

XML と Web を利用した携帯端末向け家計簿アプリケーションの開発

横田寧子 野村直之
ジャストシステム

1. はじめに

インターネットへ接続できる携帯端末には、携帯電話、PDA、ハンドヘルド PC などがある。とりわけ、日本では普及率の高さ¹⁾ から、携帯電話を中核としたモバイル・インターネットビジネスの発展が期待されている。インターネットビジネスを支える基盤技術としては、情報の構造記述言語 XML [XML98]と関連技術が広く採用されつつある。XML 技術は、データの再利用性を高め、開発コストの軽減や、相互運用性の向上につながるとされる[野村 01]。XML は、アプリケーションのデータの基本形式、企業間電子商取引のインフラとして利用が進んでいるが、ブラウザ、プラグインなど Web 閲覧環境の XML 対応によって、Web サーバ上の統合的な表現形式としての能力も獲得しつつある。i モード、EZweb など、携帯電話からインターネット上の Web コンテンツにアクセスするサービスが実現している現在、Web 上の XML データを携帯電話で扱うことの重要性も大きくなるとされる。実際、Java 対応などにより、その実現には近づきつつあるのではなかろうか。

本稿では、i モード対応の Web ベースの家計簿アプリケーションの開発事例の紹介を通じて、Web アプリケーションの開発における XML 技術の利点を確認し、それを携帯電話を使った Web サービスの観点で位置づけることを試みる。まず、家計簿アプリケーションを Web ベースにした背景、メリット、およびシステムの基本構成について述べる。次に、XML 技術の活用状況を述べるとともに、開発コスト削減効果について確認する。また、特に円グラフ描画で顕在化した、XSLT、SVG の標準仕様だけでは機能要件を満たせない問題を取り上げ、複数の解決方法間の得失を議論する。その後、携帯電話からの Web アクセスにおいてコンテンツ記述言語が多様化し、発散しかねないように見える現状を概観し、近い将来この問題を解決するために XML 技術がどう貢献できるかを論じる。最後に、今後の課題と展望を述べる。

2. Web ベースの家計簿アプリケーション

従来の典型的な家計簿アプリケーションの利用モデルは、1つのマシンにソフトウェアをインストールし、そのマシンで利用する、という形態であった²⁾。この形態では、例えば自宅にいるときしか家計簿をつけられない、といった不便があった。これに対して、本稿で紹介するアプリケーションは、WWW 上に家計簿データを持ち、CGI や Servlet によって U/I を提供することで、インターネットに繋がり HTML のブラウザ機能を備えるすべてのパソコンや携帯端末上で利用可能とした。これにより、例えば買い物の帰り道に歩きながら家計簿をつけるということが可能となり、大きな利便性向上につながる。

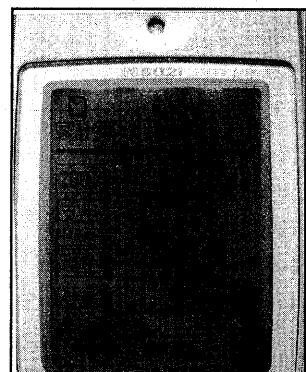


図1 携帯電話の入力画面

*1 2000 年 12 月末時点で、i モード利用者数だけで 1700 万人を越えている。

*2 例えばシェアウェア『ゆう子の家計簿』 <http://village.infoweb.ne.jp/~fwgj9552/index.htm>

図2に、システム構成の概要を示す。入力と閲覧の2つの基本機能からなり、入力とは携帯電話・PCの両方から、閲覧はPCからのみ行うことができる。入力機能は、携帯電話・PCとも共通のCGIプログラムを使用する。携帯電話からWWWへのアクセス手段としては、(株)NTT DoCoMoの「iモード」を利用した。CGIは、HTML言語のサブセットの範囲でCompact HTML言語[Compact HTML98]を用いて画面を出力する。一方、閲覧機能については、詳細は後述するが、表示用CGIを用意するのではなく、XMLデータをXSLTスタイルシートでレイアウトしてXSLT対応ブラウザ^{*}で表示させることとした。

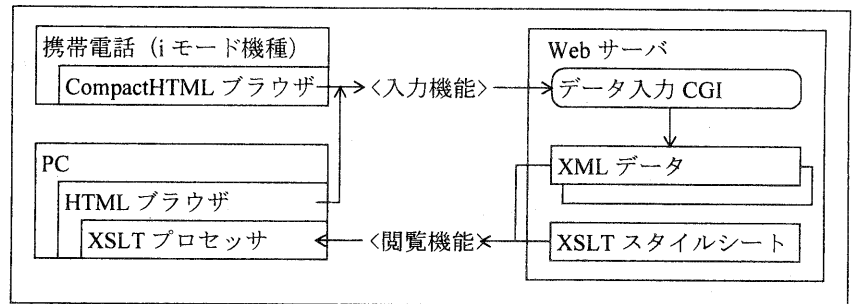


図2 iモード家計簿のシステム構成

3. XML 技術の活用と効果

3.1. XML データ

iモード家計簿では、家計簿データを月毎にXMLファイルとしてWebサーバ上に格納する。具体的には、収入、支出などのデータ1件をXMLの要素1つに対応させ、データ種類・日付・金額・口座・費目・備考を、要素の属性で表現した(図3)。その上で、家計簿が備えるべき各機能を、開発期間の数日程度への短縮を念頭において後述のように設計した。

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<monthly year="2001" month="1">
  <line type="check" day="1" account="現金" amount="10000"/>
  <line type="outgo" day="1" from="現金" amount="230" group="交通費">地下鉄</line>
</monthly>

```

図3 2001年1月の家計簿データの例；残高1万円の財布から地下鉄の切符230円を支出

3.2. 一覧機能

月ごとの収入・支出データの一覧機能は、XMLデータをHTML形式にレイアウトするための構造変換情報をXSLTスタイルシートで記述し、XSLT処理機能を備えるブラウザで表示させる方法を採用した[XSLT99]。単にHTMLに変換するだけでなく、収入・支出の合計や各口座の残高の計算もXSLTと

*1 Microsoft社のInternet Explorer 5以上

XPATH を使って表示時に行わせることができる [横田 00]。専用の CGI/Servlet の開発に換えて、ポータブルな XSLT スタイルシートの開発だけで済ませることができるので、開発コスト・部品の再利用性の両面で、XML 活用の効果があったといえる。また、XML で記述した家計簿データはアプリケーションが解釈できるだけでなく、人間が見ても意味を理解しやすいので、一般的な XML データ、あるいは単なるテキストデータとして、汎用的なツール類をつかって閲覧・編集することが十分に可能である。そのため、開発者にとってデバッグコストが低いことも利点となる。

日付	種類	金額	口座	書目	摘要
1	残高	10061	現金		
1	残高	128606	XML銀行		
1	支出	230	現金	交通費	地下鉄
1	支出	500	現金	趣味・娯楽	雑誌
2	移動	(10000)	XML銀行 → 現金		
2	支出	672	現金	食費	お弁当
4	収入	2000	現金	贈答	

収入合計	2000
支出合計	1402
収支	598

現金	XML銀行
20659	118606

図4 一覧表示画面

3.3. グラフ機能

グラフ機能の実現には、グラフ図形をベクター図形記述言語である SVG[SVG00]で表現し、ブラウザのプラグインによって表示する方法を採用した。SVG は XML の個別言語であるため、家計簿データからグラフ図形への変換にも、基本的には XSLT を利用できた。図5に、Servlet による、グラフ変換機能の構成を示す。100 行に満たないポータブルな XSLT スタイルシートによるテキスト-図形変換は、図形フォーマットへの変換を独自にプログラミングするのに比べて、開発コストが明らかに軽減される。音声など、異なるメディアのデータへの変換も、各メディアの記述言語が XML で整備されることにより、同様に軽減できることが予想される。

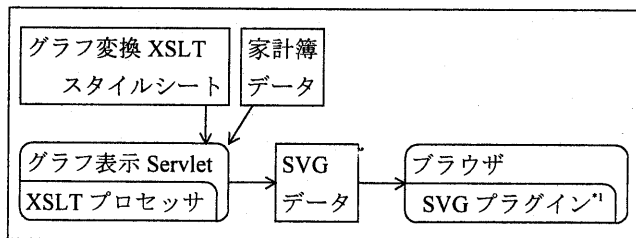


図5 SVGを使用したグラフ描画機能のシステム構成



図6 グラフ表示画面

3.4. XML 標準を拡張する手法間の比較

前節までで、グラフ閲覧機能への SVG の活用について述べた。実装設計の段階で判明した事実の1つは、現状では、作成したい図形によっては、XSLT, SVG の標準仕様、すなわち一般のプログラム言語の「組み込み関数」に相当するボキャブラリだけでは不足となり得る点である。実際、支出内訳の円グ

*1 Adobe 社の Adobe SVG Viewer

ラフを描画させる際に、この問題が生じた。

家計簿データの数値から各支出項目が全支出額に占める割合を百分率で計算するまでは XSLT で実行できる。これに 360 度の角度をかけ算して円グラフが素直に描画できれば問題はない。しかし、SVG でこのような扇形を描画するためには、角度指定による方法が標準では存在していない。円弧の開始点・終了点の座標を指定する必要がある (図 4)。そのためには、角度を引数にして sine, cosine の値を求める必要があるが、XSLT にはこれらの関数が標準では用意されていない。

このような問題を解決するには、

(1) SVG を拡張する^{*1}、(2) XSLT を拡張する、(3) 汎用のプログラム言語で個別処理ルーチンを記述する、等の方法が考えられる。本開発事例では、(2)、(3)を試してみた。すなわち、(2)に準拠して、XSLT プロセッサが拡張機能として提供している Java コード呼び出し機能 (一種のシステム・コール関数) の利用^{*2}を、また、(3)に準拠して、XSLT 変換結果を専用の Java プログラムでさらに一部変換、の 2 つの方法で実験を行った。その結果、少なくともこの円グラフ描画の課題に関しては、XSLT に関数を追加する考え方の前者のほうが、開発工数が少なく済んだ。

```
<?xml version="1.0" encoding="iso-8859-1"?>
<svg width="200px" height="200px" viewBox="0 0 200 200">
  <path d="M100,100 L100,20 a80,80 0 0,1 68,38 z" style="fill:red"/>
  <path d="M100,100 L168,58 a80,80 0 0,1 -77,121 z" style="fill:green"/>
  <path d="M100,100 L91,179 a80,80 0 0,1 9,-159 z" style="fill:purple"/>
</svg>
```

図 7 扇形の描画を行うための SVG ソースの例；

各 path 要素が扇形に対応。1 番上の path 要素の円弧は (100,20) で開始し (68,38) で終了する。

では一般に XML 関連の標準規格類では足りずに、機能拡張を行う際のガイドラインはどうなるであろうか。機能拡張の自由度の観点では、データ記述言語の SVG のみを拡張する (1) の手法が最も自由度が小さく、チューリング機械と等価な汎用のプログラム言語を一定の作法に従って呼び出す (3) の手法が最も自由度が大きい、といえる。(2) の手法、すなわち、XML インスタンス間のデータ変換言語 XSLT を用いるやり方はその中間と考えられる。実際、XSLT は、入力 XML 文書の要素名、構造、内容をパターン記述でマッピングさせ、XML 部分要素を変形、挿入、削除したり、内容や属性値の計算を行うのが基本であり、汎用のプログラム言語に備わっている if 文・switch 文・for 文・サブルーチン呼び出しを備える一方、for 文の break や while 文・配列変数などは備えていない。^{*3}

機能拡張の自由度が大きければ、自在な拡張が可能な反面、一定の制約の中での記述とならずに、一般に開発工数が増大し (バグの温床)、ポータビリティ、再利用性 (保守性、可読性)、高速プロトタイプ

*1 ただし、SVG の拡張があまりにも物理的な、非互換をまねくやり方だと、(2)、(3)よりも生産性の低下を招く可能性もある。しかし、規格に定義された作法通り foreignObject 要素によって別の言語、例えば MathML など子を要素として記述する手法によるならば、おそらくそのような問題は発生しないであろう。

*2 Apache XML Project の Xalan の提供する Java コード呼び出し拡張機能を利用。

*3 本来的な使い方ではないが、template の再帰呼び出しにより while 文と同様の処理を行うことは可能。

ピング、等で、一般に劣ったものになるであろう。そこで、もし(3)の方法に頼らざるを得ないのであれば、これらの欠点を少しでも補ってくれるプログラム言語の採用を検討すべきである。

この観点で、汎用のプログラム言語の中では次の3つの理由により、Java 言語が最も推奨されるべき、と判断した。

- －マルチプラットフォーム（OS、マシン）対応という点で、移植性が優れている；
- －C++との相対比較では、OOP を自然に強制してくれる Java が再利用性で優れている；
- －特に数値計算、グラフ描画に目玉機能がある家計簿アプリでは、Fortran 言語（FORmula TRANslator:数学公式翻訳器）を参考にして設計されたと言われる Java が十分に優れている；

4. Web リソース共有とコンテンツ記述言語の問題

コンテンツプロバイダにとっては、対象とする端末が異なっても、同じ内容のコンテンツであれば1つだけ用意して済ませることが望ましいと思われる。i モード家計簿でも、CGI プログラムを PC 用・i モード用で共有できたことは開発コスト削減に役だった。

携帯電話からインターネット上のコンテンツにアクセスする手段として、日本には現在、i モードのほか、KDDI グループの EZweb サービス、J-PHONE の J-SKY ウェブサービスがあり、コンテンツ記述言語も異なっている¹。そのため、1つのコンテンツをすべてのサービスから利用できるようにすることは難しい。

このようなコンテンツ記述言語の多様化は、端末が多様化していく中では、避けられない面があると思われる。従って、Web リソースの共有のためには、トランスコーディング技術²（記述言語間の自動変換）が必要である。現状では、記述言語間の双方向の変換処理を個別に開発する必要があるが、将来的には、XML 形式のマスターデータから各記述言語への変換を XSLT で行うことで、変換処理の開発効率を向上させられる可能性がある。また、個別の記述言語向けのタグを XML の名前空間をつかってマスターデータに埋め込むことで、コンテンツは1つのまま、複数の端末固有機能をそれぞれ活かす、といった Web 制作手法の実現も見込める。

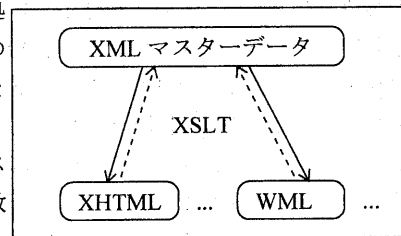


図8 XSLT による将来の記述言語変換

5. おわりに

5.1. 携帯電話と XML コンテンツ

次世代 Web は "Semantic Web" の方向を目指すということが言われている [野村 01]。i モード家計簿で扱う家計簿データは、アプリケーションのデータであるだけでなく、ブラウズ可能な Web コンテンツとしても位置づけることができる。XML コンテンツのブラウズ機能のキーとなる技術のひとつは、XSLT であろう。生の XML コンテンツを人間が直接見るのではなく、XSLT を用いて、SVG などの非テキストデータ、およびそれを包括する XHTML 複合文書の形にレイアウトしてブラウズすることが一般化

*1 i モードは CompactHTML, EZweb は HDML, J-SKY ウェブは MML・制限つき HTML

*2 製品に、IBM WebSphere Transcoding Publisher がある。

すると思われる。PC とブラウザの世界では、XML に対応ブラウザや SVG プラグインなどが実質無料で入手できるため、このような新しい Web への動きがすでに始まっている。しかし、携帯電話においてはまだ実現できる環境が存在しない。その一因は、携帯電話のブラウザに XML 処理系を組み込むのは処理が重すぎるからであろう。これに関しては、サーバ側で XML 処理系を動かすことなどによる解決が望まれる。なお、サーバ側で XML 処理系を動かすことには、クライアント側の処理を軽くすること以外にも、XML データを部分的にユーザーに見せないといったセキュリティ管理がしやすくなるといったメリットもある。

5.2. 家計簿アプリケーションの発展可能性

最後に、本稿で紹介した家計簿アプリケーションの展望を述べる。家計簿ソフトとしての機能充実のほか、モバイル・XML というキーワードで、1) 入出力メディアの拡大、2) 他サービスとの連携、といった広がりが見えらる。

1) については、音声による入出力、商品のバーコード読みとりによる入力、また、特に携帯電話では銀行の振り込み通知のような自動電話連絡サービスとの連動による家計簿自動記入といった発展が期待できる。

2) に関しては、XML の重要な潜在能力として、相互運用性 (interoperability) が指摘されている。すなわち、あらかじめ接続させることを特に意図していないシステム同士を低コストで連携させることができる。たとえば、家計簿内の食品購入データと、料理レシピ DB や料理手順の構造化記述 [浜田 00] とを連動させて、材料に見合う料理レシピをリコメンドするサービスなどが考えられる。また、確定申告が必須となっているアメリカでは、申告の手間の軽減や節税のためにクイッケンや MS Money などの家計簿ソフトが広く普及している。その強力なシミュレーション機能などから、預金・年金・金融商品のライフプラン設計の基盤となるプラットフォームと位置づけることができるという [竹中 00]。日本でもネットでの確定申告開始などによって家計簿ソフトの需要が大きくなれば、オンライン証券、オンラインショッピングなどとの連携を効率的に実現する上で、XML 技術の相互運用性が役立つと思われる。また、アメリカでは家計簿ソフトは Windows アプリケーションとして普及したが、これから普及が進むとすれば日本においては ASP などの形態が主流になる可能性がある。そうなれば、本稿で述べてきた開発コスト軽減の利点などとあわせ、XML 技術がさらに大きなメリットを生むであろう。

参考文献

[XML98] Extensible Markup Language 1.0, W3C Recommendation (1998)

<http://www.w3.org/TR/1998/REC-xml-19980210>

[野村 01] 野村直之：ナレッジマネジメントツールの配備, 実践動向と次世代技術, 人工知能学会誌 Vol.16 No.1, 2001

[CompactHTML98] Compact HTML for Small Information Appliances, W3C NOTE 09-Feb-1998

<http://www.w3.org/TR/1998/NOTE-compactHTML-19980209/>

[XSLT99] XSL Transformations (XSLT) Version 1.0, W3C Recommendation (1999)

<http://www.w3.org/TR/1999/REC-xslt-19991116>

[SVG00] Scalable Vector Graphics (SVG) 1.0 Specification, W3C Candidate Recommendation (2000)

<http://www.w3.org/TR/2000/CR-SVG-20001102/>

[横田 00] 横田寧子：XML で実現する「i モード対応家計簿」, IDG Japan 月刊 JavaWorld 2 月号

[浜田 00]浜田玲子，井出一郎，坂井修一，田中秀彦：料理教材における手順の構造化，情報処理学会第 60 回全国大会（平成 12 年前期）講演論文集，2000

[竹中 00]竹中平蔵：IT 革命は国民運動，日経ビジネス人文庫『経済を見る目はこうして磨く』，2000