレームである root フレームに記入される。

object-retrieval-sql スロット 概念対象を表現するために定義した関係の一覧を求める付加手続きであり、以下の SQL 文が記入される。

```
select relation from TRICS_OBJECT
```

id-number-retrieval-sql スロット 包含従属の識別番号の集合を求める付加手続きであり、以下に示す SQL 文が記入される。

```
select distinct id from TRICS_ID
```

id-retrieval-sql スロット 識別番号 n の包含従属を求める付加手続きであり、以下に示す SQL 文が記入される。

```
select lhs_relation, lhs_attribute,
       rhs_relation, rhs_attribute, type
from TRICS_ID
where id = $ID
```

4 おわりに

本システムを利用することにより、概念対象と関連が整理されて、データベースの概念構造が再構成される。その過程で得られたスキーマ情報は関係によって記述されるので、データベースの自己記述性が高まる。また、本システムによって作成されるフレーム型知識ベースはデータベースへのオブジェクト指向のインターフェイスとして有用であり、データベースの概念構造を直感的に理解することが容易になる。

参考文献

- [1] C. J. Date. *An introduction to database systems*. Addison-Wesley, 6th edition, 1995.
- [2] 梶山民人, 伊藤秀昭, 飯田三郎, 福村晃夫. スキーマ変換システム TRICS における関係型データベースの概念構造構築について. 情報処理学会研究報告, Vol. 97, No. 81, pp. 13-18, August 1997.
- [3] Tamito Kajiyama, Hideaki Ito, Saburo Iida, and Teruo Fukumura. Eliciting concepts and their relationships from relational instances. In *Proceedings of the Fourth World Congress on Expert Systems*, pp. 590-597, March 1998.
- [4] アスキー. *informix-SQL User's Manual*, 第 3 版, 1988.