

# フォームフローに基づくWebアプリケーション連携方式の研究

中 所 武 司 明治大学理工学部情報科学科教授

## 1 はじめに

近年、ビジネスの世界では、情報通信技術（ICT）の有効活用が不可欠となっている。それに伴い、ビジネスの形態は、情報通信技術の急激な進歩に呼応する形で、日々、変化している。このような状況に対処するために、インターネットをはじめとする情報通信技術を用いてビジネスモデルを具現化する Web アプリケーションの構築・運用や相互連携に関して、短期開発と継続的保守への要求が増大している。

しかしながら、従来のソフトウェア開発技法では、既存のパッケージやコンポーネントの再利用が不十分であり、アプリケーションの一部分または全体をプログラミング言語を用いて記述する必要があるため、それに見合った開発期間が必要とされてきた。さらに、このようなアプリケーションについては、運用後の機能変更に対しても、プログラミング言語を用いてソースコードを変更する作業が必要となるため、継続的保守をタイムリーに実施することが困難であった。

そこで、本研究では、業務の専門家に理解容易なフォームの概念を中心に置き、フォームの定義と、そのフォーム間の変換方法の定義によってシステムの仕様を記述する方法を研究開発し、業務の専門家主導での短期開発と継続的保守を実現することを目的とする。

## 2 変化に対応するアプローチ

このような問題を解決するためには、次のような二つの新しい考え方が必要である。

(a) ソフトウェアからサービスへの視点の移行：

情報社会が必要としているものは、ソフトウェアそのものではなく、ソフトウェアが提供する機能すなわちサービスをインターネットなどの情報通信技術を用いて授受することである。したがって、ソフトウェアの短期開発と継続的保守への要求は、言い換えれば、新しいサービスの迅速な実現と運用後の迅速な変更である。

(b) 情報処理専門家主導から業務専門家主導への開発主体の移行：

情報社会が必要としているサービスがどのようなものかを決定するのは、業務の専門家である。したがって、社会の変化に迅速に対応するためには、業務の専門家が情報処理の専門家に開発・保守を依頼するような従来の情報処理専門家主導の開発ではなく、業務の専門家が主導権を持って開発し、情報処理の専門家がそれを支援するような業務専門家主導の開発が必要である。

現在、ソフトウェアからサービスへの視点の移行に関しては、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が、実際の具体的な適用技術は従来の延長線のものが多い。また、これらは情報処理専門家主導の開発を前提としており、業務専門家主導の開発を支援するような技術を提供していない。

## 3 Webサービスへのシフト

### 3-1 エンドユーザ主導のシステム構築技術

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[2-4]。

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。すなわち、インターネットを介してやり取りするサービスのインタフェースのメタファーをフォームととらえるアプローチにより、ユーザインタフェース中心のフロントエンドサブシステムに用いられるフレームワーク主体のUI駆動型開発と、ワークフローとビジネスロジック中心のバックエンドサブシステムに用いられるビジュアルモデリング主体のモデル駆動型開発を統合する。特に、コンポーネント間の連携をメッセージフローで表現していた部分をフォームフローで置き換えたサービス連携とし、コンポーネントの機能をフォーム変換機能とする。このような開発技法により、業務の専門家主導のアプリケーション開発実現のための大きな課題であったコンポーネントの新規開発の問題をフォームベースのWebサービス連携で解決することができる。

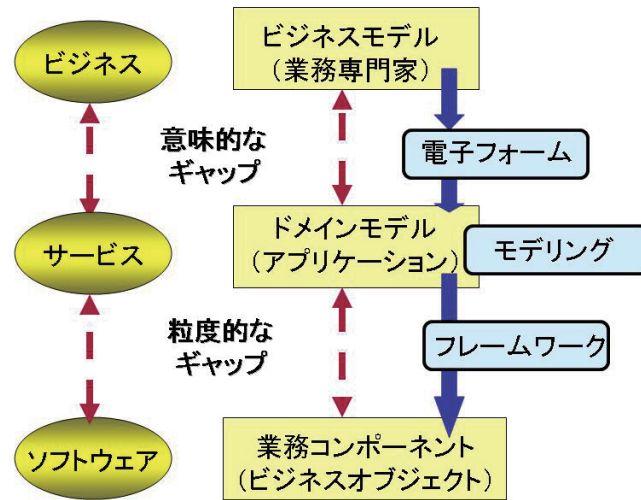


図1 エンドユーザ主導のシステム構築技術

### 3-2 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。

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、そのインタフェースをフォームとすることにより、ワークフローを以下のようにとらえる。

- ・業務コンポーネント → Web サービス、
- ・メッセージフロー → フォームフロー
- ・メッセージ変換 → フォーム変換

図2は、従来のワークフロー、図3はフォーム変換の一般形を示す。

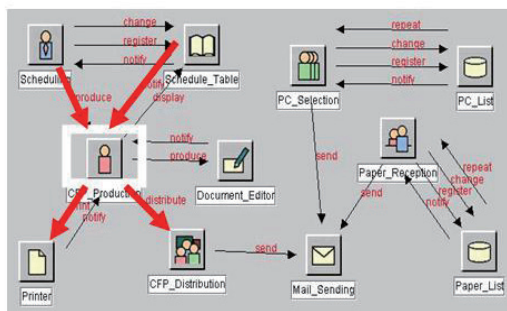


図2 ワークフローの例

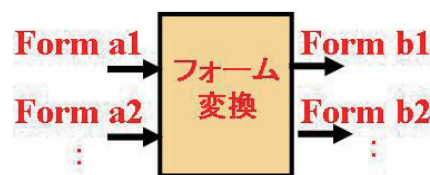


図3 フォーム変換

## 4 XSLT によるフォーム変換

### 4-1 基本変換

フォーム変換については、いくつかの方法が考えられるが、XML 形式の個人別履修情報と試験時間割表の2つの入力から個人別試験時間割表を出力する例題をとりあげ、XSLT[5]を用いる変換方法を研究試作した。すでに学内システム Oh-o!Meiji [6]からインターネット経由で HTML 形式で提供されている個人別時間割表を XML へ変換する処理を XSLT で記述するとともに、XML 形式で表現した試験時間割表と共通する科目情報（その学生の履修科目でかつ試験実施科目）を抽出して、XML 化する処理、すなわち、複数の XML 文書のマージ処理を XSLT で記述する方式の開発を通して、フォーム変換の定義を支援するビジュアルツールの必要性を明らかにした。

この XML マージ処理は以下のような処理になる。

1. 個人別時間割表 HTML から得られた履修状況 XML および試験時間割表 XML から比較対象となる要素（この例では科目要素）を一つずつ取り出す。
2. その要素内の比較したい要素（この例では科目名と教員名）の持つデータを比較し、同じ場合は、指定した形式に従ってデータを出力する。
3. 処理1,2を比較対象要素の全ての組み合わせについて行う。この処理を XSLT によって記述するために、外部の XML 文書へのアクセスを可能にする XPath の document 関数を用いて、ひとつの XSLT で複数の入力を取り扱えるようにした。また、ある親要素の直下の子である同じ要素の中でのノードの位置を返す position 関数を用いて、現在比較している要素の位置を変数に保持し、出力データの指定に利用するようにした。

図4に示すように、この XSLT を入力 XML へ適用することで XML マージ処理を行う。

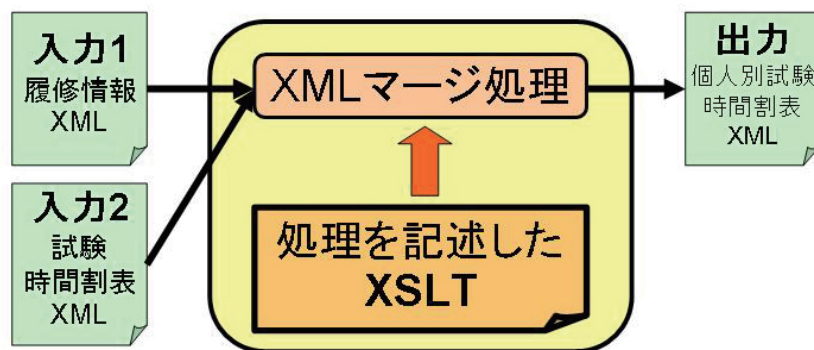


図4 XSLT によるフォーム変換の例

### 4-2 複雑なビジネスロジックのマージ処理方式

本研究では、より複雑なマージ処理を必要とする卒業条件判定システム[7,8]を考えた。これは、明治大学の学内システム Oh-o!Meiji から得られる、修得済み科目の名前と単位数と成績が記載された個人別成績表と、卒業までに必要な単位数や必修科目、選択必修科目等の条件が記載された卒業条件表の情報をマージし、卒業するために修得すべき単位の情報を各学生に提供するシステムである。

卒業条件判定システムは、入力XML から直接得られる一次情報だけでなく、入力XML の情報を組み合わせて加工した二次以上の情報も出力情報に含むという点で複雑である。例えば、必修科目の総修得単位数は、履修済み必修科目の単位数の合計なので、個人別成績表XMLから得られる科目の履修済み情報や単位数と、卒業条件XML から得られる必修科目か否かという一次情報の組み合わせから得られる二次情報である。

このような二次の情報が必要な場合、C 言語やJava などを用いたプログラムであれば、総修得単位数をあらわす変数を用意し、各科目を調べて、科目が必修かつ履修済みであれば、その単位数を変数に足しこむという方法が考えられる。しかし、XSLT では更新可能な変数がないため、このような方法が取れない。

そこで、回数毎に処理を区切ることを考える。すなわち、まず一次情報を取得して一次情報を含んだ中間出力XML を生成し、中間出力XML を用いて二次情報を得るようにする。一次情報を取得するXSLT は以下のような処理手順となる。

1. 入力となる2つのXML ファイルのノード集合を変数へ格納する。
2. 一次情報を取得し、出力XML に埋めこむ。

この一次の中間出力 XML には、2つの入力ファイルに含まれる情報に加えて、例えば、個人別成績表 XML に記載されている科目の履修済み情報などが新たに埋め込まれる。

二次以上の情報を取得する XSLT は以下のような処理手順となる。

1. 前の次数の中間出力 XML を読み込む。
2. 現在の次数の情報を取得して、出力 XML に埋めこむ。

例えば二次の場合、一次の中間出力 XML を読み込み、必修科目の総修得単位数を計算して二次の中間出力 XML に埋め込む。

このように、前の次数の情報を埋め込んだ中間出力 XML を作成することで、XSLT を用いて二次以上の情報を得ることができる。上記の処理手順を記述した XSLT を用意し、図5のように適用することで XML マージ処理を行う。

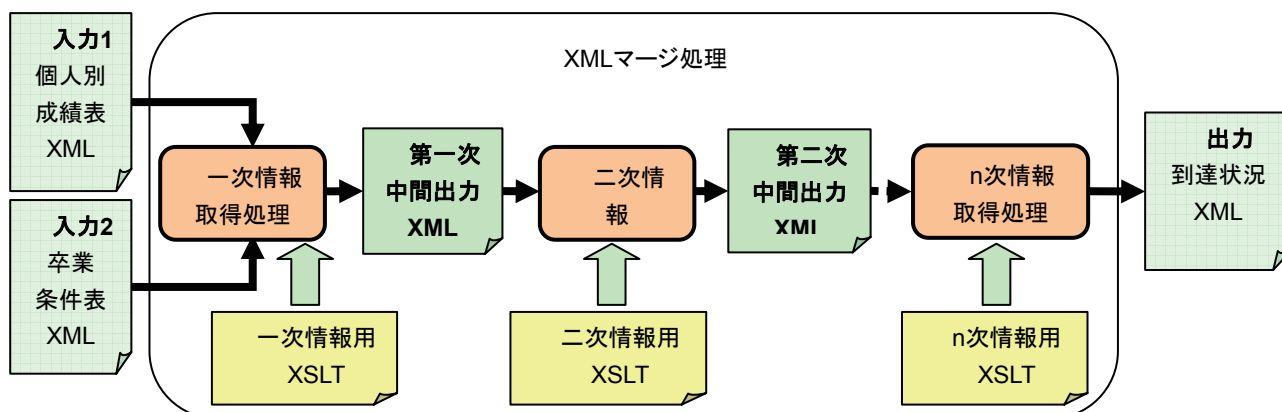


図5：卒業条件判定システムの XML マージ処理

試作した XSLT は、入力 XML のルート要素、科目区分、必修科目、選択必修科目の各要素に対応する、4つのテンプレートを含む。マージ処理で入力の要素を読むたびに対応するテンプレートが呼ばれ、情報の取得と出力作業を行う。例えば、二次情報用 XSLT の必修科目要素に対応したテンプレートは以下のようになる。

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output encoding="UTF-8" indent="yes" method="xml" version="1.0"/>
<xsl:template match="必修科目">
<xsl:element name="必修科目">
<xsl:attribute name="条件単位数">
<xsl:value-of select="current()/@条件単位数"/>
</xsl:attribute>
<xsl:attribute name="大小関係種別">
<xsl:value-of select="current()/@大小関係種別"/>
</xsl:attribute>
<xsl:attribute name="総修得単位数">
<xsl:value-of select="sum(current()/科目[@済='true']/@単位数)"/>
</xsl:attribute>
<xsl:attribute name="完全達成">
<xsl:value-of select="not(current()/科目[@済='false'])"/>
</xsl:attribute>
<xsl:for-each select="current()/科目">
<xsl:element name="科目">
<xsl:attribute name="id"><xsl:value-of select="current()/@id"/></xsl:attribute>
<xsl:attribute name="単位数">
<xsl:value-of select="current()/@単位数"/>
</xsl:attribute>
<xsl:attribute name="済"><xsl:value-of select="current()/@済"/></xsl:attribute>
```

```

</xsl:element>
</xsl:for-each>
</xsl:element>
</xsl:template>
</xsl:stylesheet>

```

必修科目は、条件単位数と、条件単位数に対する総修得単位数の関係を指定する大小関係種別と、総修得単位数と、完全達成の4つの属性を持ち、子に科目を持つ。科目は、科目名と単位数と履修済みか否かの3つの属性を持つ。まず、一次情報の条件単位数と大小関係種別をそのまま出力している。次に総修得単位数は、必修科目に含まれる済属性が true である科目の単位数の総和として得られ、以下のようなXPath式で表される。

```
sum(current()/科目[@済='true']/@単位数)
```

式中の sum 関数は引数の総和を計算する関数で、current 関数は現在の要素(ここでは必修科目)を得る関数である。また“科目[@済='true']”が、済属性が true である科目を表している。この式によって表される値を xsl:value-of を用いて出力している。

次に完全達成は、必修科目に含まれる済属性が false の科目要素が存在しなければ true, 存在すれば false として得られ、以下のようなXPath式で表される。この値を xsl:value-of で出力している。式中の not 関数は引数の論理否定を返す関数である。

```
not(current()/科目[@済='false'])
```

次に xsl:for-each によって必修科目に含まれるすべての科目要素の情報を取得して出力する。情報はいずれも一次なので、そのまま出力している。

### 4-3 XSLT 記述支援ツール

XSLT の全てをエンドユーザが記述するのは困難なので、XSLT を自動生成するツールを開発した。ツールへの入力は以下のようなものである。

- ・ マージ処理の入力 XML の構造 (2 ファイル分)
- ・ 出力 XML の構造
- ・ 出力 XML の各要素・属性の情報
- ・ 自動生成した XSLT の保存先

実際の処理を図6に示す。

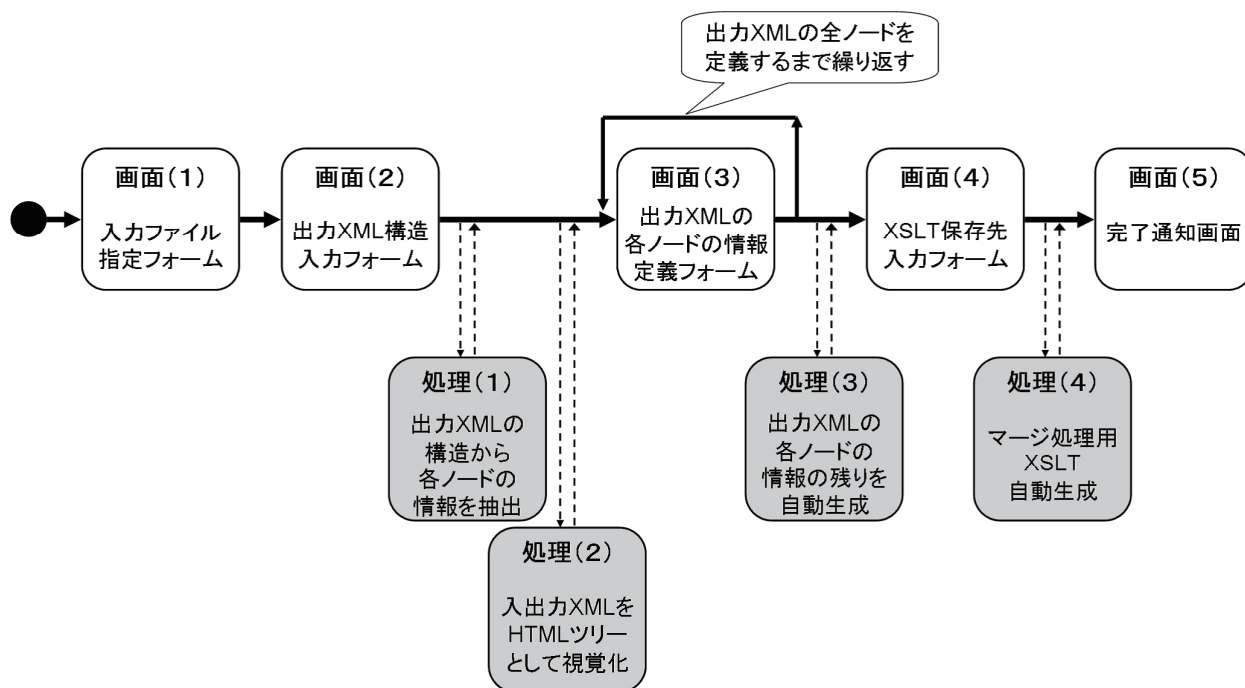


図6 XSLT 記述支援ツールの主要な画面遷移と処理フロー

ここでは、画面（3）における出力XMLの各ノードの情報定義について説明する。例題に登場する処理は全てIF-THENルールとして表すことができた。IF部については、分岐条件を記述するが、単純な式か、複数の条件のANDをとるものがあった。THEN部については、以下の処理が必要であった。

- ・入力情報の埋め込み
- ・値の合計の計算
- ・ブール値(真偽)の出力

実際の画面を図7に示す。

マージ処理用XSLT記述支援ツール: Step 3/4

### ノードの定義

Hint: ページ下部に使い方が載っています。

---

#### 情報入力欄

/到達状況/科目区分/必修科目/@総修得単位数の情報定義

出現条件

以下の要素に対応して繰り返す:  追加

---

マージ処理IF-THENルール

IF-THENルールの入力

IF:

THEN:  IF=THENを追加

ELSE:  当該要素から値を得る

---

入力確定

入力 キャンセル

---

#### XPath記述支援ツール

図7 出力XMLの各ノードの情報定義のフォーム

XPathの記述が必要な場合は、図8のようなチェックボックス付きの出力XML構造を表示して、チェックしたノードのXPathを生成する。

#### 入出力XMLの要素・属性一覧

- 到達状況:
  - 年度:
  - 総修得単位数:
  - 条件単位数:
  - 大小関係種別:
  - 完全達成:
  - 科目区分:
    - 区分名:
    - 総修得単位数:
    - 条件単位数:
    - 大小関係種別:
    - 完全達成:
    - 科目:
    - id:
    - 単位数:
    - 済:
  - 選択必修科目:
    - 総修得単位数:
    - 条件単位数:
    - 大小関係種別:
    - 完全達成:
    - 科目:
    - id:
    - 単位数:
    - 済:
  - 科目:
    - id:
    - 単位数:
    - 済:
  - 副科目区分:

- 卒業条件:
  - 年度: 2001
  - 条件単位数: 132.0
  - 大小関係種別: 以上
  - 科目区分:
    - 区分名: 総合文化科目
    - 条件単位数: 8.0
    - 大小関係種別: 以上
  - 科目区分:
    - 区分名: 健康・スポーツ学
    - 条件単位数: 2.0
    - 大小関係種別: 以上
    - 必修科目:
      - 条件単位数: 2.0
      - 大小関係種別: 丁度
      - 科目:
        - id: 健康・スポ-
        - 単位数: 1.0
      - 科目:
        - id: 健康・スポ-
        - 単位数: 1.0

- 履修状況:
  - 総修得単位数: 135.0
  - 科目区分:
    - 区分名: 総合文化科目
    - 総修得単位数: 8.0
    - 科目:
      - id: 自然科学史B
      - 単位数: 2.0
    - 科目:
      - id: 心理学A
      - 単位数: 2.0
    - 科目:
      - id: 現代政治論A
      - 単位数: 2.0
    - 科目:
      - id: 運動の科学A
      - 単位数: 2.0
  - 科目区分:
    - 区分名: 健康・スポーツ学
    - 総修得単位数: 2.0
    - 科目:
      - id: 健康・スポーツ学
      - 単位数: 1.0
    - 科目:
      - id: 健康・スポーツ学
      - 単位数: 1.0
  - 科目区分:
    - 区分名: 第一外国語
    - 総修得単位数: 14.0
    - 副科目区分:
      - 区分名: 第一外国語
      - 条件単位数: 8.0
      - 大小関係種別: 丁度
      - 科目:
        - id: 英語
        - 単位数:

図8 チェックボックス付きXML構造表示例



## 5 操作手順の定義による変換

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[9]が考えられる。図9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。

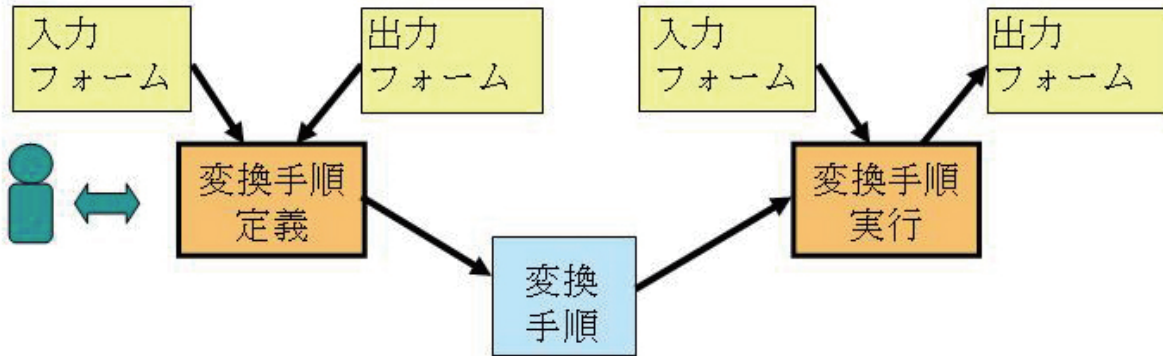


図9 フォーム変換手順の定義と実行

## 6 おわりに

エンドユーザ主導開発の研究の一環として、フォーム変換定義による要求記述について述べた。類似の技術としては、Programming by example (PBE) [10]がある。これは、操作例からユーザの意図を推測し、類似の状況下での操作を自動化するものである。また、データベースの問い合わせに関して、Query by example (QBE) [11]がある。これも、SQL のような問い合わせ言語を使わずに、具体例を用いて問い合わせ内容を指定するものである。本論文の方式は、例示による推論ではなく、入出力フォーム間の変換手順を厳密に定義するものである。

### 【参考文献】

- [1] 中所, 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] Brown, A. W., (Ed.), Component-based software engineering, IEEE CS Press, 1996
- [3] Fayad, M. and Schmidt, D. C. (Ed.), "Object-Oriented Application Frameworks," Commun.ACM, V39, N10, pp. 32-87, 1997.
- [4] Gamma, E., Helm, R., Johnson, R. and Vlissides, J., Design Patterns, Addison-Wesley, 1995.
- [5] W3C, "The Extensible Stylesheet Language Family(XSL)," <http://www.w3.org/Style/XSL/>, 2006.
- [6] 明治大学, Oh-o!Meiji, <http://oh-o.meiji.ac.jp/>
- [7] 西田晋平, エンドユーザ主導開発を志向した Web サービス連携における XML マージ処理方式の提案と評価, 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 2006 年度修士学位請求論文.
- [8] Takeshi Chusho, Ryousuke Yuasa, Shinpei Nishida, and Katsuya Fujiwara, Web Service Integration Based on Abstract Forms in XML for End-user Initiative Development, Proc. The 2007 IAENG International Conference on Internet Computing and Web Services (ICICWS'07), pp.950-957 (Mar. 2007).
- [9] 中所武司, エンドユーザ主導の要求記述のためのフォーム変換定義法, 情報処理学会 ウィンターワークショップ 2007・イン・那覇 論文集, pp.49-50 (Jan. 2007).
- [10] Lieberman, H. (Ed.), "Special issue on Programming by example," Comm. ACM, V43, N3, pp.72-114, 2000.
- [11] Ozsoyoglu, G. and Wang, H., "Example-Based Graphical Database Query Languages," IEEE Computer, V26, N5, pp.25-38, 1993.

〈発 表 資 料〉

題 名	掲載誌・学会名等	発表年月
エンドユーザ主導開発のためのフォームベース要求定義法	情報処理学会 要求工学WG 第25回要求工学ワークショップ	2007年5月
Web Service Integration Based on Abstract Forms in XML for End-user Initiative Development	The 2007 IAENG International Conference on Internet Computing and Web Services (ICICWS'07)	2007年3月
エンドユーザ主導の要求記述のためのフォーム変換定義法	情報処理学会 ウィンターワークショップ 2007・イン・那覇 論文集	2007年1月
エンドユーザ主導の要求記述におけるフォーム変換定義法	情報処理学会 要求工学WG 第23回要求工学ワークショップ	2006年10月
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	第6回情報科学技術フォーラム (FIT2007)	2007年9月 (予定)