

RDB のリレーションに基づく連携サービス設計方式

Service Design Method based on Relation of RDB

米田 貴雄†
Takao Yoneta

山足 光義†
Mitsuyoshi Yamatari

1. はじめに

近年、システムをサービスと呼ばれる機能の組み合わせで構築する SOA の考え方が普及し、現在ではサービスを活用した開発手法は一般的なシステム開発手法の一つとして確立されてきている。

機能をサービス化する単位はシステムの性質に合わせて決める必要があるが、最もプリミティブな単位としては、データベース操作の単位で機能をサービス化できる(以降、本稿ではこの単位でサービス化されたものを DB サービスと呼ぶ)。既存の DB サービス同士を組合せることでデータ連携が可能であるが、利用シーンによっては連携させたいデータへの操作がサービス化されていない、連携させたいデータ同士が直接関係付かないといった場合があり、サービス開発者の負荷が高くなるという課題があった。

本稿では既存の DB サービスを活用して新規に連携サービスを開発する場合に、データ間の関係に着目し RDB のリレーションに基づいてサービスを設計する方式を提案する。

2. 関連技術

2.1 サービス化技術

データベース操作を Web サービス化することにより、利用するアプリケーション側のプラットフォームや言語に依存せずにデータアクセス手段を提供することができる。データベース操作を Web サービス化する方法としてはサブレットを利用する方法があり、データ操作の実行結果を XML 形式のメッセージを SOAP や REST などの通信方式で交換する方法などが一般的である[1][2]。

2.2 サービス連携技術

公開されている既存のサービスを利用して機能やサービスを開発することで、新規に開発するよりも開発コストを抑えることができる。複数のサービス同士を連携させて複雑な処理を実現するコンポジット・サービス(以降、連携サービスと呼ぶ)を実現する方法としては大きく二通りの方法がある。一つ目はプロセスエンジンを使う方法で、BPEL (Business Process Execution Language) などのプロセスモデリング言語により記述された定義に従いサービス呼出しを実行する方法である。二つ目は、ESB (Enterprise Service Bus) を使う方法で、ルーティングされた定義に従いサービス呼び出しを実行する方法である。どちらの方法でも、複数のサービスの実行を束ねたサービスを提供することができる。本稿では、DB サービスはステートレスであり、また、大量のデータを扱う可能性があるという点から、パフォーマンスに利点のある ESB を用いることを前提としている。

3. 課題

DB サービスを活用してシステムを構築する際、以下のような場面で問題となる。

(1) 連携させたいデータ操作がサービス化されていない

例えば、データ X に対応するサービス X はあるが、データ Y に対応するサービス Y が存在しない場面である。この場合、システム開発者はデータ X、Y を利用するサービス XY を作成するか、データ Y をサービス化した上でサービス X と連携させたサービス XY' を作成する必要がある。前者については既存サービスを活用できず、後者については、サービス化に加え連携のための作業が発生することになり、サービス開発のための負荷が高い。

(2) 連携させたいデータ同士が直接関係付かない

例えば、データ X とデータ Z に対応するサービス X、サービス Z が存在する場合に、これらのデータを連携させたデータ XZ を扱いたい、データ X とデータ Z には直接関係付けられる項目が存在しないため、サービスを連携させることができないという場面である。この場合、サービス開発者は、データベース構造の仕様書を参照し、データ X、Z を連携させるための関係 Y を見つけ、データ X、Y、Z を包括的に扱うサービス XZ を作成する必要がある。このため、サービス開発者はデータベース構造を熟知している必要があり、サービス開発の障壁となっている。

4. 解決策

上記課題に対する解決策として、RDB のリレーションに基づいて連携サービスを設計する方式を提案する。提案方式では、以下の処理を組合せることにより、連携サービスの設計を支援する。

(1) テーブル間のリレーションに基づき関連テーブル群と連携パスを抽出する

連携サービスの作成において、個々のサービスで利用するテーブル間のリレーションを洗い出し、指定されたデータ間を紐付けるために必要となるテーブル群を抽出する。またテーブル間をつなぐ連携パスを抽出する。

(2) データを保持しているテーブルのスキーマ情報に基づいてサービス化する定義を生成する

連携に必要なデータへの操作がサービス化されていない場合には、そのテーブルのスキーマ情報を基に参照/更新するための SQL とサービス化のための定義を生成する。この定義に基づき実行される DB サービスは、連携サービスの実行時に利用する。

(3) 連携パスに対応させてサービスを連携させる定義を生成する

(1) で抽出したパスの順番に、使用するテーブルに対応するサービスを組合せ、各サービス間の入出力項目をマッピングした連携定義を生成する。

5. 実現方式

提案方式を適用した設計支援ツールを用い、サービス連携を実現する際のシステム構成を図1に示す

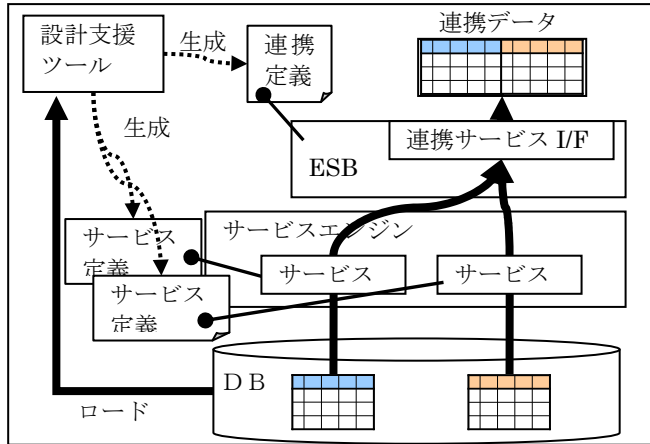


図1 システム構成図

システムの構成要素は以下の通り。

- 設計支援ツール：ユーザ指定のデータ間を連携させるサービスを設計するためのツール。サービス化に必要な定義と連携に必要な定義を出力する。
- ESB：サービス呼び出しを制御する実行基盤。連携定義に基づき連携サービスを提供する。
- 連携定義：連携サービスで呼び出すサービスのルーティングを定義した情報。
- サービスエンジン：DB サービスを生成するためのサービス生成エンジン。与えられた SQL 定義に基づき DB サービスを生成する。
- サービス定義：DB サービスで実行するデータベース操作を定義した情報。

6. 評価

提案方式の有効性について机上検証した内容について述べる。

6.1 評価シナリオ

直接関係付かないデータ同士を連携させるサービスを作成するシナリオを考える。A社では、社員の健康管理を支援するために社員食堂で喫食した献立の情報を提供するサービスを新たに提供することとした。この際、既存の社員台帳と献立予定表を連携するために喫食管理台帳を新規にデータベースに追加した。このときのテーブル構造を図2に示す。このような状況下で、サービスを開発するために提案方式を適用した設計ツールを利用することを考える。ここで、社員検索サービスと献立検索サービスは既に存在しており、活用できるものとする。

6.2 ツールを用いた連携サービスの設計

作成するサービスは、氏名と期間を入力条件として、喫食した献立のリストを取得する検索サービスとする。まず、設計者は検索結果として得たいテーブルの属性情報を指定する。ここでは、社員台帳の社員 ID と献立予定表の提供日(喫食日)、献立名、総カロリーをそれぞれ指定する。次に、検索条件に使用する属性情報を指定する。ここでは、社員台帳の社員 ID と喫食管理台帳の喫食日を指定する。

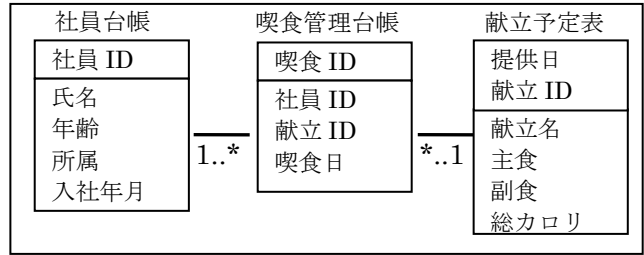


図2 データ間の関連付け

設計ツールは指定された入出力項目を持つテーブル間の関係から喫食管理台帳を見つけ、社員台帳-喫食管理台帳-献立予定表という連携パスを抽出する。このとき、社員台帳、献立予定表に対応する既存サービスとして社員検索サービスと献立検索サービスがあるが、喫食管理台帳は新規テーブルなので対応するサービスは無い。そこで、設計ツールは次のような SQL を実行するサービス定義を出力する。太字部分はユーザカスタマイズ部分であり、@で囲まれた範囲は変数を表している。

```
select 社員番号, 献立番号, 喫食日 from 喫食管理台帳
where 喫食日 between @開始日@ and @終了日@
```

最後に、連携サービスに設定した入出力項目と各サービスの入出力項目をマッピングした連携定義を出力する。

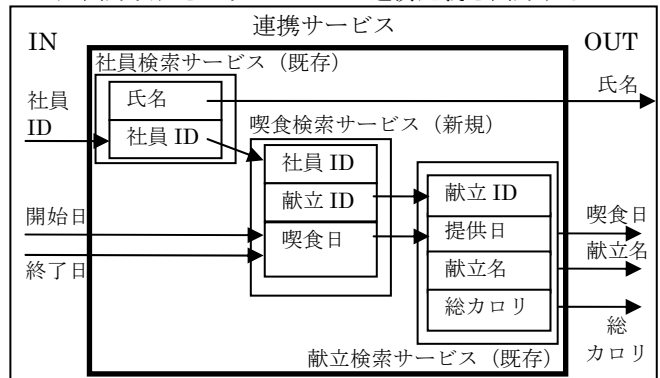


図3 設計された連携サービス

上記のように設計した定義を用いることで、データ間に直接的な関係がない場合でも、既存サービスを活用した連携サービスを実行することが可能となる。

7. おわりに

本稿では、データ間の関係に着目し連携サービスを設計する手法を提案した。提案方式を机上評価した結果、テーブル間のリレーションを利用することで、既存サービスを活用した連携サービスを設計できることを示した。今後はツールを実装し実適用における有効性を評価する。

【参考文献】

[1] IBM, 「WSAD で Web サービスを作ってみよう!! : 第6回 DB2 から Web サービスを作る」, http://www.ibm.com/developerworks/jp/webservices/library/j_ws-wsad6/index.html, 2006.
 [2] Oracle, 「Oracle Application Server Web Services 開発者ガイド」, http://download.oracle.com/docs/cd/E18355_01/web.1013/B31868-01/devdbase.htm, 2006.