

# 業務の知識を有するエンドユーザ主導のアプリケーション開発技法 —フレームワーク・ドメインモデル・サービス連携—

中所 武司<sup>†</sup>

<sup>†</sup> 明治大学理工学部情報科学科 〒214-0033 神奈川県川崎市多摩区東三田 1-1-1

E-mail: <sup>†</sup> chusho@cs.meiji.ac.jp

**あらまし** 変化の激しい時代には、業務の知識を有する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。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必要性が高まると考え、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。具体的には、アプリケーションとコンポーネントの間の粒度的なギャップは、フレームワークやビジュアルモデリングツールで解決するとともに、ビジネスモデルとアプリケーションの間の意味的なギャップは、フォームベースのサービス連携で解決するアプローチをとってきた。本論文では、その研究成果と課題について述べる。

**キーワード** エンドユーザコンピューティング, フレームワーク, ビジュアルモデリング, Web サービス

## Enduser-Initiative Application Development Methods with Business Knowledge — Framework, Domain Model, Service Integration —

Takeshi CHUSHO<sup>†</sup>

<sup>†</sup> Department of Computer Science 1-1-1 Higashimita, Tama-ku, Kawasaki, 214-0033 Japan

E-mail: <sup>†</sup> chusho@cs.meiji.ac.jp

**Abstract** Enduser-initiative application development and modification become important because business models are changing rapidly. Especially, Web application for small groups or individuals should be developed in short term with low cost and be maintained frequently with functional specification change. The granularity gap between applications and software components is bridged by frameworks and visual modeling tools. The semantic gap between business models and applications is bridged by form-based Web service integration.

**Keyword** End-User Computing, Framework, Visual Modeling, Web Service

### 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、10年来の研究を行ってきた。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も対応するために、その分野の業務の専門家主導で開発・保守ができるような技法を研究してきた [11].

### 2. 研究の背景

#### 2.1. IT による豊かな生活の実現

情報システムは、従来、情報処理の専門家が開発し、限られた人たちが利用してきた。しかし、90年代以降、パソコン、ネットワーク、インターネットの普及とともに、オフィスの内外でエンドユーザが増加し、情報化の第一義的な目的は、すでに業務の効率化といった従来の枠組みを超えていると考えられた。

そこで、我々は、文献 [5] において、ITによる豊かな生活の実現を意味する”CS-life”(Computer-Supported Life)を研究目標として掲げた。文献 [6] では、生産者中心の視点から利用者中心の視点への転換の必要性を唱え、仕事の効率化よりも生活を豊かにすることにもっと知恵を絞るべきと述べた。

このような考えは、その後、内閣のIT戦略本部が発表した2001年のe-Japan戦略でも「すべての国民が情報通信技術（IT）を積極的に活用し、かつその恩恵を最大限に享受できる知識創発型社会の実現」として、また、2006年のIT新改革戦略においては、「いつでも、どこでも、誰でもITの恩恵を実感できる社会の実現」として掲げられている。しかしながら、そのなかで「使い勝手が利用者の視点に立ったものとなっていない等の理由から、国民・企業等による電子政府の利用は進んでおらず、また、住民サービスに直結する地方公共団体の電子化が十分ではないなど、国民・企業等利用者が利便性・サービスの向上を実感できていない」と反省が述べられている。[17]

## 2.2. エンドユーザ主導開発の位置づけ

その実現のための一つのシナリオとして、全ての日常の仕事はコンピュータが代行すべきであるという立場で、エンドユーザが自分のエージェントを自ら作り、自ら利用するためのツールを開発することによって、オフィスにおいては、創造的な仕事により多くの時間をかけるとともに、家庭においては、余暇を楽しむ時間を増やす効果を期待した[5]。

## 3. 従来の技術

### 3.1. エンドユーザコンピューティング

エンドユーザコンピューティング（EUC）という言葉は80年代によく目にするようになったが、エンドユーザ及びエンドユーザコンピューティングの定義と分類については文献[12]に詳しい。また文献[3]では、エンドユーザコンピューティングの管理に関する論文90件以上を調査分析しているが、技術的な内容は少ない。

筆者の解説論文「エンドユーザコンピューティング」[4]では、業務の知識を有するエンドユーザ、すなわち業務の専門家が自らの業務を自動化するソフトウェアを自ら開発する、という意味で用いている。ここでは、ソフトウェア産業の進化過程を図1のように表した。

ソフト産業の形態	主要な技術職	主要技術
労働集約型産業	プログラマ	自動化 (CASE)
知識集約型産業	設計者	標準化 (パッケージ)
知恵集約型産業	業務専門家	エンドユーザコンピューティング

図1 ソフトウェア産業の進化過程 ([4] の表2)

プログラムをすべて手作りする時代から業務用パッケージを利用する時代を経て、業務の専門家が知恵を絞ってビジネスモデルを構築する時代になれば、E

UCが主要技術になると考えられる。

また、プログラミング言語の発展過程を図2のように表した。

時期	目的	内容
1960年代	量的高級化	記述水準（機械語への展開率）の向上
1970年代	質的高级化	プログラミング方法論（構造化技法）の導入
1980年代	パラダイム転換	宣言的記述（手続き型から非手続き型へ）
1990年代	エンドユーザコンピューティング	脱プログラミング

図2 プログラミング言語の発展過程 ([4] の表3)

業務の専門家がアプリケーション開発を主導するためには、プログラミングの概念を必要としないソフトウェア構築技術が必須である。

### 3.2. 脱プログラミング

ここでは簡単に脱プログラミングに関連すると思われる従来技術の一部を紹介する。

1989年4月の情報処理学会の特集号「新しいプログラミング環境」の視覚的プログラミング環境[19]の解説論文では、ソフトウェア設計やプログラム作成のために直接的に2次元の視覚的表記を用いる視覚的言語をその図的表現形式の違いにより、グラフ指向言語、フォーム指向言語、アイコン指向言語に分類している。支援対象に関しては、データの視覚化、プログラムとその実行の視覚化、上流工程の設計の成果物の視覚化、プログラミング工程の視覚化が紹介されている。最後のプログラミング工程の視覚化が脱プログラミングの概念に近い。

先に述べた1991年の文献[4]では、エンドユーザがプログラミング言語を用いなくて業務のコンピュータ化を行なえるための現実的な技術として、4GL(第4世代言語)、日本語プログラミング、ビジュアルプログラミング、AI(ルール、ファジー、ニューロ)の4項目を取り上げた。ただし、いずれも脱プログラミングの概念に基づいているが、実際には脱プログラミング言語のレベルであり、プログラミングの概念やセンスが不要というわけではない。

4GLの特徴については、James Martinが、非職業的プログラマが使用可能であること、非手続的記述形式であること、記述量と作成工数がCOBOLより1桁少ないことなどを挙げている。

日本語プログラミングについては、業務用語辞書作成保守機能やプログラミング言語の知識を不要とする開発保守環境に加えて、日本語表現が持つ本来のあいまい性を避けながら、自然な表現を保持した文法規則の導入などが課題である。

ビジュアルプログラミングについては、先に述べた視覚的言語の図的表現形式について言及した。

AIについては、推論機能ではなく、簡易プログラミングという観点から、エキスパートシステム構築ツールについて言及した。

著者が編集に参加した1998年の単行本「コンポーネントウェア」[2]では、完成度の高い部品の組み合わせによるシステム構築ツールとして、APPGALLERY, HOLON/VP, IntelligentPad, VisualAgeなどが紹介されている。

2000年3月のCACMのProgramming by example特集[18]では、操作例からユーザの意図を推測し、類似の操作を自動化するシステムがいくつか紹介されている。また、データベースの問い合わせに関して、Query by example (QBE) [21]がある。これも、SQLのような問い合わせ言語を使わないで、具体例を用いて問い合わせ内容を指定するものである。

2004年9月のCACMのEnd-user development特集[22]では、IT技術者の支援なしで、ユーザ自身の問題解決ソフトを自ら開発する技術の研究動向をまとめている。

エンドユーザ主導開発の事例としては、長崎県庁で職員が画面設計と業務要件を決定後、詳細仕様書に基づき入札を実施し、業者に発注する例などがある[15]。

### 3.3. 研究アプローチ

著者が最初にエンドユーザコンピューティング技術としてとりあげたのは、80年代後半の知識ベースシステムである。文献[5]の研究経過でも述べたことであるが、「エンドユーザが頭の中に作成したプログラムを自らコンピュータ化するためには、その業務モデルあるいは計算モデルをそのまま表現し、入力できるようにする必要がある。そのためには、マルチパラダイム化は避けられないが、実際には、「多機能、即、複雑」という技術的課題を解決しなければならない」という考えから、知的電子秘書システムを開発した経験がある。しかし、残念ながら、エンドユーザが利用するには複雑すぎるものになってしまった。

90年代には、コンポーネントベースのエンドユーザ主導型アプリケーション開発技法として、特定分野のアプリケーションの枠組みをフレームワークとしてあらかじめ開発し、個々のアプリケーション固有の処理部分のみを後から追加するというUI駆動型の開発技法と、ビジュアルツールと業務コンポーネント(ビジネスオブジェクト)を用いて業務モデルを構築し、その上でのシミュレーション検証を繰り返しながらアプリケーションを開発するモデル駆動型の技法を研究してきた。最近では、Webサービス連携技法を用いてこれらを統合したフォーム駆動型開発技法を研究して

いる。以下では、これらについて述べる。

## 4. フレームワークを用いた開発方法

### 4.1. 基本方針

アプリケーションフレームワーク[9]は、ある程度アプリケーションの分野を特定することにより、そのソフトウェアアーキテクチャの基本的な枠組みと、その枠組みを構成する基本的なクラスライブラリを備えたものである。個々のアプリケーションは、そのフレームワークをカスタマイズして作成することになるので、短期開発と低コストと高品質を実現する。

カスタマイズのレベルとしては、図3に示すように、コンポーネントの属性値の設定や既存のコンポーネント群から必要なものを選択するだけの簡単なものから、部品として用意されたクラスにソースコードを書き加えるものや、新たに追加すべきクラスを開発するものもある。カスタマイズのレベルによって開発者に要求されるスキルが異なるが、既にソフトウェアのアーキテクチャとそこで必要とされるクラス群におけるクラス間の静的構造が決定しているため、設計作業は大幅に削減できる。

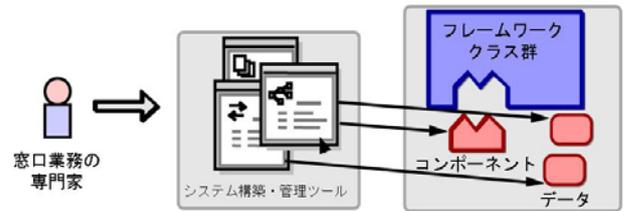


図3 フレームワークを用いた開発

アプリケーションフレームワークの適用分野に関しては、特定業務に特化したものと、業務は特定しないがユーザインタフェースの処理が主になるフロントエンド系(Struts[1]など)やデータベース処理が中心のバックエンド系(Hibernate[14]など)のように、特定の情報処理内容に特化したものがある。特定業務に特化したものは、既に業務の知識が業務コンポーネントとして用意されているので、カスタマイズは比較的容易である。特定の情報処理内容に特化したものは、アプリケーション固有の業務処理(業務ロジック)を新たに付加する必要があるが、業務に横断的に適用可能である。

エンドユーザ主導開発では、業務の専門家が主体なので、特定業務に特化したフレームワークに着目し、その中でも汎用性が高いと思われる窓口業務を取り上げて研究してきた。文献[5]で述べたように、「これらの窓口業務の担当者も毎日同じような質問を受け、<中略>ルーチンワーク化した判断処理をしている。



この方式の課題の1つは、複雑な業務ロジックの記述方法であるが、図4はM-baseを用いて国際会議のプログラム委員長の業務を開発する例である。ドメインモデルの構築は、UMLのコラボレーション図(UML2.0のコミュニケーション図)に似たメッセージフロー図(図4の①)の記述から始め、その各要素の詳細を専用のエディタ(図4の②)を用いて定義して完成させる。そのとき、フローで表現できない業務ロジックをルール形式で定義できるようにした。図4の③は自動生成されたUI画面、図4の④はシミュレーション実行による検証の様子である。

もう1つの課題は、必要な業務コンポーネントが存在しない場合である。現状では、IT技術の専門家に外注せざるを得ないが、要求仕様はエンドユーザが定義すべきものである。その実現方式として、業務コンポーネントをWebサービスとみなすアプローチをとる。この場合、メッセージをフォームで表現することにより、先に述べたUI駆動型の窓口業務フレームワークと同じレベルで扱うことができる。

## 6. Webサービス連携を用いる開発方法

### 6.1. 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5に示す[8]。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。

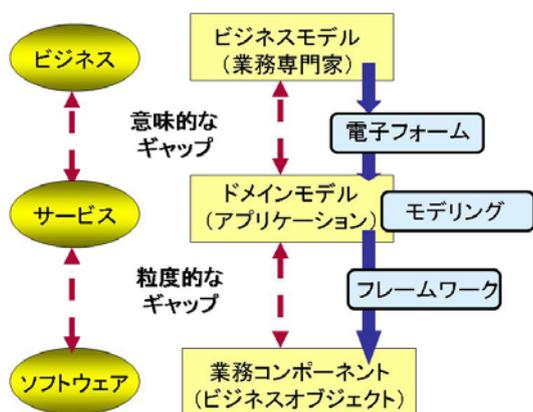


図5 エンドユーザ主導のシステム構築技術

### 6.2. 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、M-baseで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する。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6は、従来のワークフロー、図7はフォーム変換の一般形を示す。

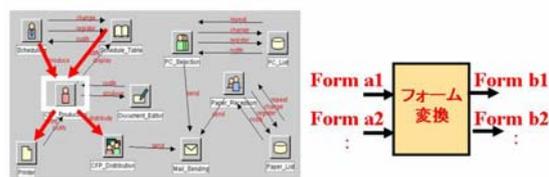


図6 ワークフローの例 図7 フォーム変換

### 6.3. フォーム変換

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、XSLTを用いる方法を研究試作した[20]。図8は、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である。この方法では、エンドユーザ向けにXSLT記述を支援するビジュアルツールを開発した。

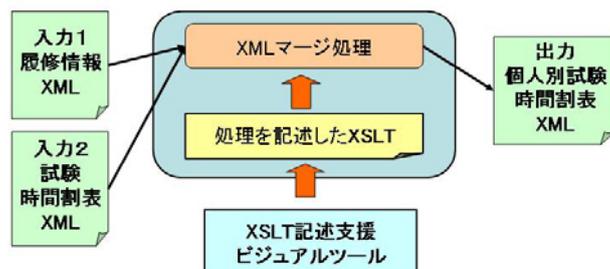


図8 XSLTによるフォーム変換の例

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 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 [23]。

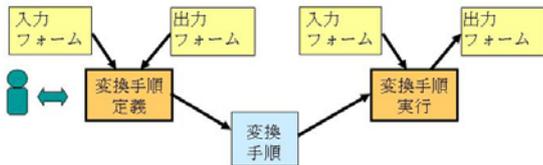


図 9 フォーム変換手順の定義と実行

図 10 は、商品販売システムのフォームフローの一部である。エンドユーザ主導開発の特徴として、ユーザインタフェースの必要のない内部のフロー部分もビジュアルなフォーム（抽象フォーム）を用いている。

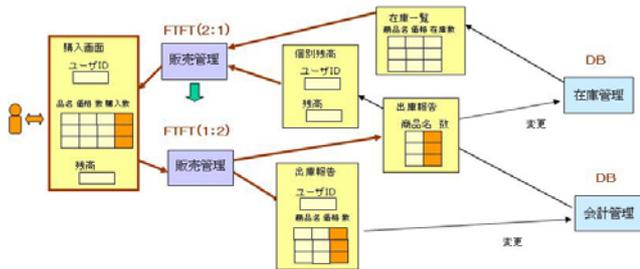


図 10 フォーム変換とフォームフローの例

## 7. おわりに

以上、業務の知識を有するエンドユーザ（業務の専門家）が、自らの業務をコンピュータ化する技法について述べた。主に 80 年代からエンドユーザコンピューティングとして注目されてきたこの分野の従来技術を概観するとともに、我々がエンドユーザ主導開発の基本技術として注目した、アプリケーションフレームワーク、ビジュアルモデリング、Web サービス連携に関する研究状況について報告した。

## 文 献

[1] Apache Jakarta Project, “Struts,” <http://struts.apache.org/>.

[2] 青山幹雄, 中所武司, 向山博 (編), コンポーネントウェア, 共立出版, 東京, 1998.

[3] James C. Brancheau and Carol V. Brown, The management of end-user computing: status and directions, ACM Computing Surveys, vol.25, no.4, pp. 437 – 482, Dec. 1993.

[4] 中所武司, “エンドユーザコンピューティング ーソフトウェア危機回避のシナリオー,” 情報処理, vol.32, no.8, pp.950-960, Aug.1991.

[5] 中所武司, “wwHww : 分散オフィスシステムのためのエンドユーザコンピューティング向きオブ

ジェクト指向モデル,” 情報処理学会ソフトウェア工学研究会資料, 94-SE-97, 97-5, pp.33-40, Mar.1994.

[6] 中所武司, “CS-life,” コンピュータソフトウェア(日本ソフトウェア科学会), vol.11, no.6, pp.1-2, Nov. 1994.

[7] 中所武司, “M-base : 「ドメインモデル≡計算モデル」を志向したアプリケーションソフトウェア開発環境の基本概念,” 情報処理学会ソフトウェア工学研究会資料, 95-SE-104, 104-4, pp.25-32, May.1995.

[8] 中所武司, 藤原克哉, 石樽久嗣, 島田圭, “絶えざる変化に対応するエンドユーザ主導型アプリケーション開発技法ーCS-life の実現をめざしてー,” 情報処理学会 第 62 回全国大会 特別トラック(4)「IT 革命を支えるソフトウェア開発技術」講演論文集, 6H-01, pp.87-92, Mar. 2001.

[9] 中所武司, ソフトウェア工学 (第 2 版), 朝倉書店, 東京, 2004.

[10] 中所武司, 津久井浩, “予約業務を例題とした Web アプリケーション用フレームワークの再利用性の評価,” 電子情報通信学会 和文論文誌 D-I 分冊, vol.J88-D-I, no.5, pp.930-939, May 2005.

[11] 中所武司, <http://www.chusho.jp/>.

[12] W. W. Cotterman, K. Kumar, User cube: a taxonomy of end users, Communications of the ACM, vol.32, no.11, pp. 1313-1320, Nov. 1989.

[13] 藤原克哉, 中所武司, “窓口業務を例題としたエンドユーザ向き分散アプリケーションフレームワーク wwHww の開発と適用評価,” 情報処理学会論文誌, vol.41, no.4, pp.1202-1211, Apr. 2000.

[14] Hibernate, <http://www.hibernate.org/>.

[15] 広岡延隆, “職員が自ら詳細仕様を作成し, IT 調達コストを大幅ダウン,” 日経コンピュータ 05.7.25 号, July 2005.

[16] 石樽久嗣, 紺田直幸, 中所武司, “メッセージフローモデルに基づくエンドユーザ主導型アプリケーション構築・検証技法,” 情報処理学会 オブジェクト指向 '2000 シンポジウム論文集, pp.133-140, Sep. 2000.

[17] IT 戦略本部, “IT 新改革戦略,” <http://www.kantei.go.jp/jp/singi/it2/>.

[18] H. Lieberman(Guest Ed.), Programming by example, Communications of the ACM, vol.43, no.3, pp. 72-74, Mar. 2000.

[19] 西川博昭, 寺田浩詔, “視覚的プログラミング環境,” 情報処理, vol.30, no.4, pp.354-362, Apr.1989.

[20] 西田晋平, 中所武司, “Web サービス連携のための XML マージ処理方式の実験と評価,” FIT2005 第 4 回情報科学技術フォーラム, 第一分冊, pp.129-130, Sep. 2005.

[21] Gultekin Ozsoyoglu and Huaqing Wang, Example-Based Graphical Database Query Languages, IEEE Computer, vol.26, no.5, pp. 25-38, May 1993.

[22] A. Sutcliffe and N. Mehandjiev(Guest Ed.), End-user development, Communications of the ACM, vol.47, no.9, pp. 31-32, Sep. 2004.

[23] 八木紀幸, 中所武司, “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,” FIT2007 第 6 回情報科学技術フォーラム, B-016, 113-116, Sep. 2007.