APIを用いたデータ移行並びに検証方式の提案

古田 裕久† 石原 鑑† 山本 浩和‡
三菱電機（株）先端技術総合研究所† 三菱電機（株）神戸製作所‡

1. はじめに

ミドルウェアやフレームワークなどのソフトウェア技術は、システム開発において欠かせない技術となりつつある。しかし、どのようなソフトウェア技術でも、技術進歩に伴い陳腐化し、新たなソフトウェア技術に取って代わられるのが宿命である[1]。多くのシステム開発では、陳腐化したソフトウェア技術を、新たなソフトウェア技術に置き換えるといったリプレイス開発が行われている。

そのリプレイス開発を進めていくにあたっては、陳腐化したソフトウェア技術に依存したシステムやシステムに登録されたデータを、新たなソフトウェア技術に対応させる設計と開発が必要となる。その設計と開発を容易に進めていくため、多くのシステム開発では、システムがソフトウェア技術に依存する度合いを低くし、ソフトウェア技術を容易に交換するレイヤアーキテクチャパターンを採用している[2]。

しかし、レイヤアーキテクチャパターンは、システムと陳腐化したソフトウェア技術との間の依存性をなくしたに過ぎず、システムが登録したデータは、依然して陳腐化したソフトウェア技術に依存したままである。その為、データを新たなソフトウェア技術に対応させる設計と開発が必要となり、その結果、システムはソフトウェア技術のリプレイス開発がなされないまま運用が続けられている。

以上の点を踏まえ、データが陳腐化したソフトウェア技術に依存したまま、新たなソフトウェア技術へ移行方式を提案する。本提案方式ではまず、レイヤアーキテクチャパターンで分割したある層を、上位階層に対してシステムが取り扱うデータのCRUD（Create Read Update Delete）機能をソフトウェア技術に非依存で実装したApplication Programming Interface 部(以後APIと呼ぶ)と、ソフトウェア技術に依存したAPIの実装であるAPIソフトウェア実装部（以後実装部と呼ぶ）とに分離する[2]。次に、開発したAPIと移行元のAPI実装を用いてデータを収集し、収集したデータをAPIと移行先のAPI実装を用いて移行先に登録する。このことにより、ソフトウェア技術の依存性に関わらず、すべてのデータを移行元から移行先へと移行することが可能となる。また、本提案方式を採用することにより、新ソフトウェア技術に対応したAPI実装の開発のみで、データの移行、そしてシステムの運用再開を実現することが可能となる。

以下では、APIやデータ移行を支援するツールとして設計と開発を行ったAPIを用いたデータ移行並びに検証ツール(Data Migration and Verification Tool、以下DMVT)のソフトウェア構成並びに、実システムへの適用結果について述べる。

2. ソフトウェア構成

本章では、APIやDMVTのソフトウェア構成について述べる。図1は、APIやDMVTのソフトウェア構成図である。

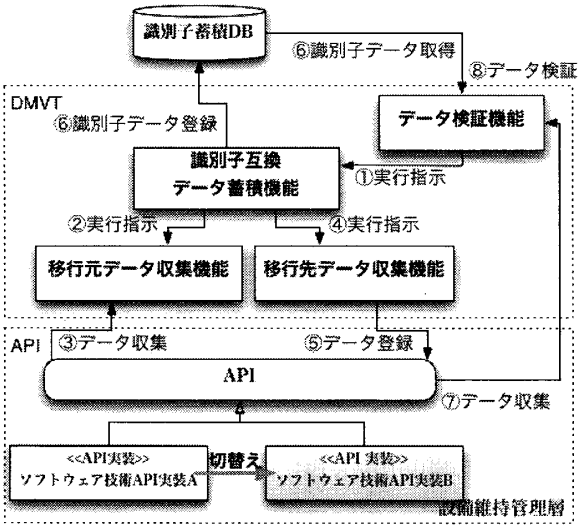

図1 DMVTのソフトウェア構成図

2.1. APIのソフトウェア構成

レイヤアーキテクチャパターンで分割したある層を、APIと実装部に分けることにより、APIを使用する上位層がソフトウェア技術に非依存になる。このことにより、用途に応じて、ソフトウェア技術に依存したAPI実装を使い分けることができるといった交換容易性が向上する。

しかし、交換容易性を実現するためには、APIが上位階層に提供する機能の振る舞いは、ソフトウェア技術に関わらず必ず同じでなくてはな

Proposal of Data Migration and Verification Using an API.
† Hirohisa Furuta, Akira Ishihara, Advanced Technology R&D Center Mitsubishi Electric Corporation
‡ Hirokazu Yamamoto, Kobe Works Mitsubishi Electric Corporation

らない。それを保証するため、API の試験プログラムとその試験プログラムを各実装に対して実行する試験実行環境を事前に準備し、いつでも API の自動回帰試験が実行できるようにした。

2.2. DMVTソフトウェア構成

DMVT は大きく分けて４つの機能から構成されている。

移行元データ収集機能： API が提供している取得機能を用いて、API が取得可能なすべてのデータを移行元から収集する機能である。

移行先データ登録機能：API が提供している登録機能を用いて、移行元データ収集機能により収集したデータを移行元に登録する機能である。

識別子互換データ蓄積機能：移行元データ収集機能で収集したデータと、そのデータを移行先データ登録機能で登録した結果のデータが、同一のデータであることを示すため、両データの識別子（例えばＩＤなど）を関連情報として蓄積する機能である。識別子を蓄積している背景には、移行元と移行先のソフトウェア技術が識別子を採番する方法が異なる可能性があるからである。

データ検証機能：識別子互換データ蓄積機能で蓄積した移行元、移行先の識別子を元に、それぞれ接続先の API を用いて、該当するデータを取得し、その上で、識別子を除くデータが保有するその他の値が一致しているかを検証する機能である。

3. 実システムへの適用

本章では、ある実システムに対して、ソフトウェア技術のリプレイス開発へ本方式を適用した結果について述べる。

3.1. 適用システムの構成と概要

図 2は、APIを用いて構築したシステムの構成図である。本システムは、ある企業や自治体が管理している設備を維持管理することを目的としたシステムであり、大きく分けて 4 つの層、画面などからなるプレゼンテーション層、画面からのリクエストを受け付けるアプリケーションコンポーネントなどからなるアプリケーション層、設備維持管理に特化した機能のAPI並びにその実装部からなる設備維持管理層、そしてデータを蓄積するデータベース層から構成している。DMVTは、設備維持管理APIをベースに設計・開発を行った。

3.2. 実験

今回のリプレイス開発では、実装部を採用していたソフトウェア技術の API 実装 A から、他システムで適用実績がある新たなソフトウェア技術で実装した API 実装 B に入れ替えを行った。その上で、DMVT に、移行元に採用していたソフトウェア技術の API 実装 A と移行先に新たなソフトウェア技術の API 実装 B を設定し、DMVT を用いてデータを移行した。

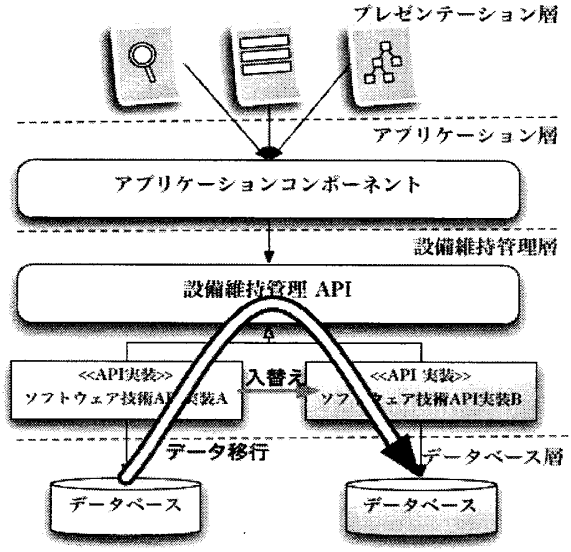

図 2 適用対象システム構成図

3.3. 実験結果

DMVT は、適用システムが保有していた約 420 万件のすべてのデータを、約 2 日かけて新たなソフトウェア技術のデータベースに移行した。そして、実装部に新たなソフトウェア技術の API 実装 B を切り替えることにより、システムの運用を再開することができた。

総取扱いデータ数	4,201,291 件
[内]設備・台帳定義データ	404 件
[内]設備・台帳データ	65,644 件
[内]その他（地図などのデータ）	4,135,647 件

表 1：システムのデータ件数について

4. まとめ

本稿では、システムのある層を API と実装部に分割し、その API を用いてデータ移行を支援することにより、新ソフトウェア技術に対応した API 実装の開発のみで、すべてのデータを移行元から移行先に移し、システムの運用再開まで実現することができた。今後は、どのような API が本方式で適用できるかについて検証を行い、アーキテクチャパターンの一つとして確立していくことが課題である。

参考文献

1. ＩＴアーキテクト　Vol 13 「システム統合技術の今を探る」、小野沢　博文（アクセンチュア　プリンシバル）、IDC Japan、2008 年 4 月。
2. フランク ブッシュマン（著）、金沢 典子（翻訳）：「ソフトウェアアーキテクチャ―ソフトウェア開発のためのパターン体系」、近代科学社、2000 年 12 月。