データ交換用途における XML の実力

富士ソフトＡＢＣ 株式会社

■ 執筆者Ｐｒｏｆｉｌｅ ■

須　田　浩　章

1984 年　富士ソフトウェア(株)入社
　　ソフトウェア開発部
1993 年　設計グループ
2003 年　IT 事業本部エキスパートハウス
　　XML 室所属

■ 論文要旨 ■

旧来からシステム間のデータ交換には，依存性の低さや取扱いの容易さから CSV 形式（Comma Separated Value format）などのテキストベースのデータ形式が広く用いられてきた.

現在でもデータ交換用途に CSV 形式は一般的であるが，いくつかの問題も露呈してきている．そのような状況の中で数年前からデータ交換用のデータ形式として XML (Extensible Markup Language) が寵児として取り沙汰されてきている.

本稿では，CSV 形式の問題点と XML の特長に視点をおき，実際のシステム開発においてデータ交換用途にどのように XML が利用できるのか，また開発したシステムにおける XML の効果と今後の展開について考察した.

■ 論文目次 ■

１．はじめに ……………………………………………… 《 3》
１．１　当社概要 ……………………………………………… 《 3》
１．２　データ交換の重要性 ……………………………………… 《 3》

２．システムの概要 ………………………………………… 《 4》
２．１　システム要件 ………………………………………… 《 4》
２．２　システム概要 ………………………………………… 《 5》

３．CSV 形式の問題点 ……………………………………… 《 6》
３．１　システム固有の問題点 ………………………………… 《 6》
３．２　一般的な問題点 ……………………………………… 《 8》

４．XML で解決できるか …………………………………… 《 9》
４．１　システム固有の問題 …………………………………… 《 9》
４．２　データフィールドに変化があった場合 ………………… 《 10》
４．３　形式の変換 …………………………………………… 《 11》

５．システムの評価 ………………………………………… 《 12》
５．１　システムの評価 ……………………………………… 《 12》
５．２　XML の評価 …………………………………………… 《 12》

６．おわりに ……………………………………………… 《 13》

参考文献 ………………………………………………… 《 14》

■ 図表一覧 ■

図１　既存システムと新規システムの連携 ………………… 《 3》
図２　システムのイメージ ………………………………… 《 4》
図３　システムの概要 …………………………………… 《 5》
図４　プレーンテキストの電子メール ……………………… 《 7》
図５　HTML 形式の電子メール …………………………… 《 7》
図６　データフィールドの変化 …………………………… 《 8》
図７　複数データ形式への対応 …………………………… 《 9》
図８　中間形式の構成 …………………………………… 《 10》
図９　データの構造 ……………………………………… 《 11》
図１０ XSLT によるデータ変換 …………………………… 《 12》

１．はじめに

１．１　当社概要

当社は 1970 年の創立以来，独立系ソフトハウスとして，特定分野に偏らないソフトウェアの受託開発を中心に発展してきた．

コンサルティングからシステム運用に至るまで，「総合システムビルダー」として，IT にかかわるすべての技術サービスを提供している．

１．２　データ交換の重要性

IT 不況が騒がれる昨今にあって，大規模なシステムの新規開発は減少の一途をたどっていると言わざるを得ない．そのような状況もあって顧客からは，既存システムを最大限に活用した短期間でのシステム開発が要求されている．

既存システムに新たな機能を追加する場合，すべてを新規開発するのではなく，**図１**に示すように付加機能部分とデータ交換機能部分を新規に開発し，既存システムとはデータ交換のみの疎結合によって連携する解決手法が広く用いられている．

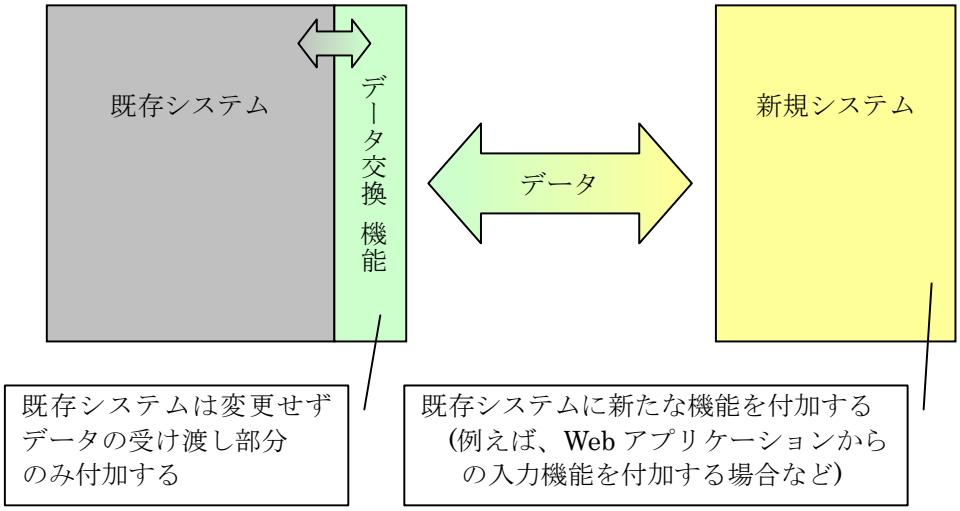

図１　既存システムと新規システムの連携

このような方法でのシステム連携は，顧客の要求とも相俟ってその重要性が今後とも高まってゆくことは想像に難くない．

既存システムと新規システムとのデータ交換には，旧来から一般的に CSV 形式(Comma Separated Value format)が広く用いられてきた．しかし，CSV 形式にはいくつかの問題点が指摘されており，将来的なシステム拡張に有効ではないと言われている．

本稿では，CSV 形式の問題点を提示しデータ交換の新たな寵児として注目されている XML (Extensible Markup Language) でどのように解決されるのかを実際に開発したシステムの結果から考察する．

２．システムの概要

今回開発したシステムは，既存システムと新規システムの連携ではなく，既存システム間の連携がテーマであった．しかし，データ連携を検証する目的としては，より適していると判断しここに紹介することとした．

２．１　システム要件

パーソナルコンピュータや携帯電話などをはじめとするさまざまな機器で電子メールが利用でき，情報の交換手段としての電子メールがすっかり根付いた感がある．

しかし，パーソナルコンピュータと携帯電話間などでの電子メールのやりとりにはいくつかの制限が存在する．例えば，タイトル（Subject）の文字数やメール本文の文字数，添付ファイルの有無などがそれである．

パーソナルコンピュータの電子メールと携帯電話などのさまざまなメールサービスとの連携を実現するミドルウェアを開発することが今回のシステム要件である．

また，将来的にデータを解析し宛先によって処理を変更したり，広告を掲載したりするなどの付加機能も検討されている．

イメージしているシステムの姿を**図２**に示す．

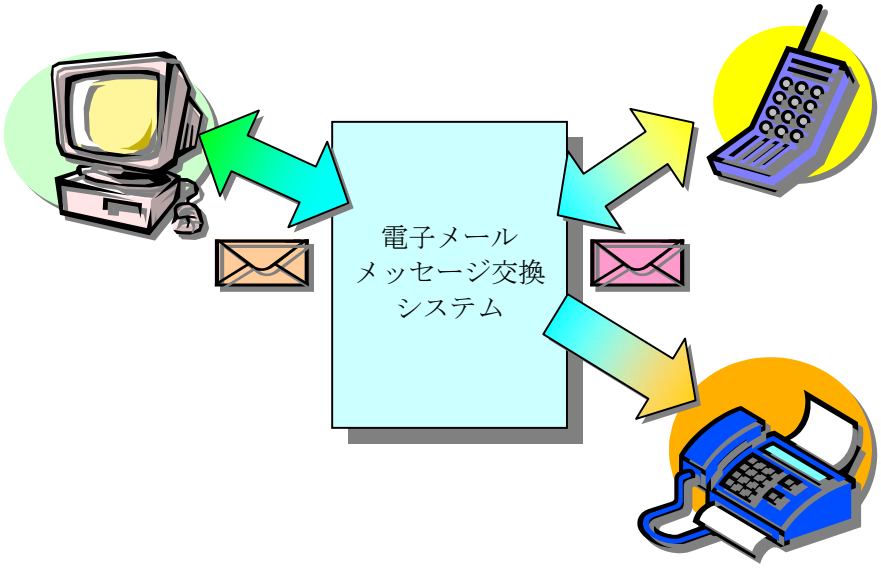

図２　システムのイメージ

端的に表せばこのシステムは，パーソナルコンピュータから発信された電子メールを何らかの形式に変換した後に携帯電話に送信したり，ファクシミリに送信したりする機能を実現するメッセージ交換システムである．

データ交換が中心となる今回のシステムの開発にあたり，そのデータ交換に用いるデータのフォーマットに関して，CSV 形式と XML について比較検討した結果，XML を採用するに至った．

以降開発したシステムの概要と XML を採用するに至った経緯，及び開発したシステムの評価について順次解説する．

２．２　システム概要

今回のシステムは，パーソナルコンピュータや携帯電話などから送信された電子メールを受信したメールサーバから取り出し，〝他のメッセージサービスの電子メール形式に変換〟して送信するという一連の作業のうちで，〝**変換**〟を実行する機能部分となる．

取り出されたある１つの電子メールデータは，複数のメッセージサービス及びファクシミリに送信する画像データに変換する必要がある．それらをふまえシステムの設計にあたり，以下の処理方式を検討した．

・メールサーバから取り出したデータを中間形式に変換

　ワンソースでマルチデバイスに対応する必要があるため，電子メールのデータを一旦処理しやすい中間形式に変換する．

・日付や差出人などの各種情報を明確にする

　中間形式は将来的な拡張のために，日付や差出人など各種の情報を明確にするとともに，第三者でも理解できるようにする．

・中間形式から目的に合った形式に容易に変換できる

　中間形式から目的の形式に変換する際の処理は，追加や変更が発生した場合でもプログラムを変更せずに対応可能とする．

中間形式のデータフォーマットとしては，テキスト形式で扱いやすいという面から，CSV 形式と XML 形式のいずれかを採用することにした．CSV 形式を採用した場合には，第３章に示すような問題点があることが分かり，最終的に XML を採用するに至った．

中間形式に XML を採用して開発した今回のシステム概要を**図３**に示す．

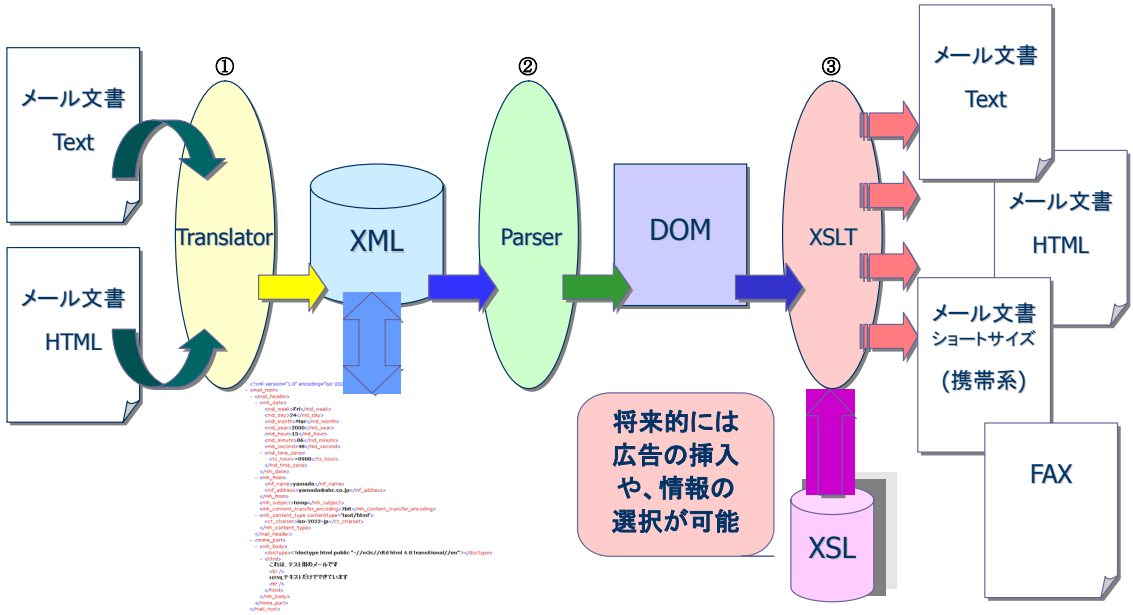

図３　システムの概要

システムの処理概要は以下の通りである.

(1) 電子メールデータから XML への変換

図３－①は電子メールデータから中間形式の XML へ変換する処理である.

中間形式となる XML のスキーマは, このシステム用に独自の仕様を策定している. また, HTML 形式の電子メールデータは一旦 XHTML 形式に変換した.

(2) XML をメモリ上に展開

図３－②に示す処理で, XML パーサにより中間形式の XML をメモリ上に **DOM** (Document Object Model) として展開する.

(3) 目的のフォーマットに変換

変換先の形式に合った **XSL** (Extensible Stylesheet Language) ファイルを選択し, **図３**－③に示す **XSLT** (XSL Transformations) にて変換する処理である.

この XSL を使用することより中間形式の XML をさまざまな形式に変換することができる.

それでは, 中間データの決定時に検討した CSV 形式の問題点と XML にてそれがどのように解決されるのかを見てゆこう.

３. CSV 形式の問題点

今回のシステム開発にあたり, 中間形式のデータフォーマットを今までデータ交換用途において一般的に利用されてきた CSV 形式にすべきなのか XML にすべきなのかに関して検討を行った. その結果 CSV 形式にはいくつかの問題点があることが明確になった.

今回のシステムに固有の問題点と一般的に CSV 形式が抱えている問題点とに分けて以下に記述する.

３. １　システム固有の問題点

今回のシステムでは, プレーン・テキストの電子メールデータ及び HTML 形式の電子メールデータを処理対象として取り扱う.

図４に示すようにプレーン・テキストの電子メールデータでは, 添付ファイル (Multi-Part) が複数存在するものの一次元的な広がりしか持っていないため, CSV 形式にて管理することが可能である.

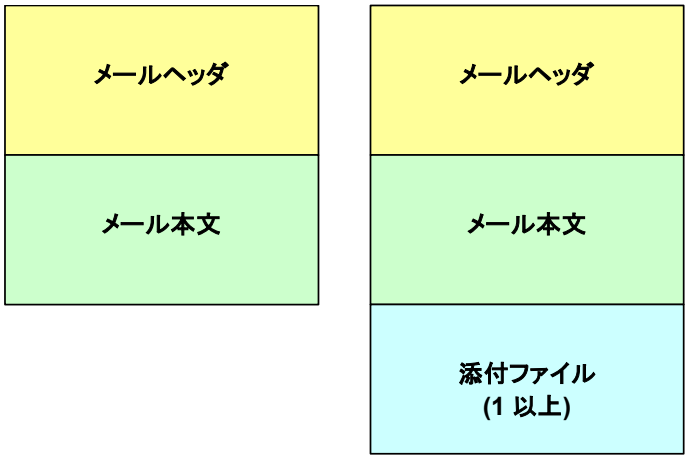

図４　プレーン・テキストの電子メール

しかし，**図５**に示す HTML 形式の電子メールデータでは，HTML 形式のメール本文が添付ファイルであったり，HTML 形式内に存在する画像データがさらにその添付ファイルとなっていたりしており，データが 2 次元的な広がりを持っている．

このような順序不定であり不定数の階層を持つデータを CSV 形式で管理することはとても困難である．

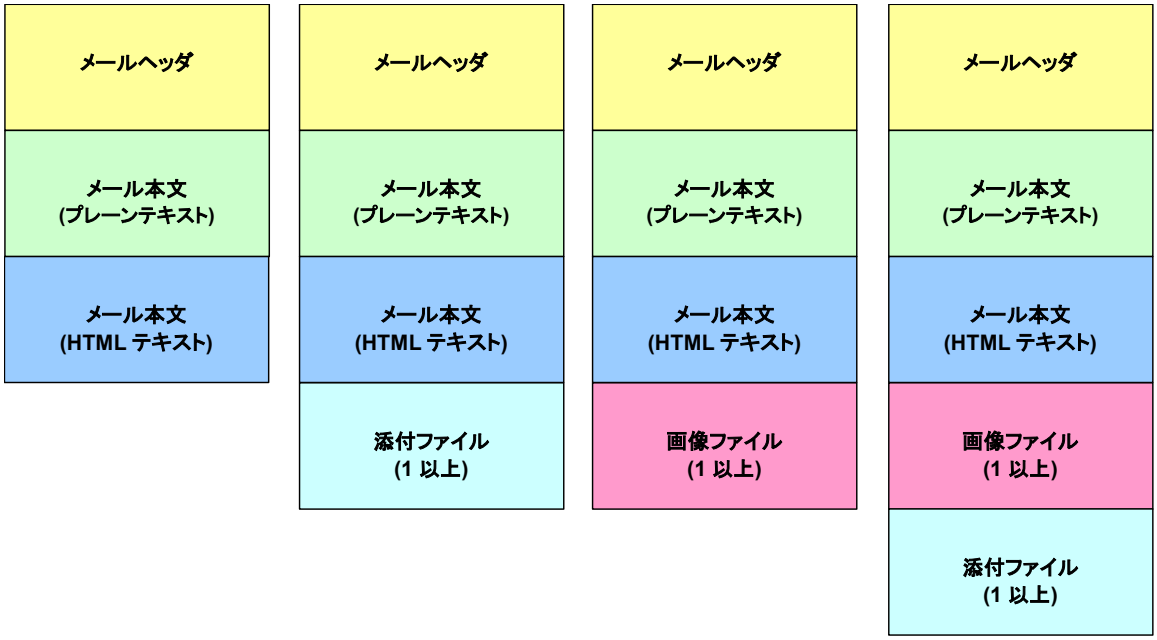

図５　HTML 形式の電子メール

日付や差出人を記述する「メールヘッダ」部分は，CSV 形式で管理することが容易だが，メール本文をプレーン・テキストや添付ファイルなどの各パートに分けて表現することは中間形式に CSV 形式を採用した場合，開発期間などを含め総合的に検討し不可能であると結論した．

３．２　一般的な問題点

システム固有の問題点の他に，CSV 形式には以下に列挙するような一般的に知られている問題点が存在する．これらの問題点は CSV 形式の特長としてとらえることもできるが，今回開発するシステムにおいてはやはり問題としてとらえざるを得なかった．

(1) データフィールドに変化があった場合

CSV 形式では基本的に「データの個数」及び「データの並び順」に意味があり，CSV 形式を処理するプログラムもデータの個数や並び順に応じた仕組みとなっている．

このような性質から，CSV 形式では不定形のデータを取り扱うことが困難である．**図６**に示すようにデータの個数や並び順に変化が生じてしまうと CSV 形式を処理するプログラム自体も変更する必要がある．

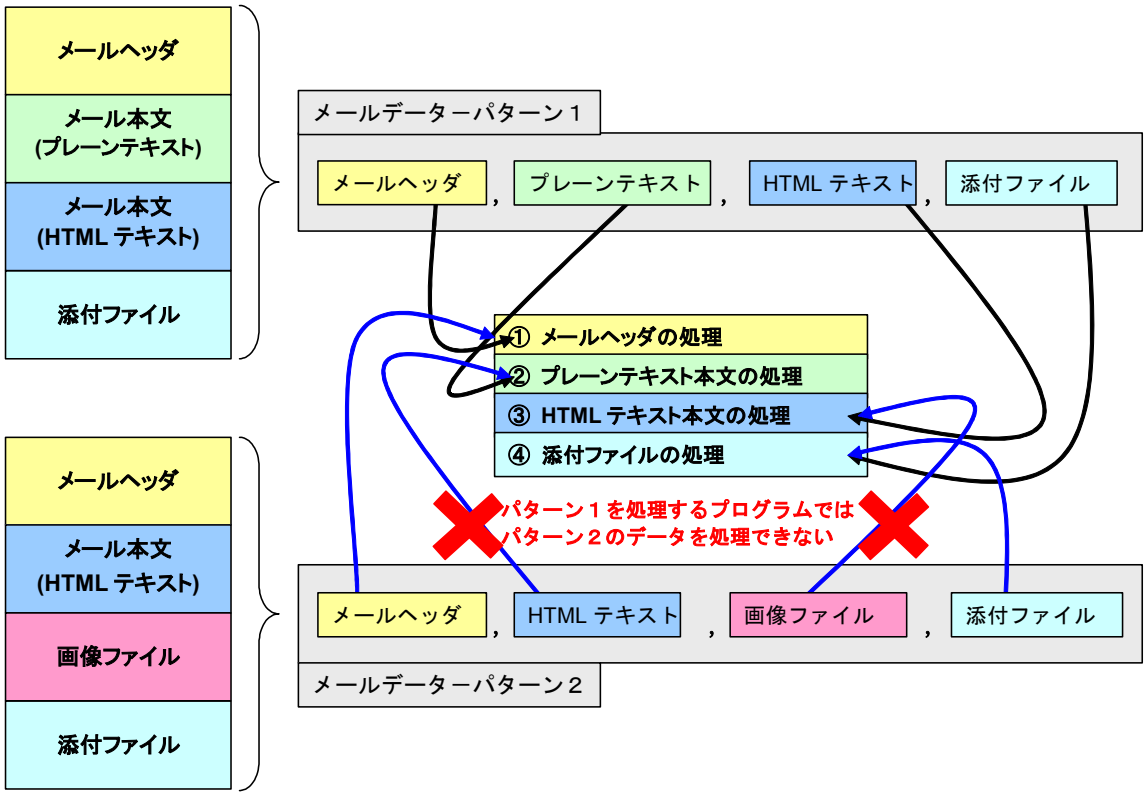

図６　データフィールドの変化

「３．１システム固有の問題」ではメールデータの 2 次元的な広がりを問題にしたが，単に処理対象のメールデータに HTML 形式のメールデータや添付ファイルなどが不定個数，不定順序で出現することも問題となる．その結果このような不定形のデータを処理するプログラムは複雑にならざるを得ない．

(2) 形式の変換

今回のシステムは，処理対象の電子メールデータを**図７**で示すように，さまざまな

形式に変換することが目的である．さらに，プログラムを作成せずに複数のデータフォーマットに対応したいという要求もある．

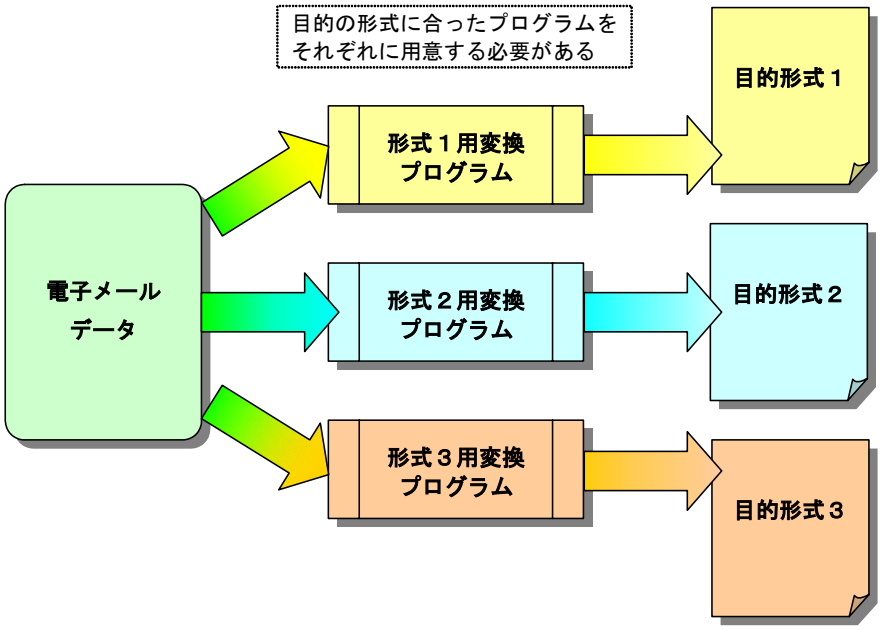

図７　複数データ形式への対応

さまざまなデータフォーマットへ変換するためには，CSV 形式を処理できるアプリケーションを用意したり，各データフォーマットに沿ったプログラムを新たに作成したりすれば対応することは可能だが，目的とするデータフォーマットのすべてに対して迅速に対応することは容易ではない．

また CSV 形式では，データが単なる文字の羅列になってしまい，一見しただけではそのデータが持っている意味が分からないと言う面ももう一つの問題である．ただし，今回のシステムでは大きな障害にはならないと判断した．

４．XML で解決できるか

それでは，第 3 章で掲げた CSV 形式の問題点が XML を採用することによって，どのように解決されるのかについて，以降順を追って説明する．

４．１　システム固有の問題

今回のシステムにて処理する対象のメールデータは，２次元的な広がりを持っているために，CSV 形式で表現することが困難であることを問題であるとした．この″２次元的な広がり″を別の言葉で表現すれば″階層的″になっているということである．

データ表現に焦点を絞って XML を見たときに，このような階層的な構造のデータを表現しやすいことが XML の特長としてあげられる．

つまり，CSV 形式で問題となっていた２次元的な広がりを持つデータ形式を XML では容

易に表現することができるのである.

今回のシステムでは，**図８**に示すような階層構造を持った中間形式を独自の DTD (Document Type Definition) を作成したうえで使用している.

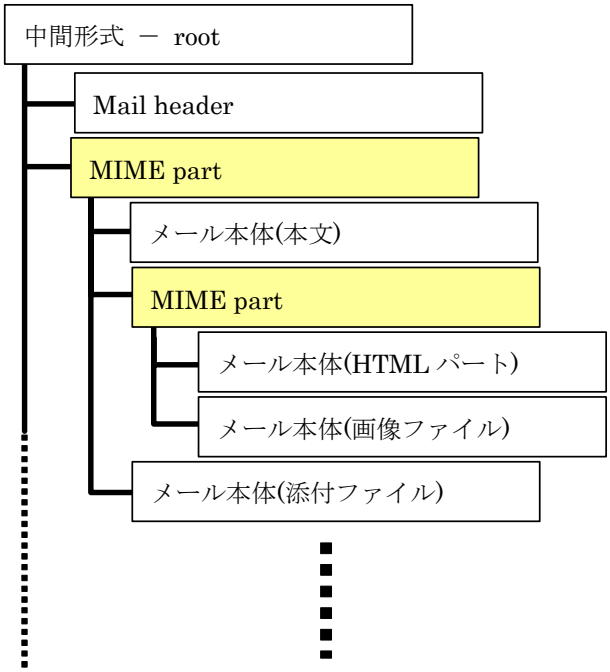

図８　中間形式の構成

また，HTML 形式の電子メールのフォーマットには，各電子メールクライアントソフトウェアによって大きな差違があり，また HTML の文法に合っていない場合が多いため，あらかじめ XHTML (The Extensible HyperText Markup Language) に変換してから中間形式に登録した.

４．２　データフィールドに変化があった場合

データフィールドの出現順序やデータフィールド自体の有無など，今回のシステムで取り扱うメールデータの形式が不定形であり，CSV 形式ではこのようなデータを表現しがたいことを問題とした.

XML では，このような不定形のデータであっても DTD などのスキーマを定義することにより，データ構造を的確に表現することが可能である．DTD の定義にもよるが，XML ではデータフィールドの出現順序やデータフィールドの有無に左右されないデータ構造を作成できる.

図９に示すように，データを要素に分解し各々の要素をタグで表現した中間形式では，順番が入れ替わったり要素そのものが存在していなかったりしても処理するプログラム自体に影響を及ぼすことはない.

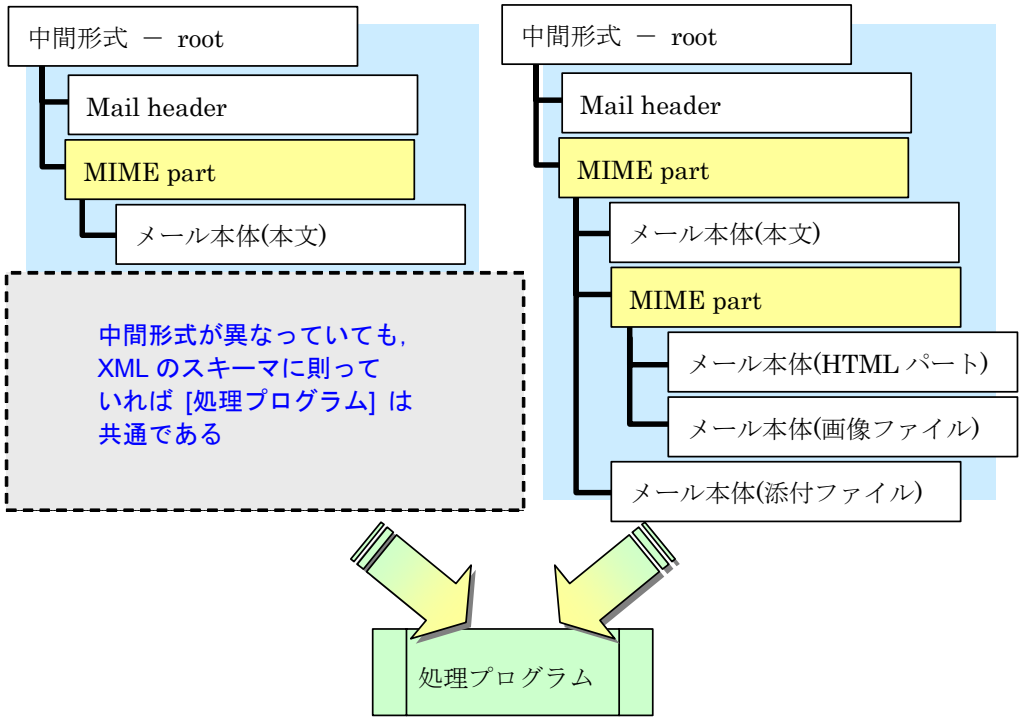

図９　データの構造

また，XML 文書は文法に誤りのない整形式であることがあらかじめ保証されているので，誤り判断をする基準がない CSV 形式と比較してデータ自体の信頼性が高いことも優位点の一つである．

４．３　形式の変換

CSV 形式のデータは，単にシステム間のデータ交換用途として考えれば十分に有効であるが，それ以降のデータ処理に関しては余り考えられておらず，データ処理のためには専用のプログラムが不可欠である．

今回のシステムでは，システム間のデータ交換用途だけではなく，さまざまな形式のメッセージデータに変換することが必要である．また，新たなデータ形式に変換する場合でも，プログラムを作成することなく対応したいという要求もある．

XML では，データを HTML やプレーン・テキストなど他の形式に変換するための仕組みとして XSLT が用意されている．XSLT は**図 10** に示すように，スタイルシートである XSL を記述することで，対象の XML 文書を意図した別の形式に容易に変換することができる．また，この XSL 自身も XML で記述された XML 文書である．

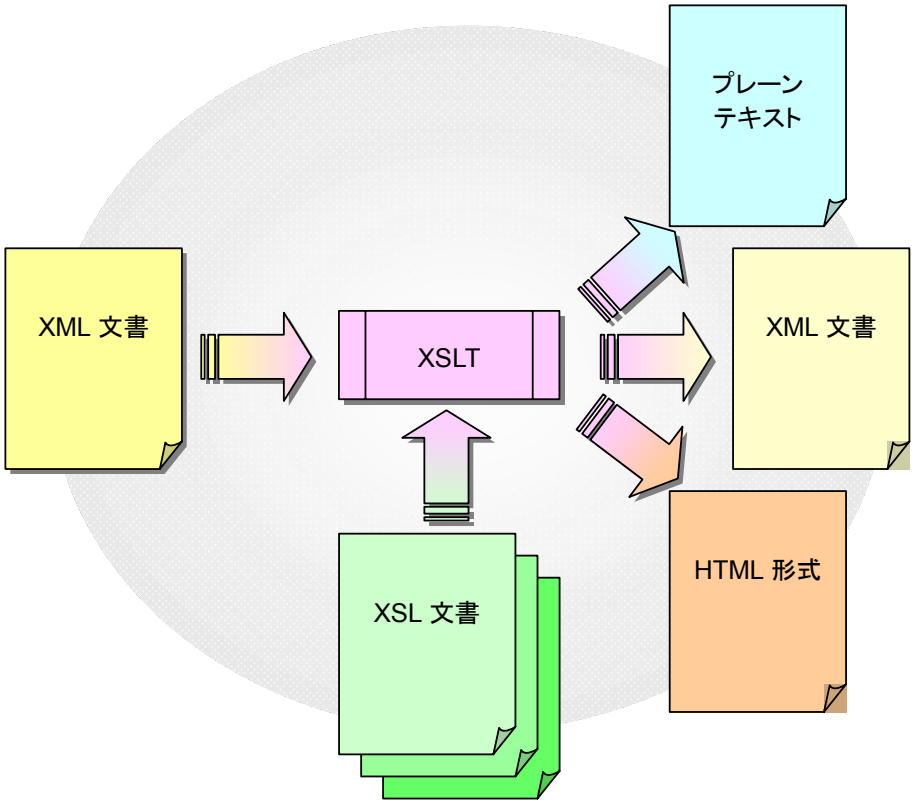

図１０　XSLT によるデータ変換

今回のシステムには，ファクシミリ向けの変換も機能として含まれている．しかし，XSL/XSLT にてイメージデータを作成することは困難であるため，一旦 HTML に変換してから別途レンダリングを行う方法にて実現した．

５．システムの評価

５．１　システムの評価

システム要件である「メールデータ」をさまざまな形式の「メッセージデータ」に変換するという目的を達成することができ，満足できるシステムとして完成した．

今回の開発にあたって最も時間を費やした処理は，対象となるメールデータの解析処理であった．特に HTML 形式のメールデータに関しては，その内容が各社各様であり，HTML の文法にすら則っていない状況であった．そのため HTML を XHTML に変換する処理を書き起こす必要が生じてしまい，思いもよらない時間を必要としてしまったのである．

５．２　XML の評価

データ交換用の中間形式として XML を採用したが，将来的な拡張も考慮し全体として CSV 形式を採用した場合よりも短期間に高品質のシステムを開発できたと思っている．

前章にて CSV 形式で問題となっていた事項は，XML を採用すれば解決できると結論した．ここでは，それらの問題点が最終的にどのように解決できたのかとりまとめておく．

(1) システム固有の問題

２次元的な階層構造を持つデータを表現する手段として，XML は優れていると評価できる．それは，さまざまな形式のメールデータを問題なく表現することが可能であったことから明らかである．

また，中間形式のデータである XML は，人間が見ても構造化されたデータとして各要素を容易に判別できることも評価できる．特に Microsoft 社の Internet Explorer などの一部の Web ブラウザでは，データをより見やすい形で閲覧できることがテスト段階でとても有意義であった．

(2) データフィールドに変化があった場合

今回処理対象のメールデータは不定形であり，中間形式を CSV 形式とした場合には，複雑な処理が必要であると予想された．しかし，XML の場合にはスキーマをきちんと定義することにより，このような不定形のデータを容易に取り扱うことができると実感できた．

また，"宛先"や"日付"など目的の要素を的確に取り出したり，変更したりできることも将来的にデータを加工する場合に有効である．

(3) 形式の変換

XSLT を利用した形式変換は，スタイルシートとなる XSL を記述することが必ずしも容易ではない．そのため，エンドユーザによる運用にあたって問題になると考えられる．

しかし，CSV 形式であれば，処理するプログラムを作成する必要があるということを考えれば，プログラミングせずに他の形式に対応できることは評価できる．また，この問題は XSL を作成するユーティリティの登場によって今後改善されるものと思っている．

６．おわりに

今回 XML を使用した開発を行い，システム要件を満たすことができた．

今回の開発経験から，今後のシステム開発に際して有効と思われるいくつかの注意点も明確になった．以下に箇条書にてそのいくつかを列挙する．

- データ交換に XML を使用する場合最も重要となるのはスキーマ定義である

  今回独自の DTD を作ったのだが汎用性がないという問題が残った．閉じた環境でのデータ交換であればこれでもよいが，通常このようなスキーマは，多くの人が共通に使用してはじめて意味を持つものである．
- 内部的に DOM にて処理を行っているため，処理対象の電子メールデータが巨大な場合，処理に時間がかかる

  大量データを処理する場合は，XML パーサの選択や SAX での処理を検討する必要がある．
- XSL の記述にはコンパイルなどの作業は必要ないが，プログラミング言語と同等に習得が難しい

XMLは，気づかないあいだに身の回りのさまざまな箇所に利用されており，Webサービス技術の中核となるデータ交換手段として欠かせない存在である．

XML及びその関連技術は，EAI（Enterprise Application Integration）など社内システムのデータ交換用途として，既にCSV形式を凌駕している．今後は電子商取引をはじめとするさまざまなデータ交換の分野にてさらに利用されてゆくのは間違いない．

データ交換用途でのXMLの実力は思った以上に高く評価できるものであった，今後のシステム開発に今回の経験で得た知識を生かしてゆきたい．

参考文献

［１］RFC # 822：STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES
August 13 1982

［２］RFC # 1341：MIME (Multipurpose Internet Mail Extensions)
June 1992

［３］Extensible Markup Language (XML) 1.0 (Second Edition)
W3C Recommendation 6 October 2000

［４］XSL Transformations (XSLT) Version 1.0
W3C Recommendation 16 November 1999