6R−7

異データベース間におけるデータマッピング手法の提案(3)
—データベース内における制約条件のマッピングへの適用方法 *—

坂田哲夫† 石黒正典‡ 大沼守一§

NTT 情報通信網研究所¶

1 はじめに

1.1 背景

近年、一企業内で個別に開発された複数のデータベース(以下DBと略記)上のデータを統一的に取り扱いたいという要求が高まってきた。この要求を満たすため、DBの定義情報であるスキーマを仮想的に統合する方法[1]や、それを統合して制御(アクセス、一貫性の保証)する方法[2]が盛んに研究されている。

しかし、これらの方法の適用には複数のDBやスキーマ間の関係が既知であることが前提になっており、これらの関係の分析手法は未解決である[3]。

1.2 目的

筆者らは複数DB間におけるスキーマ間の対応関係を分析する一手法として、DBの内部スキーマのデータ項目(ここでいうデータ項目とは意味ある情報としての最小の単位であり、関係DBではカラムに相当する)単位でのマッピング手法を提案している[4]。

ところが、現実世界の実体をDB上で表現するためには、複数のデータ項目が用いられることが多く、しかもDBが異なれば同一実体に対する表現方法も異なりうるので、データ項目単位でのマッピングだけでは不十分である。

そこで本稿では、関係DB—上述のデータ項目は関係DBではカラムに相当する—に関して、カラム間でデータマッピングがなされていることを前提に、同一実体を表現しているカラムの集合(以後データセットと呼ぶ)同志の以下のようなデータマッピング手法を提案する。

1. カラム間の依存関係をグラフとして表現する(DBグラフ)。
2. カラム間のデータマッピングの結果[4]とDBグラフを用い、実体を表すためのグラフ(マッチンググラフ)を生成する。

2 DBグラフの記述法

2.1 DB内における制約

データセットに含まれるカラムは、以下のような制約によって関係付けられている。

あるカラムの値が決まれば、他のカラムの値もそれに関連して決まる

これらの2つのカラムは、同一の表に属する場合と、別の表に属する場合に分けられる。我々はこの種の制約の中で、もっとも一般性のあるものとして、前者の代表として関数従属性[5]を、後者の代表として参照一貫性[6]を選んだ。

2.2 DBスキーマのグラフ表現

制約を表現するため、DB内のカラムを節点とし、制約を弧とする有向グラフを作る。これをDBグラフと呼び、以下の手順でスキーマから生成する。なお、関数従属性を→で表す。

1. 節点の生成
   - 参照一貫性で関連づけられるカラムには1つの節点を生成する。
   - それ以外のカラムには1カラム毎に1つの節点を生成する。
2. 節点の追加

   →の左辺に、複数のカラムが現われているなら、このカラムの集合に対応する節点を一つ生成して追加する。
3. 弧の生成

   全ての関数従属性に関して、以下のように定まる始点と終点に対し、弧を生成する。
   - →の左辺が単一のカラムなら、これに対応する節点を始点とする。
   - →の右辺が単一のカラムの時、これに対応する節点を終点とする。
   - →の左辺が複数のカラムならば、先に述べた手順で追加した節点を始点とする。
   - →の右辺に複数のカラムが存在するときには、個々のカラムに対応する複数の節点を、終点とする。この時は、1つの関数従属性によって、対応する始点との間に複数の弧を生成する。
4. 弧の追加

   以下の自明な従属性に対応する弧を追加する。
   - 2つの節点に対応するカラムの集合が包含関係にあるなら、包含するほうを始点、されるほうを終点とする弧を生成する。

3 DBの比較

3.1 マッチンググラフの生成

カラム間の対応関係をもとに、DBグラフを用いて、実体を表現するデータセットの間の対応関係を表す有向グラフ、マッチンググラフを生成する。

---

*An Application using some Constraints in Databases to Data Mapping Method among Heterogeneous Databases.

†Testuo SAKATA
‡Masanoni ISHIGURO
§Shuichi OHNUMA
¶Network Information Systems Labs., NTT

いま、比較しようとするDBをDB_1, DB_2、DB_iのカラムの集合をC_{ik}、DB_iに対応するDBグラフをG_i、生成されるマッチンググラフを$M = (N, F)$とする。また、$p \Rightarrow q$はグラフ上でpからqに至る路(path)が存在することを表す。

3.2 マッチンググラフの生成手順

1. Mの節点Nの生成

   対応関係にあるカラムの集合の対(C_{1k}, C_{2k})を、マッチンググラフの節点$n(\in N)$とする。この時C_{ik}に対応する節点がG_iになければ対応する節点と弧を、2.2の手順に従って追加する。

2. Mの弧Fの生成

   Nから得られる全ての対(n_1, n_2)に関して、以下の操作を施す。

   - $n_1 = (C_{11}, C_{21})$, $n_2 = (C_{12}, C_{22})$とするとき、$C_{11} \Rightarrow C_{12}$かつ、$C_{21} \Rightarrow C_{22}$であるならば、$n_1$を始点、$n_2$を終点とする弧$(n_1, n_2)$を生成する。

3.3 データセットの候補の抽出

まず、マッチンググラフから冗長な弧を削除する。冗長な弧とは、$p \Rightarrow q$である全ての2節点について、削除してもこの関係が変化しない弧である。

次に、実体を表現しているデータセットの候補は、冗長な弧のないマッチンググラフM'のサブグラフ$\bar{\mu}$から以下のように得られる。

1. マッチンググラフM'のいかなる弧の終点にもならない節点をμ、μとの間に路が存在するM'のサブグラフを$\bar{\mu}$とし、
2. $\bar{\mu}$に含まれる節点$n = (C_{1k}, C_{2k})$を導きだした、DBグラフ中の節点に対応するカラムの集合を、実体の属性に結びつけ、
3. 弧(n_1, n_2)を導きだしたDBグラフ中の弧を、実体の属性を関連づける制約に結び付ける。

このとき、μに対応するG_i中の節点の対応するDB_iのカラムの集合C_{ij}の値が定まると、弧(n_1, n_2)によって関連づけられるカラムの値も一意に定まるため、$\bar{\mu}$によって定まるカラム群は、μに対応するカラムを識別子とする実体のデータセットと見なすのが妥当である。

4 名称と構造の不整合の解消

4.1 名称と構造の不整合

マッチンググラフによるデータセットの候補の抽出に際しては、対応関係にあるデータセットに関しては、名称の分析で「同等の内容を表現している」と判定されたカラム同志は、それぞれのDBグラフの中で「同等の位置を占めている」ことが前提であった。

この前提が満たされていない場合、名称と構造の不整合という。まず、名称と構造の整合性を次のように定義する。記号は3.1と同様である。

> $(C_{11} \Rightarrow C_{12})$と$(C_{21} \Rightarrow C_{22})$が共に成り立つか、双方とも成り立たないとき、名称と構造は整合しているという。

4.2 不整合の解消法

上の定義より、$C_{11} \Rightarrow C_{12}$か$C_{21} \Rightarrow C_{22}$のうち成立しない側のDBグラフにおいて、これが成立するような制約がDB中に存在すれば、不整合が解消されマッチングの結果はより信頼性を増す。

そこで、上記の条件を満たす制約の候補を挙げて、検証する必要があるが、1つのインスタンスを検査するだけでは不十分で、全てのあるべきインスタンスを検査して初めて検証できる。従って、これらは実際のDBのデータでは検証できず、設計情報の見直しなどによって可能となる。

4.3 不整合が解消できない場合

上記4.2において不整合を解消する制約の候補が発見できないか、検証に全て失敗する事も有り得る。これは、「名称の類似性によるカラム間の関係」を仮説と見なしたとき、これを反証したことを意味する。即ち、名称によるマッピングが誤っているか、2つのDBが共通の実体を表現するデータセットを有するという根本の仮定が誤っているかのいずれかである。

5 まとめ

我々は、複数のDBを統合する際における、スキーマ同志の対応関係の分析を支援する方法を提案した。ここでは、関数従属性と参照一貫性によってスキーマの構造を表現して、データセットをマッチングする。これによって、当初問題になっていたスキーマ間の構造上の違いを吸収することができる。

今後の課題として、不整合を解消する可能性のある制約を機械的に発見する手法の確立が挙げられる。また、2つのDBで同一の情報を表現しているにも係わらず、それらの違いがスキーマに現れないものについても検討する。(了)

参考文献

[1] C.Batini and M.Lenzerini. *A Comparative Analysis of Methodologies for Database Schema Integration.* ACM Computing Surveys, Vol. 18, No.4, Dec., 1986.

[2] L.Witold, L.Mark and N.Roussopoulos. *Interoperability of Multiple Autonomous Databases.* ACM Computing Surveys, Vol.22, No.3, 1990.

[3] 石黒 他. 異データベース間におけるデータマッピング手法の提案(1). 情報処理学会第45回全国大会論文, 1992.

[4] 大沼 他. 異データベース間におけるデータマッピング手法の提案(2). 情報処理学会第45回全国大会論文, 1992.

[5] J.D.Ullman. *Principles of DATABASE SYSTEMS (2nd Ed.).* Computer Science Press, 1982.(データベースシステムの原理. 日本コンピュータ協会)

[6] C.J.Date. *A Guide to THE SQL STANDARD* Addison-Wesley, 1987.(標準SQL. トッパン)