5S-3

データベース統合のための作業手順作成支援システムの開発

三森 祐一郎 † 森嶋 厚行 ‡

筑波大学 図書館情報専門学群 † 筑波大学大学院 図書館情報メディア研究科 ‡

1. はじめに

本稿で扱う**データベース統合**とは，複数の異なるデータベースのコンテンツをまとめて一つのデータベースに移行する作業の事である．本稿では特に，実務で広く扱われているリレーショナルデータベースを対象とする．このようなデータベース統合は，さまざまな場面で必要とされる．例えば，合併などにともなう組織のデータベースの統合や，データウェアハウス構築の際のデータ移行などの際に必要となる．

データベース統合における問題は，その移行作業が一般に困難であることである．第一の理由は，データベースに格納されているデータには，欠落，間違い，例外，などが存在し，統合前にはそれらの処理 (データクリーニング) を行う必要があることである．第二の理由は，データベース統合はしばしば人手を要する作業であるため，作業コストを下げるための最適化が必要となることである．これらを考慮すると，その必要が無い理想的な状況とくらべて，移行作業手順が急激に複雑になり，データ移行の正当性を保証したままその作業手順を手作業で作成することが困難になる．

本稿では，このデータベース統合のための作業手順の作成を支援するためのシステムを提案し，その開発について述べる．システムの利用者は，作業手順の作成者である．提案するシステムの動作は次の通りである．まず，利用者は，例外的な作業や細かい属性の違い，作業コスト等を無視して，抽象度の高い (一般に単純な) 作業手順をシステムに入力する．つぎに，利用者と対話しながら，最初に入力された単純な作業手順を，順次複雑な作業手順に変換していく．システムには正しいことが分かっている手順変換規則が組み込まれており，変換の際にはこの規則を利用する．これにより，データ移行の正しさについて作業手順の作成者が悩むことなく，複雑な作業手順を作成できることを目標としている．

関連研究. 既存の各種 ETL ツール[1] では，データベース間のデータ移行のプロセスを管理するためのツールを備えている．Simitsis らの論文[2] では，ETL のためのワークフローを最適化する研究が行われている．本研究がこれらと異なる点は，(1) 抽象的な作業の具体化を支援するシステムであること (2) データベース統合の作業手順作成に人手による作業が不可欠と考え，それらもフレームワーク内に組み込んでること，である．

本稿の構成は次の通りである．まず，2 章で本稿で扱うデータベース統合の作成手順について説明する．3 章で，提案システムの概要，および，提案システムで利用する書き換え規則の例について説明する．4 章で，実装について説明する．5 章はまとめである．

2. データベース統合の作業手順例

本章では，データベース統合の作業手順を，リレーショナル代数を用いて表記する．(3 章ではリレーショナル代数を拡張する).

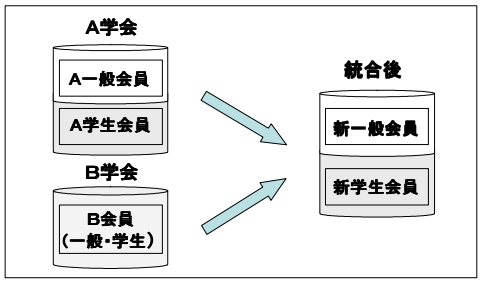

図 1 データベース統合の例

図 1 は学会 A と学会 B という二つの学会データベース (以下 DB) の会員データの統合例である．A 学会は会員を学生リレーションと一般リレーションに分割して保存しており，B 学会ではリレーショナル属性「会員種別」により区別して一つのリレーションで管理している．これを抽象度の高い作業手順では以下のよう表すことができる．

$$\text{新一般会員} \leftarrow \text{A 一般会員} \cup \sigma_{\text{種別}=\text{一般}}(\text{B 会員})$$
$$\text{新学生会員} \leftarrow \text{A 学生会員} \cup \sigma_{\text{種別}=\text{学生}}(\text{B 会員})$$

このように，本質的にはオペレータの適用数が 4 つの単純なデータベース統合であるが，実は，実際に必要な作業手順は大幅に複雑なものになってしまう．このようになってしまう理由は，下記の通りである．(1) リレーションによって属性が異なり和両立にするため，データの変換を行う必要がある．(2) 属性値のフォーマットが間違って入力いたり，値の更新が必要なものがあり，これらのクリーニングの手順が必要となる．(3) 人手による作業を最小化する必要がある．

3. 提案システムの概要

本システムは以下の流れでの作成を支援する (図 2).
(1) 抽象度の高い作業手順式を入力として受け取る．(2) ユーザとの対話を通して作業手順を展開する．(3) 展開した結果を出力する．

Development of a Support System for Constructing Database Integration Workflows
Yuichiro Mitsumori† Atsuyuki Morishima‡
School of Library and Information Science, Univ. of Tsukuba.†
Graduate School of Library Information and Media Studies, Univ. of Tsukuba.‡

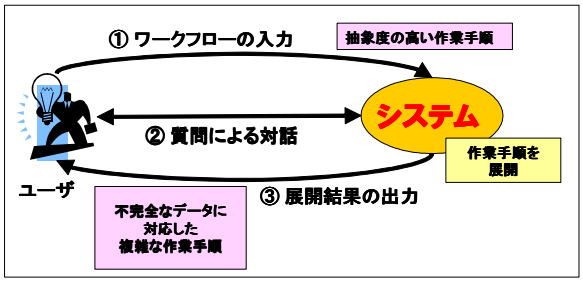

図 2 システムの概要

3.1 追加オペレータ

データベース統合のための作業手順を記述するために，ここでは一般的なリレーショナル代数に含まれていないオペレータを定義する．これらは必ずしもプリミティブではないが，これらのオペレータを導入することによって，複雑なデータベース統合作業手順を形式的な式で表現できるようになり，式の自動検証や最適化のベースとして利用可能になる．

- i-join

$$R' \leftarrow S\ i\text{-}join_e\ R$$

リレーション S と R を，属性値だけからは明示的に計算できない結合条件 e によって結合する．これは，結合のために背景知識が必要な名寄せなどで利用される．

- expand

$$R' \leftarrow expand_{a_1,\ldots a_m}(R)$$

リレーション R に，既存の属性値だけからは明示的には求められない新たな属性 $a_1 \ldots a_m$ を追加する．

- clean

$$R' \leftarrow clean_a(R)$$

リレーション R の属性 a のすべて値を，異常値や間違いのない状態に修正したリレーション R' を返す．

- adapt

$$R' \leftarrow adapt_{a'_1=e_1,\ldots a'_m=e_m}(R)$$

リレーション R の各タプルを，変換式 $a'_1 = e_1 \ldots$ によって変換した新たなリレーション R' を計算する．

3.2 作業手順書き換え規則とその利用

前述したように，複雑な作業手順の作成時に問題となるのが，作成した作業手順が本当に意図した結果をもたらすかどうかを確認することである．本システムでは，個別に検証された**書き換え規則**を定義し，それを利用した書き換えをすることで，出力される作業手順が意図した結果をもたらすことを保証する．書き換え時には，ユーザが比較的容易に検証可能な問をおこない，その返答にしたがって，書き換えを実行していく．図 3 は書き換え規則の例である．この例では，意味的には和集合をとりたい二つのリレーション R と S が存在するときに，単なる和集合演算を実行すべきか，それともより複雑な手順を踏むべきかを判断し，書き換えを行う規則である．

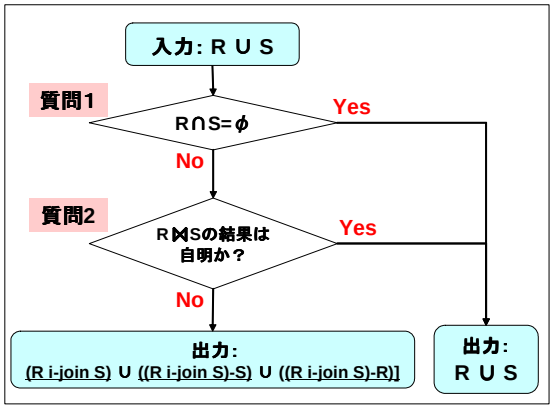

図 3 書き換え規則の例

4. プロトタイプシステムの実装

プロトタイプシステムの実装を行った．データベース統合作業手順書の書き換えモジュールの実装には，swi-prolog[3] を利用し，ユーザインターフェースの実装には Java を用いた．書き換えモジュールは，ユーザインターフェースからリストの形式で受け取った作業手順データを構文チェック用の述語に渡す．構文が正しい場合はそのまま書き換え用の述語に渡される．渡されたリストの入れ子の中身は一つずつチェックされ，変換規則にマッチする演算子が出現するたびに必要な質問が利用者に行われる．その結果，書き換えが必要な場合には実行される．最終的にすべて処理が終わったところで結果がユーザインターフェースに返される．

5. おわりに

本稿では，データベース統合の作業手順の作成を支援するシステムの開発について述べた．本システムでは抽象度の高い作業手順を最初に受け取り，作業手順の作成者である利用者との対話を通して，例外データの処理や変換作業時のコストの低減に配慮した複雑な作業手順を作成し，出力する．この書き換えは，正しいことがわかっている変換規則に従って行われるため，作業手順の正しさを検証する作業が容易になると期待される．今後は，オペレータや書き換え規則のさらなる検討，および他のデータ統合ツールとの連携などを行う予定である．

参 考 文 献

1) cloverETL, http://cloveretl.berlios.de/.
2) Alkis Simitsis, Panos Vassiliadis, Timos Sellis : *State-Space Optimization of ETL Workflows* IEEE Transactions on Knowledge and Data Engineering (TKDE), Volume 17, Issue 10, pp. 1404-1419, October, 2005.
3) SWI-prolog, http://www.swi-prolog.org/.