

# 大規模データ連携処理の実現に向けた実行時設計情報の管理方式

新井 嘉章<sup>†</sup> 楓 仁志<sup>†</sup> 石井 篤<sup>†</sup> 高山 茂伸<sup>†</sup>

三菱電機株式会社 情報技術総合研究所<sup>†</sup>

## 1. はじめに

システム間でのデータ連携を実現する場合、連携アプリを独自に開発するか、GUIで作成した設定情報に従ってデータ処理を実現する ETL 製品を利用する事が一般的である。

しかし、これらのシステム間データ連携構築手法では、連携対象となるシステムの接続情報や、システム毎の環境差に対応する為のデータ変換処理を個別に実装していく為、新たな連携先システムの追加により、連携の規模が拡大していくと、システムの変更や管理が複雑化するだけでなく、連携処理の個別実行によるデータ処理性能の劣化が問題となってくる。その為、昨今需要が高まっている散在する複数のデータベースからのデータ収集や、文献[1]で報告されている CDC を活用したリアルタイムデータ連携の構築を考えたとき、実現が困難となる。従来のシステム間データ連携の構成例を図 1 に示す。

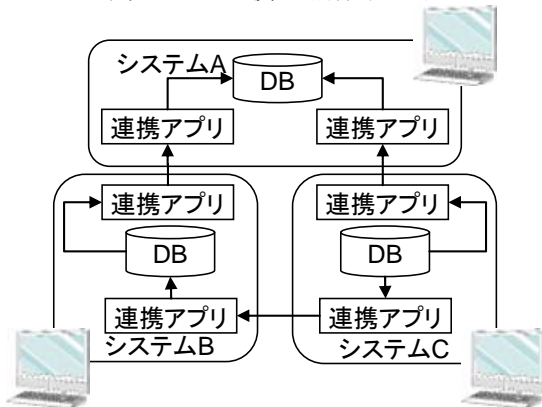


図 1. 従来のデータ連携実現方式

大規模な多対多のシステム間データ連携を実現する為には、連携システムの追加や変更の際、①接続先システムの情報を容易に変更管理できる設計情報の管理方式と、②連携元システムからのデータ取得や、データ変換処理の重複を除去し、連携先システムや連携処理の負荷を極小化する最適化が必要となる。

筆者らは、その実現方式として、複数の連携の設計情報を統合して実行する為の手法を検討した。本論文では、連携定義の設計情報管理方式について述べる。

## 2. 全体構成

これまで、システム間データ連携に参加するシステム同士は、連携アプリによって直接接続していたが、図 2 に文献[1]の報告によるハブ型構成にする事で、システムから連携処理を独立させることができ、また、データ連携の基盤化による連携の管理と実行の一元化を図ることができる。

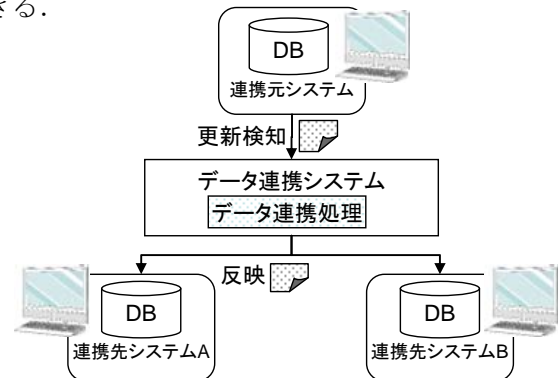


図 2. 全体構成

## 3. 設計情報の管理方式

本論文における設計情報の管理方式を図 4 に示す。本方式では、連携の定義部分である論理設計情報と、DB 接続情報等の定義部分である物理設計情報を独立して管理し、また、それらの間の対応関係を、マッピング情報として更に別に管理する事で、連携システムの追加変更時における設計情報の追加変更作業を容易にする。

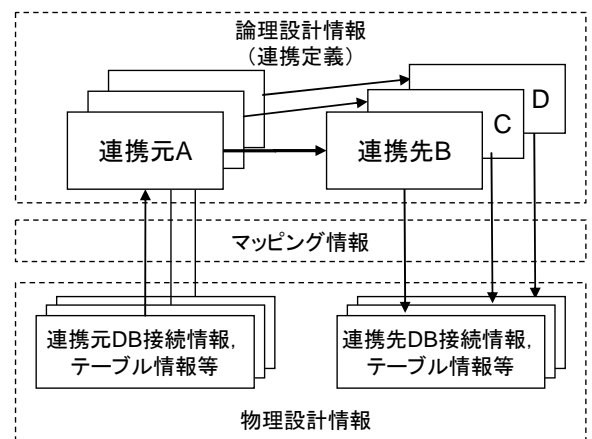


図 4. 設計情報の管理方式

A Run Time Information Management System Designed for The Realization of Large Scale Data Integration.

<sup>†</sup>Yoshiaki Arai, Satoshi Kaede, Atsushi Ishii, Shigenobu Takayama. <sup>†</sup>Information Technology R&D Center Mitsubishi Electric Corp.

4. 実行時最適化

従来のデータ連携処理は、単一業務内でのデータ転送機能として、処理を個別に実装する事が多かった為、連携元となるシステムから、連携先となる複数のシステムに連携する際、各データ連携処理間で共通的な処理がある場合も、それぞれの連携処理が独立して動作する為、処理の重複が発生していた。本論文で述べるデータ連携の実行方式では、各連携処理は、それぞれ独立して動作するのではなく、3章で示した複数ある論理設計情報を実行時に統合し、連携処理間のデータ変換処理を最適化して動作する。図5に本論文における実行方式を示す。

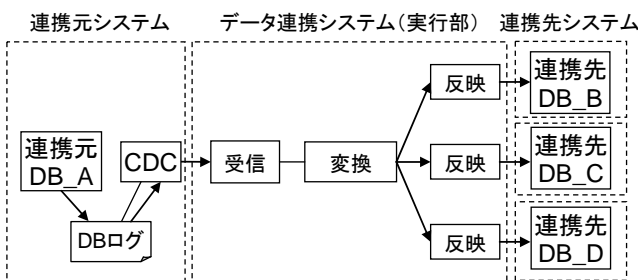


図5. 複数の連携を統合化する実行方式

連携元システムに設置したCDCが連携元DBの変更を検知してデータ連携の実行部に連携データを通知すると、データ連携システムの実行部は、受信したデータに対応した変換処理を実行し、各連携先への反映を実行する。その際、変換処理の重複が発生しないようにする為、個別に設計された設計情報を統合した統合変換情報を用いて動作する。

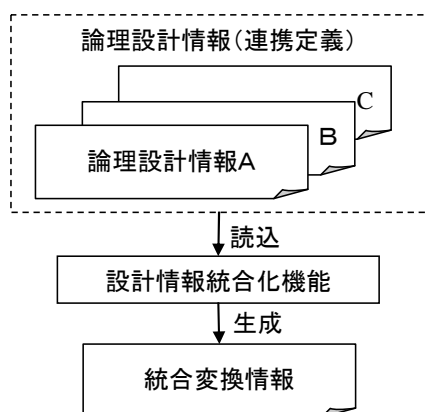


図6. 設計情報の統合化

図6に、論理設計情報から統合変換情報を生成する流れを示す。設計時に作成した論理設計情報は、設計情報統合化機能を介して、統合変換情報に変換される。

統合変換情報では、論理設計情報で記述した各連携元と連携先間の対応関係を、図7に示すような仮想テーブルを仲介した対応関係として保持する。仮想テーブルとは、連携元テーブルや連携先テーブルとは異なる、データ連携システムの実行部が使用する内部テーブルである。

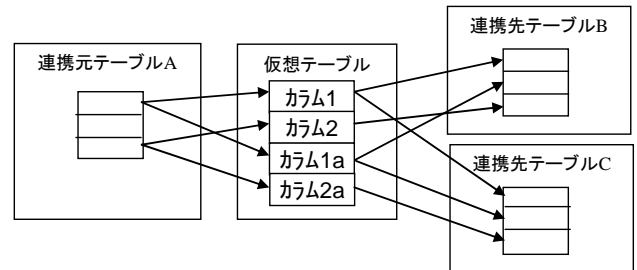


図7. 統合変換情報

連携元テーブルから受信した、ある1つのカラムのデータは、連携先毎の形式の違いに応じて変換処理が適用された後、いくつかの仮想カラムに格納される。その際、同一形式の連携先が複数ある場合には、それらを集約し、共通する1つの仮想カラムに格納する。この過程が、本論文における、実行時の重複処理を除去する最適化の仕組みである。

各仮想カラムに格納されたデータは、図5に示した反映部によって、それぞれが紐づく連携先テーブルに反映される。

5. おわりに

本論文では、大規模な多対多のデータ連携処理の実現に向けた、連携定義の設計情報管理方式について述べた。今後の課題としては、従来手法によって複雑に構成された既存のシステム間データ連携のリプレース等に向け、本手法を適用し、移行する為のスムーズな移行手順を検討し、提案することである。

参考文献

[1] Jingang Shi, Yubin Bao, Fangling Leng, Ge Yu: "Study on Log-Based Change Data Capture and Handling Mechanism in Real-Time Data Warehouse", International Conference on Computer Science and Software Engineering, pp. 478-481, (2008)  
 [2] 楓 仁志, 高山 茂伸, 菅野 幹人: "複数データソース間における効率的なデータ連携処理方式の提案", 第73回情報処理学会全国大会論文集, pp. 529-531, (2011)