

# 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 CS-lifeの実現をめざして

中 所 武 司      藤 原 克 哉  
石 樽 久 嗣      島 田      圭

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

## 1. はじめに

近年、インターネットやイントラネットに接続されたパソコンの普及と共に、オフィス業務の効率化という観点から、業務の専門家が自ら情報システムを構築する必要性が高まっている。たとえば、基幹業務システムのように投資に対する効果が明確なものは、従来のように情報処理の専門家が開発すればよいが、小さな部門あるいは個人の業務を対象とするものや頻繁に機能変更が発生するものは従来の開発方法はなじまない。

最近では、電子商取引に代表される web アプリケーションや EAI(Enterprise Application Integration) によって大規模化するスーパーアプリケーションの中にも頻繁な機能変更を伴うものが増加している。このような絶えざる変化に対応することが求められるアプリケーションは、業務の専門家が自ら開発し、自ら保守を行なうのが望ましい。

このような新しい動向に対応して、短期間でタイムリーにアプリケーションを開発するために 90 年代半ばからコンポーネントウェア<sup>1)</sup> やビジュアルプログラミングに代表されるような新しい開発ツールが実用化されはじめている。従来の構造化技法のようなトップダウン技法からコンポーネントの組み合わせ(モデリング)というボトムアップ技法へ移行しつつある。オープンなアーキテクチャとアプリケーションフレームワーク、デザインパターン、コンポーネントなどの構成要素から、ビジュアルツールを用いてアプリケーションを再帰的に構築していく技法が追求されている。

われわれは、これまで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、以下の研究を行ってきた。

- wwHww: フレームワーク&エージェント技法
  - M-base: モデリング&シミュレーション技法
- wwHww<sup>2),7),8),12)</sup> は、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開発技法である。そのアプリケーション固有の部分にエージェント技術を適用することにより、エンドユーザによるカスタマイズをさらに容易にする FACL(Form-based Agent Communication Language)<sup>5)</sup> を開発している。本技術は電子商取引や電子政府の分野にも適用可能と考えている。

M-base<sup>3),4),9)</sup> は、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技法である。

## 2. エンドユーザ主導の必要性

今からちょうど 7 年前にエンドユーザ主導型のアプリケーション開発技法を研究するために以下のような目標<sup>2)</sup>を設定した。

『(前略)。このような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越えている。情報化におけるエンドユーザも、従来の情報システム部門に対比したエンドユーザ部門という範囲を越えている。ここでは、このような情報化をコンピュータによる「より良い生活の実現」という意味で“CS-life”(Computer-Supported Life)と呼ぶ。人間とコンピュータとのかかわりあい方としては、次の 3 種類が考えられる。

- 人間の仕事を代行する。
- 人間の仕事の質を高める。
- 人間の遊びの質を高める。

ここでは、第 1 項に注目し、「すべての日常的仕事は

Enduser-Initiative Application Development for Frequent Changes

Takeshi Chusho, Katsuya Fujiwara, Hisashi Ishigure and Kei Shimada

Computer Science Course, Major in Sciences, Graduate School of Science and Technology, Meiji University

コンピュータが代行するべきである」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」(Information systems of the end-user, by the end-user, for the end-user)を開発することによって、次のような効果の実現をめざす。

- (1) 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。
- (2) 家庭においては、人それぞれの個性的な人生の楽しみにより多くの時間をかける。

( 中略 )

急速な高齢化社会の到来と若年労働者の減少への対応としても、これらの技術はおおいに役立つはずである。幸い、新社会資本の充実の一環として、コンピュータネットワークのインフラ整備に追い風が吹いている。(引用終り)

このような状況はIT革命という言葉に代表される新世紀初頭においてさらに加速されているが、実際の実現レベルはまだまだ不十分である。

例えば7年前と同じ例として、引越しの時の住所変更連絡作業についてみてみよう。1年余り前の著者自身の経験によると、住所変更連絡が53件必要であったが、その内容の伝達方法は以下の通りであった。

- 書類記入：43件(電子フォーム3件含む)
- 口頭伝達：18件(うち8件は電話で取り寄せ後の書類記入と重複)

伝達手段に関しては以下の通りであった。

- 郵送：23件(FAX：1件を含む)
- 電話：10件
- 訪問：8件
- インターネット：3件
- その他：9件

その他は相手先が自宅に来たときに渡したものが1件、転居時には連絡せず、別の機会に連絡したものが8件であった。最近、インターネット化の状況を少し確認したがあまり変化はなかった。IT革命が流行語になっている一方で、転居に伴う新住所、氏名、電話番号などの記入を数十回繰り返し、その大半は郵送または窓口への訪問による提出を伴っているのが現状である。

もう1つの身近な例として、某大学の学内(教員)向け事務処理手続きを見てみると各担当部署には数十種類の申請用紙が用意されている。そのうちのほんの一部がインターネットから入手可能になっているが、記入および提出は従来通りである。

このような分野では、アプリケーション開発を外注する予算の確保が難しい場合が多いと考えられるのが、

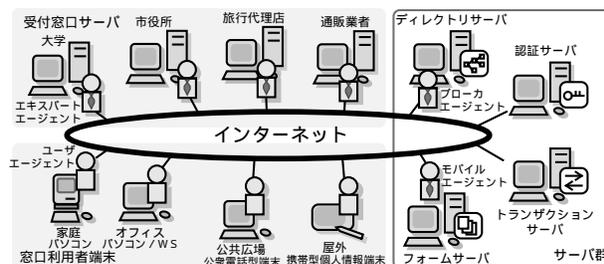


図1 多組織間ネットワーク上の分散オフィスシステム MOON の構成例

開発費の面だけでなく、サービスの変化に迅速に対応するという面からも業務の専門家が自らの業務を自ら自動化するためのエンドユーザ主導型アプリケーション開発技術が必要である。

電子商取引に代表される B2B や B2C のように投資に対する直接的な効果が期待できる分野に比べて、C2D(Client to Department)あるいはD2D(Department to Department)ともいえる上記のような分野がマーケットとしては桁違いに大きいと思われる。

### 3. アプリケーションフレームワーク wwHww の概要

#### 3.1 対象とするアプリケーション

本研究では、窓口業務を主体とした多組織間ネットワーク上の分散オフィスシステムを対象としている。マルチエージェントモデルをベースにした MOON (Multiagent-Oriented Office Network) システムの構成を図1に示す。図の受付窓口サーバは、窓口業務の担当者の端末である。依頼者端末は、従来は窓口へサービスの依頼に訪れる一般の人の端末である。

図の枠内に示すシステム共通のサーバは、以下のようである。

- ディレクトリサーバ：
    - 受付窓口のアドレスと業務(サービス)のディレクトリを管理
  - フォームサーバ：
    - 各種の提出書類のフォーム(書式)を管理
  - トランザクションサーバ：
    - 提出された書類とその識別番号を管理
  - 認証サーバ：
    - 書類の提出者の認証の管理
- 図の依頼者端末のユーザインタフェースでは、以下のような共通のインタフェースを提供する。
- 受付窓口の問い合わせ(検索)
  - 受付窓口からフォーム(書式)の取り寄せ

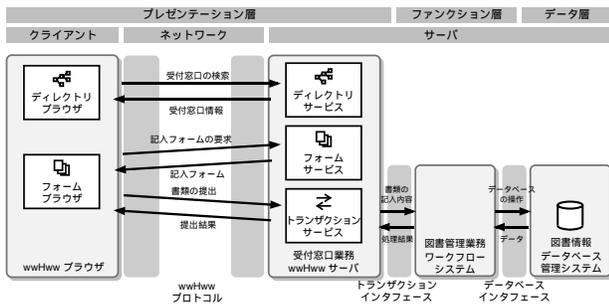


図2 図書管理システムの構成



図3 受付窓口業務アプリケーションフレームワーク

- フォームの表示とフォームへの記入
- 記入済書類の受付窓口への提出

### 3.2 例題システムの構築

このような窓口の受付業務の自動化のためのアプリケーションフレームワークを実現するために、まず、研究室内の図書管理業務を取り上げ、その一連の受付窓口業務を自動化した図書管理システムを開発した。図書管理システムの全体構成を図2に示す。

図書管理システムはネットワーク上での利用を考え、窓口利用者側と受付窓口側に分割したクライアント/サーバ型のシステム構成とした。現在、研究室で実際に使用しているシステム<sup>12)</sup>は、DBサーバを含めた3層アーキテクチャのwebアプリケーションになっている。webブラウザから利用できることにより、自宅や外出先で、図書リストの最新情報の確認や、購入した図書の登録が可能となっている。

### 3.3 窓口業務フレームワークの構成

図2の図書管理システムにおいて窓口業務アプリケーションに共通の部分はクライアント側のwwHwwブラウザ、ネットワークのwwHwwプロトコル、サーバ側のwwHwwサーバである。あらかじめこれらの共通部分を受付窓口業務におけるアプリケーションフレームワークとして提供することで、業務の専門家が自ら受付窓口業務を構築できると考えられる。図書管理システムのフレームワーク部分の構成を図3に示す。なお、図2の業務ワークフローシステムとデータベース管理システムは、図書管理業務に固有な部分なので図3のアプリケーションフレームワークには含ま

れない。

### 3.4 業務の専門家による構築手順

業務の専門家がアプリケーションフレームワークを利用して、窓口業務アプリケーションを構築する場合の具体的な手順は次のようになる。

- (1) サービスの定義
- (2) フォームの作成
- (3) 書類の処理方式の設定
- (4) サーバへの登録

フレームワークは、問題領域内の個々のアプリケーションに依存する部分(ホットスポット)と共通の部分(フローズンスポット)に明確に区別される。本システムでは、フレームワークの利用を容易にするために、以下の2種類の支援機能を実現している。

#### (1) 差分コンポーネントのプラグイン

窓口業務フレームワークの場合、提出された書類の処理方式は個々の窓口で異なるので、業務専門家が定義しなければならない。本システムでは、よく使われるデータベース蓄積方式と紙に印刷する方式をコンポーネントとして用意しておくことで、その定義を容易にしている。

#### (2) プロパティデータの設定

アプリケーションに固有の処理の作成方法としては、上記の差分コンポーネントの他に、固有のプロパティデータをフレームワークの指定した形式で用意すれば良い場合がある。例えば、窓口業務フレームワークでは、業務の専門家は受付窓口の説明などの記述や記入フォーム等はプロパティデータとして用意すれば良い。

### 3.5 知的ナビゲーション

#### 3.5.1 マルチエージェントモデル化

最初にも述べたように、本研究の目標は「エンドユーザが自分のエージェントを自ら作り、自ら利用することである。われわれはこのようなエージェントを知的クローン(IC: Intelligent clone)と呼んでいるが、図1には以下の4種類のソフトウェアエージェントが示されている。

- エキスパートエージェント: 受付窓口サーバに存在し、業務の専門家のノウハウを学習して、フォームへの記入の誘導、ヘルプメッセージの表示、記入内容のチェック等を行なう。
- ユーザエージェント: 依頼者端末に存在し、個々のユーザに固有の情報を学習し、電子フォームの名前や住所などの欄に自動記入する。<sup>10)</sup>
- モバイルエージェント: フォームサーバに存在し、クライアント側からの検索に応じて依頼者端末へ移動し、エキスパートエージェントの機能を果た

す。実際はエキスパートエージェントとモバイルエージェントは同一であり、電子フォームの中に組み込まれている。

- ブローカエージェント：ディレクトリサーバに存在し、クライアント側からの検索に応じて適切と思われるフォームを提示する。

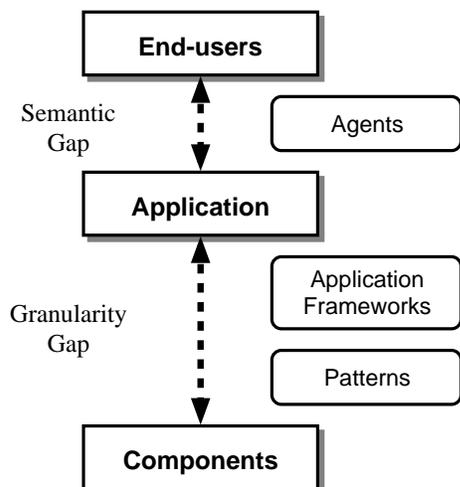


図 4 粒度的ギャップと意味的ギャップを埋める技術

図 4 にも示すように、アプリケーションフレームワークやパターンがアプリケーションとコンポーネントの間の粒度的ギャップを解消する技術であるのに対し、エージェントはエンドユーザとアプリケーションの間の意味的ギャップを解消するために有用である。

このようなマルチエージェントモデルでは、エージェント間コミュニケーション言語 (ACL : Agent Communication Language) が必要である。その代表的なものとして FIPA ACL<sup>6)</sup> があるが、われわれは FACL(a Form-based ACL)<sup>5)</sup> を開発した。主な相違点は、FIPA ACL があらかじめ 23 個の限定された対話行為 (Communicative Act) を用意しておくのに対して、FACL では、「1 サービス = 1 フォーム」という考えから、数を限定しないフォーム単位の対話を実現した。

そして、オブジェクト指向のメッセージ駆動型の分散協調モデルをベースにしたわかりやすい対話インタフェースとして「誰に何をどのように頼む」というメッセージにその識別番号 (どれ) を加えた 4 項目のパラメータを有する以下の基本形式を設定した。

このメッセージ形式にちなんで、本システムを wwHww (the Who-What-How with WWW system) と名付けた。

(Who, What, How, Which)

パラメータの説明

Who: メッセージの送信先

What: メソッド名

How: メソッドの実引数

Which: メッセージ識別番号

Who は、窓口すなわち依頼先あるいは書類の提出先である。What は、依頼業務の種類あるいは提出書類の名称である。How は、依頼業務の内容あるいは提出書類の書式である。Which は、依頼または提出書類を識別するための受付番号またはコードである。現在の web 検索でお馴染みのキーワード検索と同様に、What の項にキーワードを入力してブローカエージェントに検索依頼する方法をとっている。

### 3.5.2 XML による知的フォーム

実際のエージェントによる知的ナビゲーション機能は XML ベースで実現した。すなわち、窓口の記述には、ブローカエージェントによる自動処理に適した XML ベースの RDF (Resource Description Framework) 形式の記述言語を用いた。また、フォーム定義には従来の HTML を用いるが、その記入フォームの意味定義はフォームのメタデータとして、コンピュータが処理しやすい XML による RDF 形式の記述言語で定義する。このメタデータを用いてユーザエージェントやエキスパートエージェントによる自動記入、記入チェック、オンラインヘルプなどのナビゲーション機能が実行される。記述形式の詳細は文献<sup>8)</sup>に詳しい。

## 4. M-base のモデリングプロセスの概要

### 4.1 基本コンセプトと方法論

本研究では「すべての日常的な業務をコンピュータ化する」、即ち「日常的業務はマニュアル化でき、マニュアル化できればコンピュータ化できる」という観点から、オフィスでの業務を中心にグループウェアやワークフローシステムなどを対象とする。規模的には、中、小規模のアプリケーションソフトウェアを想定することになるが、ネットワーク接続することによりシステムとしては大規模化することもある。

基本的コンセプト<sup>3)</sup>として、

「ドメインモデル 計算モデル」

「分析 設計 プログラミング」

をにかけている。これは、問題領域を分析してドメインモデルを構築した時点でソフトウェアの開発を完了させようというものである。即ち、

「ソフト開発 = モデリング + シミュレーション」



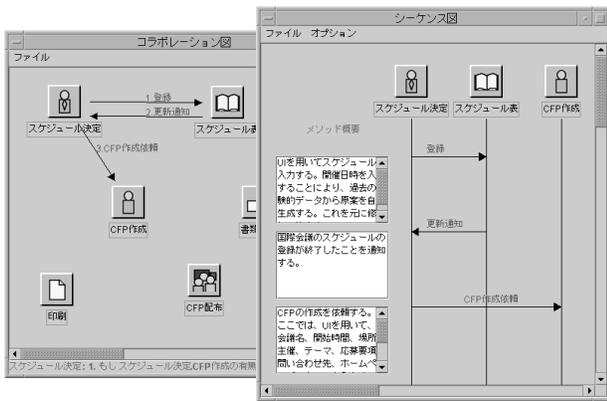


図 7 シミュレーション実行画面

- 制御用ビジュアルコンポーネント

- スクリプト言語

#### 4.7 ユーザインタフェースの構築

ユーザインタフェースはシステムから自動生成されるが、エンドユーザは必要に応じてこの UI をカスタマイズすることで、使い勝手の向上を図ることが可能である。すなわち、アプリケーション開発中のデバッグ作業では、UI 作成に手間がかからないように自動生成された UI を用い、完成後は利用者が使いやすいようにカスタマイズを行うことを想定している。

なお、この段階で UI 画面の遷移図を表示することにより、業務フローの概要を早期に検証できるようにしている。

#### 4.8 シミュレーション実行による検証

シミュレーションは、作成した業務フローの動作確認のためにユースケース単位で行う。図 7 はシミュレーションの実行画面であり、スケジュール決定オブジェクトがスケジュールを決定した後、CFP 作成オブジェクトに CFP の作成を依頼しているところである。このように業務フローの各メッセージをステップ実行し、起動したメソッドの内容をひとつずつビジュアルにチェックできることにより、エンドユーザによるシミュレーション実行を容易にしている。

### 5. おわりに

以上、エンドユーザ主導型アプリケーション開発技法の研究状況について述べた。実用化のためには個々の技術をできるだけオープンシステムとして構築することが重要である。開発ツールとしては、Java, XML, UML, JDBC などを利用しているが、さらにコンポー

ネントを JavaBeans や EJB 化すること、あるいはフォーム形式でのサービス授受に SOAP や UDDI を利用することも検討していく。

### 参考文献

- 1) 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 (1998) .
- 2) 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-97-5 (1994).
- 3) 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, No.95-SE-104-4(1995).
- 4) 中所武司, 小西裕治, 松本光由: メッセージフローに基づく分散協調型アプリケーション開発技法, 情報処理学会論文誌, Vo.40, No.5, pp.2387-2396 (May 1999).
- 5) Takeshi CHUSHO and Katsuya FUJIWARA : FACL : A Form-based Agent Communication Language for Enduser-Initiative Agent-Based Application Development, COMPSAC2000, IEEE CS, pp.139-148 ( Oct. 2000).
- 6) FIPA, "Agent Communication Language," FIPA Spec 2-1999, Draft ver.0.1, Apr. 1999.
- 7) 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, Vo.41, No.4, pp.1202-1211 (Apr. 2000).
- 8) 藤原克哉, 中所武司: 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 - 知的ナビゲーション機能の XML による実現方式 - , 情報処理学会 ソフトウェア工学研究会資料 2000-SE-128, pp.1-8 (July 2000)
- 9) 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, pp.133-140 ( Sep. 2000 )
- 10) 南谷圭持, 中所武司: 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 - 自動記入エージェント実現方式 -, 情報処理学会第 61 回大会講演論文集 ( 1 ) 1W-5, 273-274(Oct. 2000).
- 11) 大西淳: 要求工学ワーキンググループとその活動, 情報処理学会ウィンターワークショップ・イン・高知, 論文集, pp.21-24 (Jan. 1999).
- 12) 島田圭, 中所武司: フレームワークによる 3 層 Web アプリケーション構築法, 情報処理学会 ソフトウェア工学研究会資料, 2000-SE-129, pp.1-8 (Nov. 2000)