ソフトウェア工学 97－5
（1994．3．1）

# 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

中所武司
明治大学理工学部情報科学科
email : chusho@cs.meiji.ac.jp

近年，ワークステーションやパソコンの普及およびそれらをつなぐネットワークの普及と共に，オフィスの内外でエンドユーザが増加し，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．コンピュータによるより良い生活（"CS-life"：Computer-Supported Life）の実現の一つとして，「すべての日常的仕事はコンピュータが代行するべきである」という観点から，「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」が必須である．

本報告では，OIS (Office Information System) 分野の応用システムとして，窓口業務の自動化を取り上げ，業務の専門家が自らの業務を自らコンピュータ化できるような技術および一般の依頼者が自らコンピュータを操作できるような技術を検討する．特に，オブジェクト指向技術に着目し，分散環境下でのエンドユーザコンピューティングを実現するためのアプリケーションフレームワークwwHww (the who-what-how & when-where system) を提案する．

# wwHww : An Object-Oriented Model for End-User Computing in Distributed Office Systems

Takeshi Chusho
Meiji University, Department of Computer Science

The new trend of end-user computing and distributed computing increases end-users, who should develop their application software by themselves. A goal of information systems is realization of the better human life, which I call "CS-life" (Computer-Supported Life) . One of its approaches is that computers perform our whole routine work for us. Therefore, we need tools for developing application software of the end-users, by the end-users, for the end-users. This paper presents an application framework, wwHww (the who-what-how & when-where system), based on an object-oriented model while focusing on distributed office systems.

## 1．はじめに

情報システムは，従来，情報処理の専門家が開発し，限られた人達が利用してきた．しかし，近年，ワークステーションやパソコンの普及およびそれらをつなぐネットワークの普及と共に，オフィスの内外でエンドユーザが増加し，業務の専門家が自ら情報システムを構築する必要性も高まっている．この傾向は近い将来の多様なネットワークのさらなる普及により，情報のグローバル化と情報のパーソナル化という両面から一段と加速されるものと思われる．

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」(Infomation 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．

その第１段階として，OIS (Office Information System) 分野の応用システムとして，窓口業務を取り上げる．窓口業務が自動化され，かつネットワーク化されれば，窓口の担当者は上記（１）の観点で，また窓口への一般の依頼者は上記（１）または（２）の観点で恩恵をこうむる．そのためには，業務の専門家が自らの業務を自らコンピュータ化できるような技術が必要である．さらに，一般の依頼者が自らコンピュータを操作できるような技術が必要である．

本報告では，これらの要求に応えうる基本技術としてのオブジェクト指向技術に着目し，上記の分野を対象として分散環境下でのエンドユーザコンピューティングを実現するためのアプリケーションフレームワークを提案する．特にオブジェクト指向におけるメッセージ主体の分散協調型の計算モデルをベースに今回の業務モデルを構築する．

この分野の研究動向としては，分散オフィスシステムの観点ではすでに電子メールベースのシステム[16]などがある．オブジェクト指向技術[4]の観点では，プログラミングレベルでの実用化が進み，多くの設計法が提案されている．その詳細については，６．で述べる．

以下では，まず２．で今回の応用分野の問題を明確にし，３．でアプリケーションフレームワークの設計思想，４．でシステムモデル，５．でソフトウェアアーキテクチャについて述べる．

## 2．応用問題の概要

### 2．1　シナリオ

本報告で取り上げるオフィスの窓口業務について，そのイメージを明確にしておく．

例えば転居の場合，その前後で依頼者の立場でどのくらいの窓口業務が発生するだろうか．オフィスおよび家庭での窓口を思いつくままにあげると以下のようなものがある．

[オフィス]

勤労課，経理課，庶務課，．．．

[家庭]

区役所，水道局，NTT，郵便局，警察署，陸運支局，地方法務局，学校，電力会社，ガス会社，NHK，新聞販売店，銀行，保険会社，証券会社，クレジット会社，学会，出版社，同窓会，．．．

この中には電話で済むものもあるが，何らかの書類を提出するものも多い．さらにその多くは郵送ではなく，直接窓口へ行く必要がある．これらの処理がすべて自分のオフィスあるいは家庭のコンピュータから行えれば，時間の節約だけでも大きい．

一方，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，同じような説明を繰り返し，提出された書類の記入内容のチェックをし，ほぼルーチンワーク化した判断処理をしている．これらの処理がコンピュータ化されれば，その分サービスの向上や費用の低減につながる．

急速な老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．

このような観点から，本報告では応用問題として，われわれの日常生活に身近な窓口業務を取り上げ，その担当者の視点と依頼者の視点の両方から自動化を検討する．

## 2．2　特長

このような応用システムについて，WHY（なぜ開発？）の観点からその効果について述べておく．

（1）窓口への依頼者の利点

(a) 依頼方法（手続き）の容易化

誰に／何を／どのように頼めばよいか，あるいはどのように記入すればよいかなどについて，容易に知ることができる．

(b) 書式（書類）の入手／提出の容易化

窓口へ出向く必要がない．

(c) 結果の迅速な応答

(d) 処理状況のフォローの容易化

依頼業務の状況確認が簡単にできる．

(e) その他

窓口のたらい回し，窓口での順番待ちがなくなる．住所，氏名，電話番号などの手書き記入が不要になる．等々．．．

（2）窓口業務の担当者および組織の利点

(a) 準ルーティンワークからの解放

手続き方法に関するヘルプ機能（問い合わせ，誘導，例示）を自動化できる．

(b) 業務知識（ノウハウ）の共有財産化

マニュアル化がしづらい属人的な断片的ノウハウのコンピュータ化が容易である．

(c) より創造的な業務の達成

業務の効率化にともない，業務の質の向上により多くの時間をかけられる．

## 3．設計思想

### 3．1　研究目的

本報告での研究目的は，来たるべき情報化社会の"CS-life"をみすえて，エンドユーザコンピューティング（EUC）ツール[5]を開発することである．具体的には，分散環境下でのオフィスで使用するアプリケーションを容易に開発するためのアプリケーションフレームワークの構築である．

### 3．2　基本方針

そのためには以下の目標を達成する必要がある．

（1）業務の専門家が自らの業務を自らコンピュータ化できること．

（2）一般の依頼者が自らコンピュータを操作，またはその操作のコンピュータ化をできること．

上記（1）については，業務の専門家はすでに頭の中にはプログラムをもっているので，そのプログラムを如何に素直にコンピュータの中に移植するかという問題である．（2）については，実世界において，窓口への依頼者は担当者のアドバイスや記入方法の説明書を参考にして自分で提出書類に記入をしているので，それと同等の操作を実現し，さらにその操作を自動化するという問題である．いずれも，これまでのソフトウエアの手続き的記述にかわり，極力プログラミングの概念を排した新しいパラダイムが必要となろう．

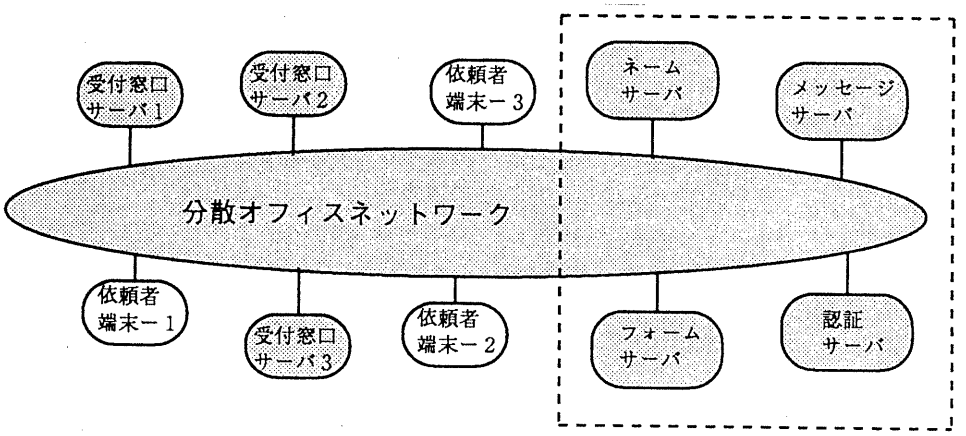


図1　窓口業務を主体とした分散オフィスシステムの構成例

## 3．3　研究アプローチ

研究の進め方については，全体像を描きながら重要な要素技術を順に開発していく．当面は以下のように進める．

（1）まず，身近な応用として，2．で述べた窓口業務の分散オフィスシステムを想定する．図1にそのシステム構成例を示す．

（2）本研究の対象と相性の良いオブジェクト指向の分散協調モデルをベースにシステムモデルを構築する．

（3）インタフェースを設定する．

（4）プロトプログラムを開発する．

# 4．システムモデル

## 4．1　対話インタフェース

今回の応用システムについて，WHAT（何を開発？）の観点からそのシステムモデルについて述べておく．

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 頼み，いつ 何処へ 結果を渡す」というメッセージにその識別番号（どれ）を加えた6項目のパラメータを有する以下の基本形式を設定する．

```
(Who: ,What: ,How: ,When: ,Where: ,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
    When：期限
    Where：結果の返信先
    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，依頼業務の種別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．Whenは，その業務の期限である．Whereは，結果の返信先である．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．

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 & when-where system) と名付ける．発音はとりあえず "who" とする．

## 4．2　機能仕様

簡単に各パラメータの仕様を示す．

Who：メッセージの送信先

定数：依頼先

変数：同上またはその問い合わせ

（注）定数または変数を複数用いた階層（ネスト）指定および中間の省略可（例：a[.b.c].d）

What：メソッド名

定数：依頼業務種別または提出書類名称

変数：同上またはその問い合わせ

（注）定数または変数を複数用いた階層（ネスト）指定および中間の省略可（例：[p.q.r.]s）

How：メソッドの実引数

定数：依頼業務内容または提出書類の書式

変数：同上またはその問い合わせ

When：期限

定数：処理期限／応答時期

変数：同上またはその問い合わせ

Where：結果の送信先

定数：返信先

変数：同上またはその問い合わせ

Which：メッセージ識別番号

定数：依頼または提出書類の識別番号

変数：同上またはその問い合わせ

（注）依頼時に割り当て，問い合わせに利用．

## 4．3　外部仕様

上記のメッセージ形式は基本的には内部仕様であり，実際のシステムとユーザの間のインタフェースは別に定める．メッセージはテンプレート言語として扱い，表示／入力形式は，フォーム（表形式書式）の穴埋め式，テキスト（キーワード）の逐次表示による入力誘導，さらに項目毎のアイコンやメニューの表示／選択など，具体的な形態は依頼者端末に依存するが，いずれにせよ，わかりやすさ，操作の簡便さ，操作誤りの回避などを考慮する．

## 4．4　使用例

システムの詳細な仕様は未だ確定していないので，ここでは使用例を示すことで内容を説明する．ただし，簡潔に説明するために，外部仕様の表示形式ではなく，以下の基本形式を用いる．

(Who: ,What: ,How: ,When: ,Where: ,Which: )

a,b,...：定数または値のバインドされた変数

?a,....：同上で指定された項目の説明要求（ヘル

プ機能）

x,y,...：値が未定義の変数

?x,....：同上に対応する項目の問い合わせ（全解探索）

（1）業務依頼の例

(a, b, c, d, e, x)

窓口aに処理bを依頼するために書類cを提出し，結果を期限dまでに部署eへ渡すように指示し，受け取った受付番号をxに記入する．入力形式は依頼者側のエージェントに依存する（以下同様）．

(a, b, c, d, e, f)

窓口aに依頼済みの処理b（受付番号f）の内容変更を依頼する．変更内容はc, d, eの一部又は全部．（再提出と同じ）

(a, b, , , , ?f)

窓口aに依頼済みの処理b（受付番号f）の状況が表示される．

(a, b, , , , -f)

窓口aに依頼済みの処理b（受付番号f）に対して取消を依頼する．

（2）依頼先，業務種別，書式の問い合わせの例

(a, p, ?x)

窓口aに処理pを依頼するための書式が表示され，入力が誘導される．表示形式は依頼者側のエージェントに依存する（以下同様）．

(a.b, ?x)

窓口a.bが担当する処理一覧が表示される．

(?x, p)

処理pを担当する窓口がすべて表示される．

(?x, ?y="p")

キーワード"k"に関連した処理を担当する窓口とその処理をすべて表示する．キーワード"k"が処理の名称または説明の中に含まれているものを検索して表示する．（具体例：駐車許可を得るために"駐車"というキーワードでその担当部署と手続きを調べる．）

(?x, ?y)

すべての窓口とその担当する処理が表示される．

(?x.a, b.?y)

aという窓口を持つすべての組織が表示され，その窓口の担当するbに関するすべての処理も表示される．

（3）業務内容の説明要求の例

(?a)

窓口aの業務内容が説明される．

(a, ?b)

窓口aが担当する処理bの内容が説明される．

# 5．ソフトウェアアーキテクチャ

## 5．1　システム構成

アプリケーションフレームワークwwHwwについて，HOW（如何に作るか？）の観点からそのソフトウェアアーキテクチャについて述べておく．窓口業務を主体とした分散オフィスシステムのシステム構成の例を図1に示す．将来的にはLAN間接続を含めたWANなどの広域のネットワークを対象にすべきであるが，当面は1組織内程度を検討する．図の受付窓口サーバは，窓口業務の担当者の端末である．この業務の専門家は従来のエンドユーザコンピューティングにおけるエンドユーザである．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．この依頼者もwwHwwシステムではエンドユーザコンピューティングのエンドユーザになる．

図の破線の枠内に示すシステム共通のサーバは，以下のようなものである．

・ネームサーバ：メッセージの送信先を管理．

・メッセージサーバ：メッセージの識別番号を管理．

・フォームサーバ：各種の提出書類のフォーム（表形式書式）およびその対応する業務担当部署と業務の内容を管理．窓口を特定しない問い合わせに対する情報を含む．

・認証サーバ：書類の提出者や閲覧者などの認証の管理．

これらの共通サーバは実際の構成では複合化する可能性がある．

## 5．2　設計思想

（1）分散協調型モデル

ビジネス分野の基幹業務向きのデータモデル中心の設計法，エンジニアリング分野の状態遷移モデル中心の設計法に対し，wwHwwでは，対象とする分散オフィスシステムに適すると思われる分散協調型モデルに基づくオブジェクト指向設計法を基本とする．

（2）トップダウン設計

開発方法論が異なっても既存の類似システムが存在する場合は，データベースやファイルの構造あるいはユーザインタフェースの操作などについて，ある程度の見通しが得られるので，クラスオブジェクトの設計やクラスオブジェクト間の関係の定義を初期の段階で行うボトムアップ的な開発

方法が可能である．一方，非定型業務を中心とした分散オフィスシステムの場合は，一般に前例のない新規開発の場合が多く，トップダウンに設計する必要がある．

本研究のアプリケーションフレームワークの目的は，応用分野を限定して，あらかじめ枠組みを作り，設計手順を確立しておくことにより，このようなトップダウン設計を基本にする場合にも個々のアプリケーションを簡単に構築できるようにすることである．将来的にはクラスライブラリの充実などにより，設計技法という概念そのものが希薄になっていくと思われる．

（3）プロトタイピングアプローチ

従来の受注ソフトウエアは一般に大規模のものが多く，多人数開発になるので，ウォータフォールモデルあるいはその改良方式を用い，各工程で仕様を検証しながら開発を進めるフェーズドアプローチをとる．それに対し，本報告で対象とするエンドユーザコンピューティングに近い分野では，まず核になる部分を試作し，それを改良しながら実用システムに仕立てあげるプロトタイピングアプローチをとる．

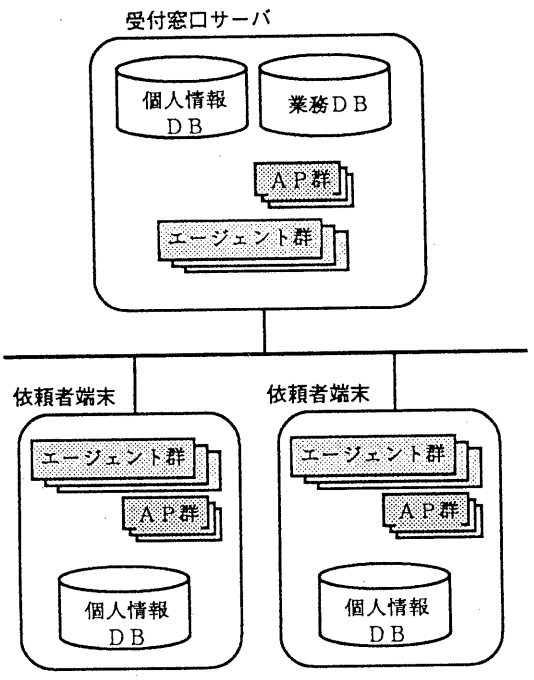


図2　エンドユーザの業務代行者としてのマルチエージェントモデル

## 5．3　要素技術と検討課題

具体的な研究開発項目を以下に示す．

（1）モデリング

分散オフィスシステムの業務モデル／計算モデルの構築．APIのレベルの設定．グループウエアあるいはCSCWとの連携．

（2）言語

EUC向き仕様記述言語（OOPL）の設計．特に，マクロレベルのオブジェクト間インタフェース言語とミクロレベルのオブジェクト定義言語．さらに並列処理の導入．

（3）GUI（ブラウザ）

表示系／入力系とＡＰを完全分離するUIMSの設計により，表示形式／入力形式に関し，画一性と多様性を実現．さらにナビゲーション方式．

（4）エージェント

AI技術を応用し，マルチエージェントモデル（図2参照），ルール記述言語およびそのインタプリタの開発．

（5）知的検索

あいまいな問い合わせに対する関連事項の検索．

（6）開発環境

・編集系：各種エディタ．クラスライブラリ構築／検索支援．

・処理系：インタプリタ．コンパイラ（C＋＋出力）．実行時ライブラリ．

・検証系：デバッガ．シミュレータ．アニメーション．

（7）実行環境

・オブジェクト管理：ハイパーテキスト，ハイパーメディア，リンクベース支援．

・ネットワーク：クライアント・サーバモデルまたはオブジェクトモデル．

・フォームサーバ：書式の登録配布（原本管理）に時間がかかる場合，クライアント側にコピーを持つ必要があり，原本との不一致問題を実世界にそくして解決．

・認証サーバ：本人の確認のための識別，認証機能．アクセス権の制御や実世界の押印相当の機能に利用．

・ネームサーバ：オブジェクト名とメソッド名の管理．

・メッセージサーバ：書類の識別番号とそのライフサイクルの管理．

・データベース：オブジェクト管理との連携．

## 6．他の研究との関連

### 6．1　研究動向

#### 6．1．1　分散オフィスシステム

応用の観点からは，電子メールベースシステム，ワークフローシステム，グループウエアなどのOIS (Office Information System ) と関連がある．

（1）電子メールベースシステム

MITのObject Lens [11] およびその発展したOval [12] の例のようなアプローチがある．エンドユーザが自らの協調作業用応用システムを構築するための簡潔で強力なツールの開発を意図している．内容を簡単に図式化すると，次のようになる．

{オブジェクト＋ハイパリンク}*{操作，検索，表示}*{自動化（エージェント）}

オブジェクトをフォルダーで管理し，オブジェクト間のリンクを可能とし，その処理のためのユーザインタフェースはテンプレートを用い，処理の自動化はルールベースのエージェントが行う．ちなみにOvalはobjects，views，agents，links，の頭文字をとったものである．

同じく電子メールをベースに電子秘書の観点でフォームに着目したシステムとして，PAGES [10] がある．電子メールのような疎結合ネットワークに各個人の電子秘書（ルールベースのエキスパートシステム）がつながり，ルール内蔵の知的フォームと呼ぶオブジェクトを処理する．

これらのシステムはいずれもエンドユーザコンピューティングツールを目的にしている点でwwHwwと同様であるが，処理する対象が主にテキストメールであるため，汎用性がある反面，自動化のためのルール記述が複雑になるものと思われる．

（2）ワークフローシステム

受発注処理や経理伝票などの定型的な業務の作業の流れ（ワークフロー）を管理し，自動化するシステム [14] がすでに商品化されている．文書フォーマット定義，回覧経路（ルーティング）の制御などの基本機能を持つ．これらのシステムは，複数のエンドユーザ部門および基幹業務／基幹データベースと関連するため，エンドユーザ自身による開発にはなじみにくい．

（3）グループウエア

これは非常に広い概念として使用されており，上記項目を含むものである．今回対象とした窓口業務に限れば，「複数の業務担当者が協調して一つの仕事を成し遂げる」という面は弱いが，技術的な関連があり，将来的には関係が深くなる．

#### 6．1．2　オブジェクト指向設計モデル

オブジェクト指向概念の定義は必ずしも明確ではないが，actorモデルなどのメッセージ交換による分散協調型計算モデル，Simula67からSmalltalk-80につながるプログラミング言語概念およびＥＲモデルなどのデータモデルがベースになっている．既存のオブジェクト指向設計技法は，G.Booch[1], P.Coad &E.Yourdon[8], S.Shlaer & S.J.Mellor[17], J.Rumbaugh[15]等，などをはじめとしてデータモデルに重きを置いたものが多い．

本報告では，システム全体の動的ふるまいに着目する立場から，分散協調型の計算モデルとしての観点を重視する．

設計技法をモデリング技法とみた場合，従来のオブジェクト指向分析，設計技法は３種類のモデルを組み合わせたマルチパラダイム型のものが多い．S.Shlaerらの情報モデル，状態モデル，プロセスモデル，J.Rumbaughらのオブジェクトモデル，動的モデル，機能モデルなどがその例である．これらの技法に共通する特徴は，設計プロセスの初期の段階でオブジェクトおよびオブジェクト間関係の定義を重視することである．そのため，とくにオブジェクト間関係のビジュアルな表記法が豊富に導入されている．

これらのボトムアップ的アプローチは，既に開発運用実績を有する従来の基幹データベースを中心としたビジネスシステムの再構築のような場合には適用可能である．しかし，システム全体のマクロレベルの動的ふるまいの設計法があいまいであることや３つのモデルの統合が不明確などの欠点[9,13]があるため，本報告で対象としているエンドユーザコンピューティングに近い非定形業務の新規開発には適さない．

このような分野では，何をオブジェクトにするかを決定するためには，データモデルよりも計算モデルを重視し，モデリングの初期の段階で，オブジェクト群の動的なふるまいを記述することが重要である．

### 6．2　研究経過

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．

著者らは，オブジェクト指向を基本としたマルチパラダイム型言語[3]の開発時に提案した２階層モデルをベースに，ミクロモデルよりもマクロモデルを，静的構造よりも動的ふるまいを優先する設計プロセス[7]を検討してきた．本報告の対象とする分野ではこの方式が適すると考えている．

このマルチパラダイム型言語ES/X90を用いて開発した知的電子秘書システムKISS (Knowledge-based Intelligent Secretary System) [6]では，ジョブの種類に応じたクラス階層の導入やジョブの生成，管理およびエージェントへのジョブ依頼のメッセージ送信はオブジェクト指向で記述した．個々のエージェントの処理に関しては，ノウハウ的なものをルールで記述し，手続き的なものはPrologで記述した．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．

分散オフィスシステムに関しては，C++を用いて，会議開催事務処理システムOOO (Object-oriented Office System)[6] のプロトタイプを開発し，オブジェクト指向の分散協調モデルが適することを確認している．しかし，エンドユーザコンピューティングツールとしては，C++よりもかなり抽象度の高い記述方式が必要である．

これらの経験が今回の「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」開発のベースとなっている．

## 7．おわりに

コンピュータによるより良い生活"CS-life"の実現のために「すべての日常的仕事はコンピュータが代行するべきである」という視点から，「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」としてのアプリケーションフレームワークwwHwwを提案した．CSという略語からは"Customer Satisfaction", "Client/Server", "Communication Satellite", "Computer Society", "Computer Science"などが連想され，興味深い．

今後は，大学における窓口業務を例としてプロトプログラムを開発していく．

## 参考文献


1) Booch,G.:Object-Oriented Design with Applications, Benjamin/Cummings(1991).

2) Chusho,T. and Haga,H.:A Multilingual Modular Programming System for Describing Knowledge Information Processing Systems,Proc. the 10th World Computer Congress IFIP'86,pp.903-908(1986).

3) 中所,増位,芳賀,吉浦:マルチパラダイム型言語における対立概念の融合方式,人工知能学会誌, 4, 1, 77-87(1989).

4) 中所武司:使いやすいソフトウエアと作りやすいソフトウエアーオブジェクト指向概念とその応用ー,電気学会雑誌, 110, 6, 465-472(1990).

5) 中所武司:エンドユーザコンピューティングーソフトウエア危機回避のシナリオー,情報処理,32, 8, 950-960(1991).

6) 中所武司：ソフトウェア危機とプログラミングパラダイム，啓学出版(1992).

7) 中所：オブジェクト指向概念の発生学的定義に基づくソフトウェア設計技法，情報処理学会研究報告，93-SE-95-7, 47-54 (1993).

8) Coad,P. and Yourdon,E.:Object-Oriented Design, Prentice Hall(1991).

9) Fichman,R.G. and Kemerer,C.F.:Object-Oriented and Conventional Analysis and Design methodologies, IEEE Computer, 25, 10, 22-39(1992).

10) Hammainen, H., Alasuvanto, J., and Arrpe, H. : Service interface approach in distributed loosely coupled information systems, Office information systems : the design process, 183-198, North-holland (1989).

11) Lai,K., Malone, T., and Yu, K. : Object Lens : A "spreadsheet" for cooperative work, ACM Trans. Office Information Systems, 6, 4, 332-353 (1988).

12) Malone, T., Lai, K. and Fry, C. : Experiments with Oval : A radically tailorble tool for cooperative work, Proc. CSCW92, 289-297 (1992).

13) Monarchi,D.E. and Puhr,G.I.:A Research Typology for Object-Oriented Analysis and Design, Comm. ACM, 35, 9, 35-47(1992).

14) ニナ・バーンズ：業務に密着した情報を伝達するワーク・フロー管理ソフト，日経コンピュータ，1993.7.26号，147-151(1993).

15) Rumbaugh,J.et al.:Object-Oriented Modeling and Design,Prentice Hall(1991).

16) 坂下善彦：グループウェアにおけるグループ活動モデルの概要，情報処理,34, 8, 1037-1045 (1993).

17) Shaler,S. and Mellor,S.J.:Object-Oriented Systems Analysis:Modeling the World in Data,Prentice Hall (1988).