B-016

エンドユーザによるWebアプリケーション開発技法の提案と試作
A Method for Web Application Development by End-Users

八木 紀幸 † Noriyuki Yagi
中所 武司 † Takeshi Chusho

1. はじめに

インターネットの普及とともに、Web アプリケーションが増大し、ASP(Application Service Provider) や Web サービスに加えて、SOA(Service-Oriented Architecture)、SaaS(Software as a Service) などのキーワードが注目されるなど，ソフトウェアのサービス化が促進されている。

我々は、変化の激しい時代には、エンドユーザ主導のアプリケーション開発とその保守が重要になるという観点から、研究を行ってきた。特に、小さな部門や個人の業務を対象とする中小規模の Web アプリケーションに関して、低コストで短期間に開発するとともに、頻繁な機能変更を伴う保守に対応するために、その分野の業務の専門家主導で開発・保守できるような技法を研究してきた。

2. エンドユーザ主導のシステム構築

エンドユーザ主導の基本的なシステム構築技術を図 1 に示す [1]。ビジネスレベルでエンドユーザ（業務の専門家）が構築したビジネスモデルは、サービスレベルでは、ドメインモデル（ワークフローを示す業務モデルなど）に変換され、アプリケーションの原型ができる。最後にソフトウェアレベルで、コンポーネントを組み合わせたアプリケーションを構築する。このとき、サービスとソフトウェアの間の粒度的なギャップは、フレームワークやパターンあるいは業務コンポーネントなどの CBSE（Component-Based Software Engineering）技術で解決できる。一方、ビジネスとサービスの間のギャップについては、エンドユーザに理解容易な電子フォームで解決する。本論文では，ビジネスロジック定義のためのフォーム変換ツールとＤＢ操作用スクリプト言語について述べる。

3. ビジネスロジックの定義

3.1 サービス授受のメタファー

ワークフローが本質的な Web アプリケーションの要求分析では、詳細なワークフローを記述する必要がある。このモデルは、オブジェクト指向技術を用いた場合は、お互いに協調して動作するオブジェクト間を流れるメッセージフローとして表現できる。ここで抽出したオブジェクトに対応する業務コンポーネントが存在する場合は、この段階でアプリケーションを構築できるが、通常は、新規に開発すべきコンポーネントが存在する。

そこで、エンドユーザ主導開発の一環として、新規コンポーネントの要求仕様は、業務の専門家になじみのあるフォームとして定義することとする。さらに、ワークフローを Web サービス化することにより、業務コンポーネントをサービスプロバイダと見なすことができ、ビジネスレベルでの理解が可能となる。

†明治大学大学院理工学研究科基礎理工学専攻情報科学系ソフトウェア工学研究室

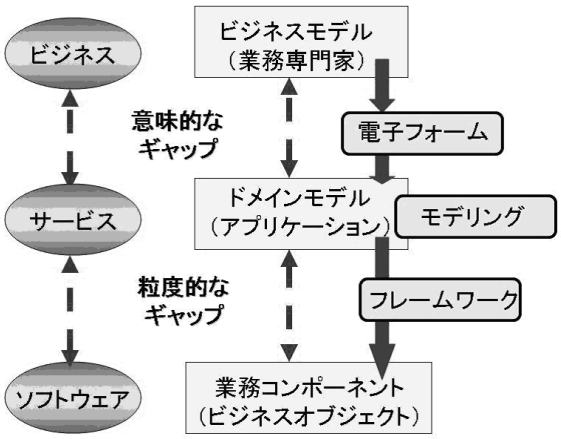

図 1: エンドユーザ主導のシステム構築技術

すなわち、窓口業務をサービス授受のメタファーとみなして、そのインターフェイスをフォームとすることにより、ワークフローをフォームフローとフォーム変換ととらえる。

3.2 フォーム変換定義

エンドユーザに XML や XSLT の構造を一切意識させない方法としては、入力フォームと出力フォームの項目間の関係をマウス操作だけで定義する方法が考えられる。図 2 に示すように、特定の入出力フォームに関して定義した変換手順を用いて、同じ形式の入出力フォーム間の変換を自動実行する。

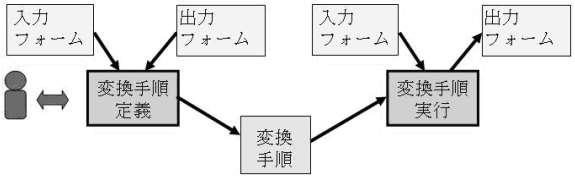

図 2: フォーム変換手順の定義と実行

3.3 ツールの機能と実装

3.3.1 例題アプリケーション

ツールの機能を説明するために、商品販売システムを例題として考える。そのフォームフローを図 3 に示す。

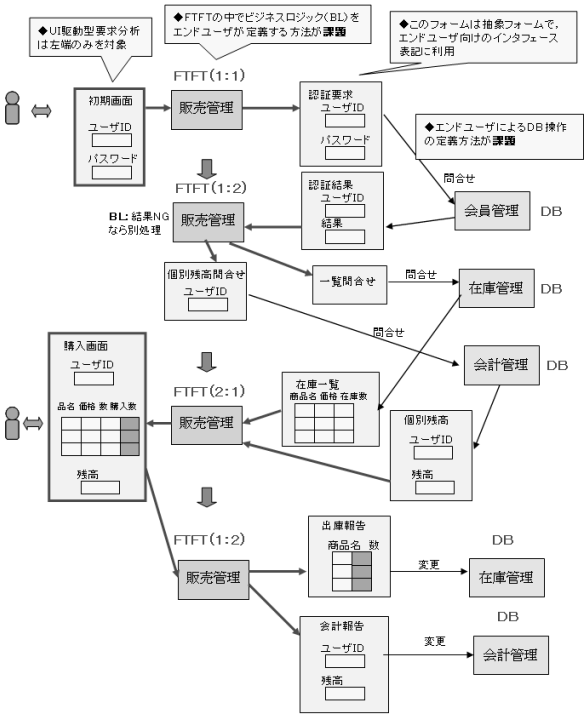

図 3: 例題アプリケーションのフォームフロー

初期画面でユーザＩＤとパスワードを入力して購入画面が表示されるまでの内部の処理フローは以下のようなものである。初期画面から抽象フォームである認証要求フォームへと変換（値を転記）する。このフォームを元に会員管理 DB への問い合わせを行い、結果を認証結果フォームとして得る。認証結果フォームから個別残高問合せフォームおよび一覧問合せフォームを出力する。会員管理 DB は個別残高問合せフォームから個別残高フォームを出力し、在庫管理 DB は一覧問合せフォームから在庫一覧フォームを出力する。これらの２つの出力フォームから購入画面フォームへと変換が行われる。

さらに、ユーザからの入力を得た後、購入画面から出庫報告フォーム（転記）と会計報告フォーム（計算）への変換を行い、それらのフォームを元に在庫管理 DB および会計管理 DB へアクセスを行う。

3.3.2　ツールによる変換手順定義

フォーム変換手順の定義を行うツールの使用例を図 4 に示す。ここでは、図 3 で示した購入画面から会計報告へとフォーム変換を行っている。右端は定義手順に対応した実行順を示す。

購入画面のユーザ ID を会計報告のユーザ ID に転記する処理は、『購入画面のユーザ ID』→『=』→『会計報告のユーザ ID』を順にクリックすることで定義する。購入画面の商品一覧の表から会計報告の残高を求める計算処理も同様に、マウス操作によって定義が可能である。

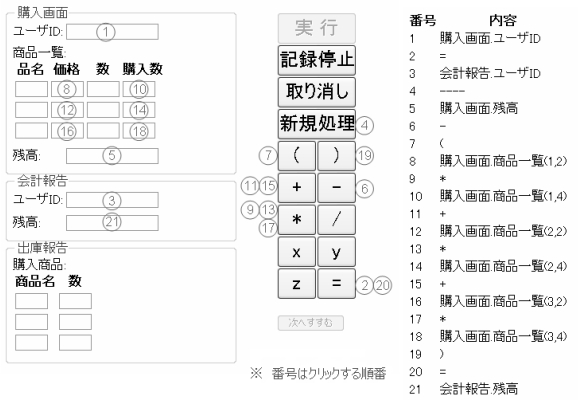

図 4: ツールと変換手順

3.3.3　変換手順テスト

さらに定義後すぐに動作確認できる機能を設けた。図 5 は、その確認画面を示す。購入画面の各項目をあらかじめ入力しておき、実行ボタンを押すことで、会計報告および出庫報告の各項目が定義に従って自動記入される。

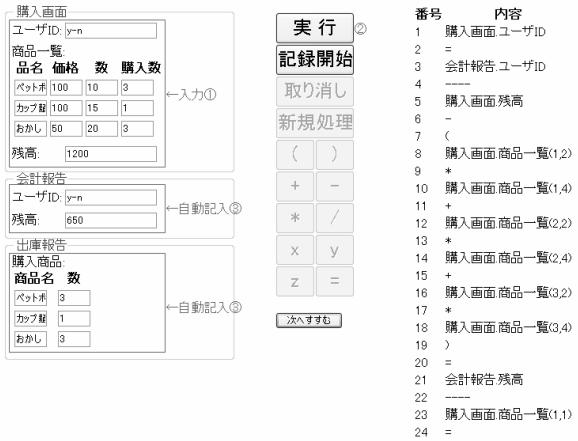

図 5: 変換結果確認画面

4.　ＤＢ操作用スクリプト言語

4.1　データの変化のみに注目

データベースの操作には主にデータの追加・変更・削除が挙げられる。データベースへの操作前後のデータの変化に注目すると、処理内容は単純なものとなる。例えば、図 3 の例では、図 6 に示すように残高の書き換え処理を行っていると見なすことができる。

本研究では Web サービス連携の観点で DB は XMLDB であることを前提とするので、このような処理を XSLT で記述する方法が考えられる。しかし、XSLT を用いた場合、データの追加・変更・削除といった処理の記述が複雑化してしまうので、エンドユーザにとって記述しにくくなってしまう問題点があった。

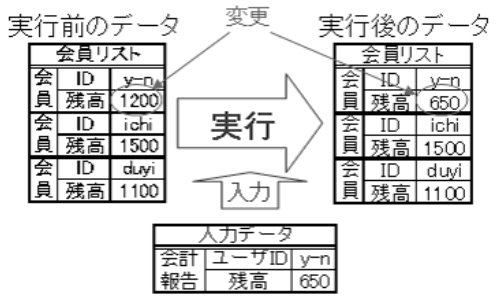

図 6: 会計管理 DB へのアクセス

```
<xsl:stylesheet version="1.0"
  xmlns:xsl=http://www.w3.org/1999/XSLT/Transform>
  <xsl:variable name="input" select="document('/入力データ.xml')"/>
  <xsl:variable name="db" select="document('/会計管理.xml')"/>
  <会員リスト>
    <xsl:for-each select="$db/会員リスト/会員">
      <会員>
        <ID>
          <xsl:value-of select="ID"/>
        </ID>
        <残高>
          <xsl:choose>
            <xsl:when test="ID = $input/会計報告/ユーザID">
              <xsl:value-of select="$input/会計報告/残高"/>
            </xsl:when>
            <xsl:otherwise>
              <value-of select="残高"/>
            </xsl:otherwise>
          </xsl:choose>
        </残高>
      </会員>
    </xsl:for-each>
  </会員リスト>
</xsl:stylesheet>
```

データの変更なし
そのまま出力

データの変更あり
場合分けを行って出力

図 7: XSLT による実装

4.2 XML データ操作言語 XDA

そこで、XML データの書き換えに独自のスクリプト言語 XDA Script を提案する。

XDA Script には基本的なデータを操作する API (insert,update,delete) を用意した。データの場所や、値を変更する際などはその場所を指定する必要があり、XPath を用いた。XPath には標準で用意されている関数 (sum など) があり、簡単な計算処理もできる。

XDA はデータの操作（選択・追加・変更・削除）の他に、if 文といった制御命令も持たせて、柔軟なロジックの記述も可能にした。

図 8 に XDA を用いた会計管理 DB へのアクセスの実装を示す。XSLT での記述と比較して、記述量が大幅に減っただけでなく、直感的なデータ操作が可能になり、可読性が高く、記述も容易であるといえる。

4.3 XDA による Web サービス呼び出し

さらに、Web サービス連携の Web アプリケーション構築のために、Web サービス呼び出しに対応する命令を用意した。

```
<xda:script>
  <xda:document name="input" src="/入力データ.xml"/>
  <xda:document name="db" src="/会計管理.xml"/>
  <xda:update where="$db/会員リスト/会員[ID=$input/会計報告/ユーザID]/残高"
              value="$input/会計報告/残高"/>
</xda:script>
```

データの変更

図 8: XDA による実装

例として、XML over HTTP の呼び出し例を図 9 に載せる。これは、図書登録時に AmazonWeb サービスから図書情報を取得する部分である。図の A は AmazonWeb サービスの呼び出しを行っており、B は取得した情報をフォームに埋め込み結果として返している。

なお、XDA の言語仕様を図 10 に示しておく。

```
<xda:script>
  A {
  <xda:xml-http name="BookInfo" end-point="http://ecs.amazonaws.jp/onca/xml">
    <xda:param name="Service">AWSECommerceService</xda:param>
    <xda:param name="Operation">ItemLookup</xda:param>
    <xda:param name="AWSAccessKeyId">1P7NHY34MAFWJ6HW7002</xda:param>
    <xda:param name="ItemId">
      <xda:value-of select="/Form[@name='図書検索']/Item[@name='ISBN']"/>
    </xda:param>
  </xda:xml-http>
  }
  B {
  <xda:response>
    <Form name="検索結果">
      <Item name="タイトル" type="text">
        <xda:value-of select="$BookInfo/ItemLookupResponse/Items/Item/ItemAttributes/Title"/>
      </Item>
      …
    </Form>
  </xda:response>
  }
</xda:script>
```

取得した情報の埋め込み

図 9: XDA による Web サービス呼び出し

4.4 フォーム変換との連携

次に XDA とフォーム変換との連携の例として、データ問合せとデータ書き換えの実装を示す。

図 11 は図 3 で示した例題アプリケーションの認証要求フォームから認証結果フォームを得る処理を XDA で記述したものである。最初の xda:document 〜から始まる命令で使用する XML ファイルを宣言する。ここで宣言した XML ファイルは name 属性で指定した名前で、以降 XPath 式の中で変数として使うことができる。ここでは、会員管理 DB に相当する会計管理 XML ファイルを指定している。次の xda:variable〜から始まる命令では、変数の宣言を行っている。xda:document〜命令と同様に name 属性で指定した名前を XPath 式内で変数として使用できる。変数の中身は xda:variable〜の子ノードとなる。ここでは、『ユーザ ID』という変数に認証要求フォームのユーザ ID を入れている。さらに『結果』という変数には、変数『ユーザ ID』と同一の ID を持つ会員のパスワードが認証要求のパスワードと同じなら”OK”、違うなら”NG”を入れている。最後の xda:response〜から始まる命令によって、レスポンスとなる認証結果フォームを作成している。認証結果フォームにはユーザ ID や結果に、先ほどの変数『ユーザ ID』『結果』の内容を xda:value-of 〜命令によって埋め込んでいる。

図 12 は同じく会計報告フォームから会計管理 DB の値を変更する処理を記述したものである。最初に

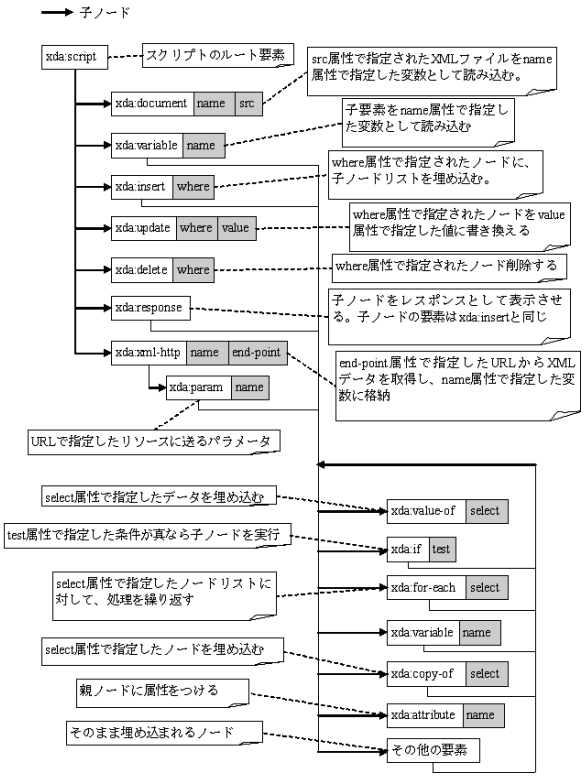

図 10: XDA の構文

xda:document 命令によって会計管理 DB に相当する会計管理 XML ファイルを指定している。次に、『ユーザ ID』という変数に会計報告フォームのユーザ ID の値を入れて宣言している。最後の xda:update 命令によって会計管理 DB の値を更新している。更新を行う場所は、where 属性に XPath を用いて指定する。ここでは、会員管理 DB にある会員リストから ID が『ユーザ ID』と同じ会員の残高を指定している。更新する値も同様に value 属性に XPath を用いて指定している。ここでは、会計報告フォームの残高を更新する値として指定している。

4.5 XDA Script における課題

XDA を習得するには、XML や XPath 等の構文と使用する XML データの構造、Web サービスを呼出すための情報といった知識が必要となる。

アプリケーションを構築するエンドユーザがこのような知識を持たなくてもいいように、ツールの作成などの支援策を考える必要があるだろう。

5. おわりに

エンドユーザ主導開発の研究の一環として、ビジネスロジック定義のためのフォーム変換ツールと DB 操作用スクリプト言語について述べた。

なお、実装技術については、フォーム変換手順の定義 UI は HTML + JavaScript によって実装した。変換手順は JavaScript によって XML 形式のデータとして、サーブレットに送り保存を行う。フォーム変換の手順実行は、XML 形式の変換手順ファイル

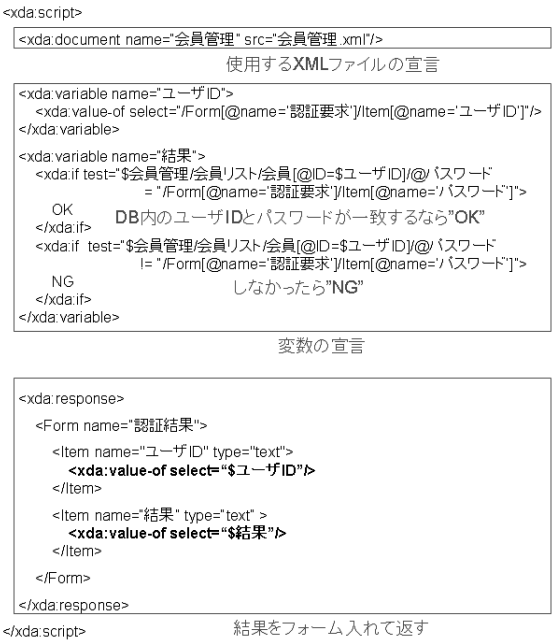

図 11: XDA による問合せ

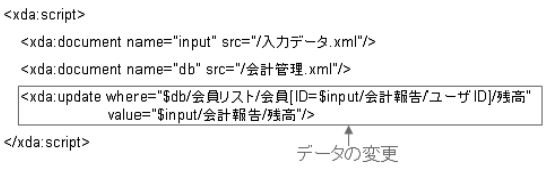

図 12: XDA によるデータ書き換え

に従って実行するようにし、インタープリタを Java により実装した。

XDA Script については、スクリプトコードを実行するインタープリタを Java により実装した。

参考文献

[1] 中所武司，絶えざる変化に対応するエンドユーザ主導のサービス連携，産学戦略的ソフトウェア研究フォーラム，ソフトウェアサービス技術シンポジウム資料集，8-1/2, 2001.

[2] Takeshi Chusho, et al., A Form-based Approach for Application Development By Web Service Integration, Applied Computing 2006, IADIS, pp.600-605, 2006.

[3] 八木紀幸，エンドユーザによる Web アプリケーション作成のためのスクリプト言語の作成と評価，明治大学理工学部情報科学科 2006 年度卒業論文.