

## 異種医療機器間の接続支援のための 効率的なメッセージ変換システム

高桑 慧† 鈴木敦志† 岡田昌也† 坂根 裕†

静岡大学† デジタルセンセーション株式会社‡

### 1. はじめに

医療機器のネットワーク化が進むにつれ、病院間連携などで医療データを相互利用する試みが増えてきている。このような連携を阻害する要因として、個人情報保護やセキュリティなど運用面の問題もあるが、異なる医療機器間でデータがそのまま利用できないことも問題となっている。医療データの標準規格は存在するが[1, 2], 拡張性への配慮やフォーマット自体のバージョンアップ, 医療機器での実装誤りにより, 相互利用のためにはデータ変換が前提となっている。特に, CT (Computed Tomography) や MRI (Magnetic Resonance Image) などの高精細画像や動画は, データ変換の技術的敷居が高く, 有効な基盤システムは整備されていない。

筆者らは, 通信メッセージを動的に可視化・変換する機能を持つプロキシサーバを実現することにより, 医療画像及び動画を柔軟に変換し, 医療データの相互運用を広く可能にすることを目指している。本稿では, システム設計及び効率的なメッセージ変換方式の提案, 開発中のプロトタイプシステムを用いて実施した有効性検証について説明する。

### 2. メッセージ変換プロキシサーバ

#### 2.1. 基本機能

筆者らが開発しているメッセージ変換プロキシサーバは, 図 1 に示すようにネットワークグループの異なる拠点間 (病院間又は部局間など) の経路上に設置する。プロキシサーバは, 拠点間を流れるメッセージを一旦受信し, メッセージを処理後, 送信先へ再送信する。本プロキシサーバが持つ基本機能は以下の 3 つである。

- (1) 送信元・送信先アドレスから, メッセージ種別を判断し内容を解析する。
- (2) メッセージの解析結果に基づき, メッセージを可視化する。
- (3) メッセージ内容を, 事前に与えられたスクリプトに従い書き換える。

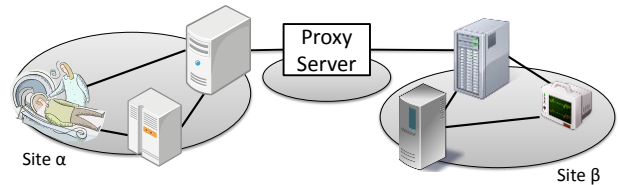


図 1 メッセージ変換プロキシサーバ

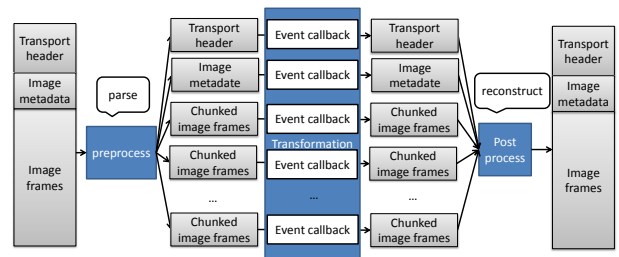


図 2 メッセージ変換方式

プロキシサーバとしてメッセージ変換の仕組みを用いる理由は, 既存の医療機器が継続利用できること (特別な機器への置き換えが不要) や, 既設ネットワークの構成変更を最小限に抑えられるためである。

#### 2.2. メッセージ変換方式

画像や動画データの変換は, データ構造が複雑なためデータ変換ルールの定義が難しいことや, データサイズが大きいためメモリ不足や処理時間が増大することが問題となる。

筆者らはこれまでに, サイズの大きなデータを, 一連のフラグメントに分割しストリーミング処理するフレームワークを設計した[3]。図 2 に, 提案したフレームワークに従い医療用画像フォーマットである DICOM (Digital Imaging and Communication in Medicine) を処理する例を示す。

データ処理は, データをフラグメントに分割する前処理と, データ変換処理, それらを結合して出力データを生成する後処理の 3 ステップで進める。変換処理は, 1 つのフラグメントが前処理で抽出される毎に, 特定の処理関数を実行することで進める。このことより, メッセージ変換スクリプト (変換ルール) は, 処理エンジンから呼び出される特定の関数を実装することで行う。開発中のプロトタイプシステムでは,

Efficient Data Transformation Proxy Server for Medical Information Systems

Satoshi Takakuwa†, Atsushi Suzuki†,  
Masaya Okada†, Yutaka Sakane†

† Shizuoka University

‡ Digital Sensation Co., Ltd.

開発言語として C#を用いている。

### 2.3. 既存システム

プロキシサーバでメッセージ変換を行う方式を採るシステムとして、Mirth[4]が広く知られている。Mirth は、さまざまな医療データのフォーマットに対応しており、JavaScript で変換スクリプトを記述できる。しかし、処理対象は比較的メッセージサイズの小さなテキストなどを中心としている。理由は、メッセージを一旦メモリに読み込んでからデータ処理し、再送するためである。2.2 節で述べたように、サイズの大きな画像や動画では、「一括して読み込んで処理する」という処理モデルはうまくいかない。

### 3. 性能評価実験

提案手法のストリーミング処理が DICOM データに対して有効であるか調査するため、開発中のプロトタイプシステムと Mirth との比較実験を行った。

#### 3.1. 処理性能

サイズの異なる DICOM ファイル (0.3MB, 0.5MB, 0.8MB, 4.4MB, 8.7MB, 901MB) を送受信し、その応答時間とスループットを測定した。このときプロキシではデータ変換を行わず、受信メッセージをそのまま再送信するものとした。

測定結果を図 3 に示す。結果から、提案手法がより効率的に処理できていることがわかる。メッセージサイズが 901MB の DICOM は、Mirth では処理できずエラーとなった。

#### 3.2. スクリプト記述

DICOM メッセージ中の特定のメタデータを変更 (画像メタデータの“Specific Character Set”を“ISO 2022 IR87”に変更) するという処理を、提案システム (C#) と Mirth (JavaScript) で記述した。図 4 にそれぞれのスクリプトを示す。

図 4(a) の C#コードは、参照定義やクラス定義、関数の引数などで見かけ上の行数は多いが、実際に変換する部分のコード量は、どちらも変わらない。

### 4. 結論

医療画像などのサイズが大きなメッセージは、ネットワーク上のプロキシサーバでストリーミング処理することで、効率的に変換し共有できる見通しを得た。

**謝辞** 本稿は、総務省・戦略的情報通信研究開発推進制度(SCOPE)地域 ICT 振興型研究開発「地域医療連携における異種医療機器間の「しなやかな」メッセージ交換ツールの研究開発(102306012)」の支援を受けて行った。ここに記して謝意を表す。

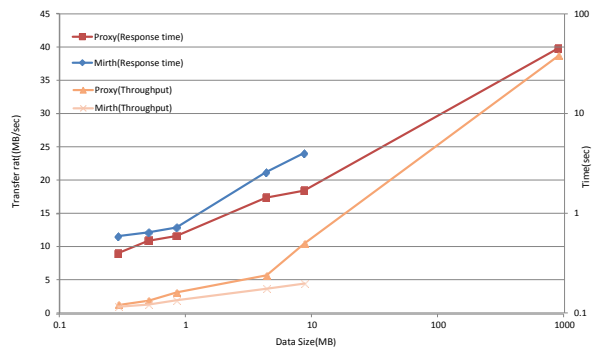
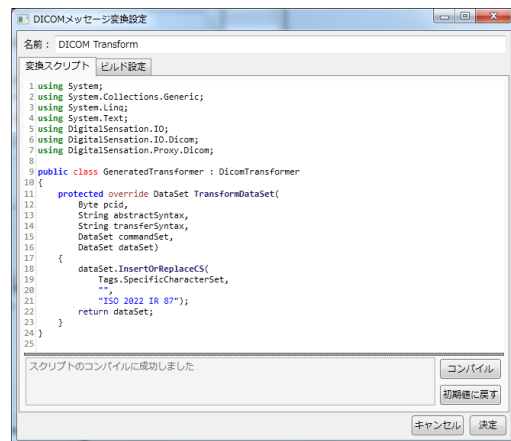
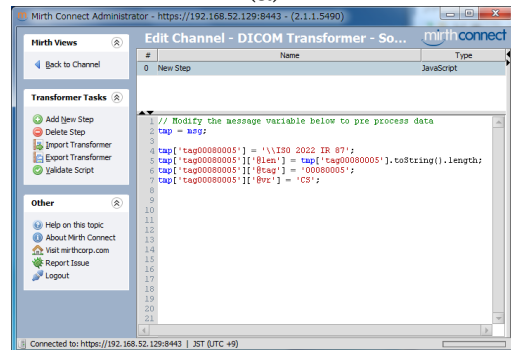


図 3 変換処理における Mirth との性能比較



(a)



(b)

図 4 スクリプト記述の比較

### 参考文献

- [1] HL7: <http://www.hl7.org>
- [2] DICOM: <http://medical.nema.org>
- [3] 坂根, 鈴木, 岡田: “通信プログラム開発を支援するデータ入出力および検証コード生成,” 情報処理学会研究報告. UBI, 2011-UBI-29(9), pp.1-6 (2011).
- [4] Bortis, Bren: “Experiences with Mirth: an open source health care integration engine,” Proceedings of the 30<sup>th</sup> international conference on Software engineering (ICSE’08), pp.649-652 (2008).