

エンドユーザ指向の分散アプリケーション フレームワークwwHwwと例題への適用

明治大学工学部情報科学科
中所 武司

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。本報告では、全ての日常的業務はコンピュータが代行すべきであるという観点から、エンドユーザが自分の業務を代行する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、窓口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wwHwwを提案し、その適用実験について述べる。

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が高まっている。

本研究では、「すべての日常的業務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」の開発の実現をめざす。

その第1段階での応用システムとして、窓口業務の自動化を取り上げる。この種の窓口業務のアプリケーションは、既にインターネット上に次々と実用化されているが、個別に開発されているため、開発コストに見合った対象に限定される。一方、依頼者にとっては、アプリケーションの増加が便利さを招く反面、ユーザインタフェースは多組織間で統一されておらず、必ずしも使い勝手の良いものではなく、個別に操作方法を習得する必要がある。

本研究では、業務の専門家自身が窓口の受付業務を自動化するためのアプリケーションフレームワークを開発することにより、上記の問題を解決することを目的とする。主な技術的な特徴は以下のようなものである。

- (1) オブジェクト指向技術のメッセージ機能を拡張した通信プロトコル
- (2) クライアント側の機種非依存の電子フォーム処理方式
- (3) クライアント側およびサーバ側のエージェントによるナビゲーション
- (4) クライアント側の多様な問い合わせに対応するサーバ側のサービス管理方式

現在、インターネット・イントラネットをベースとした例題システムとして、(1)、(2)を開発し、引き続き(3)、(4)を開発中である。

2. 応用システムの概要

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。その数は、区役所、水道局、電話局、郵便局、警察署、陸運支局、地方法務局、など枚挙にいとまがない。

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。

2.2 応用システム

本研究で対象とする、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 MOON (a Multi-Organizational Office Network system) のシステム構成の例を図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。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。この依頼者もエンドユーザである。

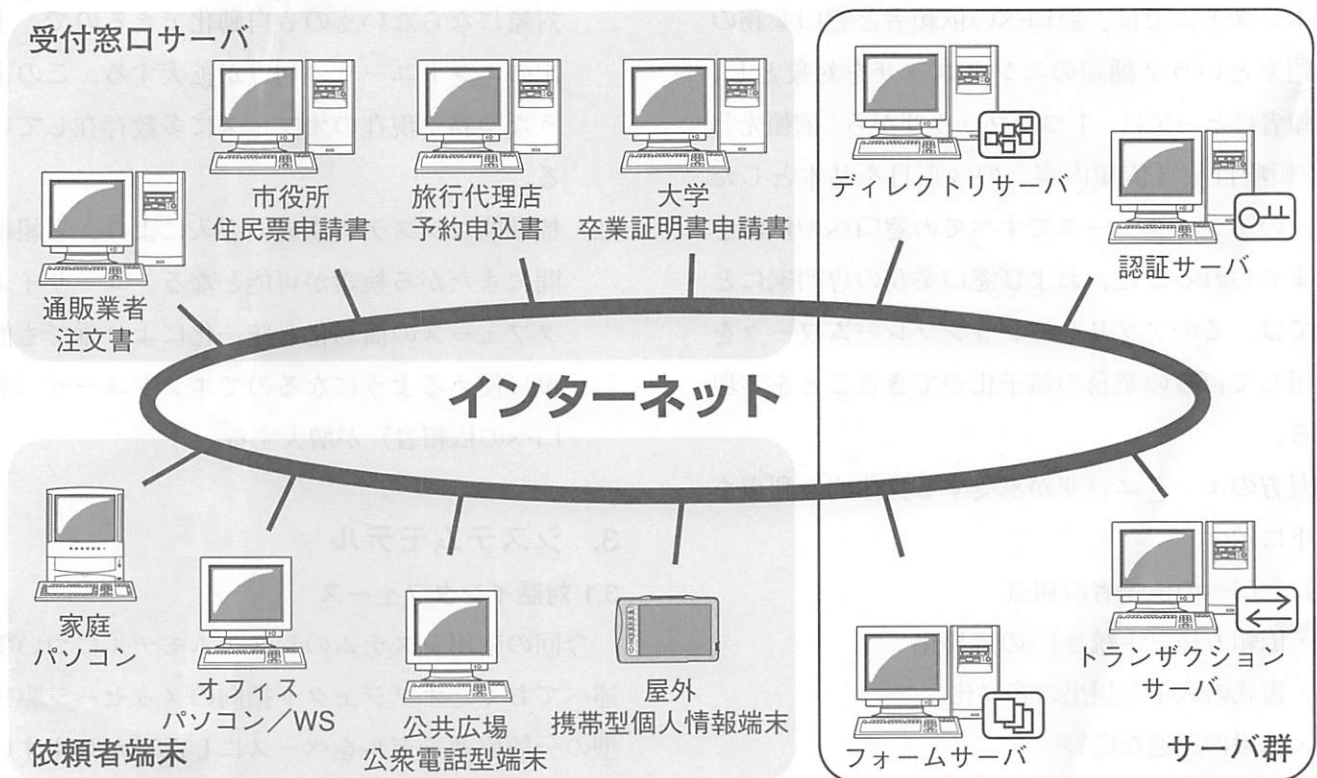


図1 多組織間ネットワーク上の分散オフィスシステムMOONの構成例

図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。

- (1) 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
- (2) フォームサーバ：各種の提出書類のフォーム（書式）を管理
- (3) 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
- (4) 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
- ・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

2.3 2種類のエンドユーザの利点

本システムでは、窓口への依頼者と窓口業務の専門家という2種類のエンドユーザを対象とし、依頼者にとっては、1つのブラウザから「依頼先」、「依頼項目」、「依頼内容」の3項目を基本とした同一のインタフェースですべての窓口への依頼を済ませられること、および窓口業務の専門家にとっては、このアプリケーションフレームワークを利用して自らの業務の電子化ができることを実現する。

双方のエンドユーザが享受する具体的な利点を以下に示す。

(1) 窓口への依頼者の利点

- ・依頼方法（手続き）の容易化
- ・書式の入手と提出の容易化
- ・結果の迅速な応答
- ・処理状況のフォローの容易化

その他、窓口のたらい回しや順番待ちがなくなることや、住所、氏名、電話番号などの記入が不

要になることなど。

(2) 窓口業務の担当者および組織の利点

- ・ルーティンワークからの解放
 - ・業務知識（ノウハウ）の共有財産化
- その他、業務の効率化にともない、創造的な業務に多くの時間をかけられることなど。

2.4 他システムとの比較

なお、これらのエンドユーザの利点のうちの幾つかは、インターネット上で稼働するアプリケーションを個別に開発して実用化したものでも達成することができる。しかしながら現在の個別開発と比較した場合、本研究のアプリケーションフレームワークwwHwwを用いてエンドユーザ（業務の専門家）主導で開発・保守する場合の主な特長は次のようなものである。

(1) 現状ではコストや技術的な問題から自動化の対象にならないものも自動化できるので、上記のエンドユーザの利点が拡大する。このような業務が現在のオフィスに多数存在している。

(2) 標準的なシステム構成の導入により、多組織間にまたがる検索が可能となる。ユーザインタフェースの簡易化と統一化により誰でも簡単に使えるようになるのでエンドユーザ（窓口への依頼者）が増大する。

3. システムモデル

3.1 対話インタフェース

今回の応用システム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

形式を設定する。

(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。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 with WWW system) と名付ける。今後、急激に増加すると思われるインターネットを利用した受付業務のアプリケーションが個別にユーザインタフェースを設定することにより、“便利だけれど面倒”という状況が生じることが予想されるが、wwHwwシステムがそれを解決できるものと思われる。

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) 階層指定および中間省略可 (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

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

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

3.3 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。

実世界の窓口利用者と受付窓口との対話は、書類のフォーム(書式)の取り寄せ、フォームへの記入、書類の提出という書類を媒介とした対話と考えられる。本システムでは、フォームを電子フォームに置き換え、書類の表示・記入のための、統一されたインタフェースを受付窓口のクライアントとして実現する。

現在のところ、標準的な端末としては Java 環境が利用できるものを想定することにより、マルチプラットフォームへの対応を容易にしているが、今後、携帯情報端末などの分野で多様な形態の端末が普及するものと思われる。いずれの場合も、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必要がある。

3.4 使用例

システムの使用例を示すことで内容を説明する。特にここでは wwHww で設定した誰に (Who) 何を (What) どのように (How) 頼むという 3 種類のパラメータと受付番号 (Which) パラメータを用いて、受付窓口業務に関連したほとんどの依頼内容を表現できることを示す。

なお、簡潔に説明するために、外部仕様の表示形式ではなく、(Who, What, How, Which) の基本形式を用いる。パラメータの a, b は定数または値のバインドされた変数を意味する。x, y は値が未定義の変数を意味する。

(1) 業務依頼の例

(a, b, c, x) : 窓口aに処理bを依頼するために書類cを提出し、受け取った受付番号をxに記入。

(2) 依頼先、業務種別、書式の問い合わせ例

(a, b, ?x) : 窓口aに処理bを依頼するための書式を表示し、入力を誘導。

(a.b, ?x) : 窓口a.b担当の処理一覧を表示。

(?x, b) : 処理bを担当する窓口をすべて表示。

(?x, ?y= "k") : 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。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する。たとえば、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時に使う。

(3) 業務内容の説明要求の例

(?a) : 窓口aの業務内容の説明。

(a, ?b) : 窓口aが担当する処理bの内容の説明。

4. 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。

4.1 システム構成

図書管理の受付窓口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。両者はインターネットを通じた共通プロトコルにより対話を行う。ネットワーク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る。

本システムは、マルチプラットフォームに対応したJavaを利用して構築している。現在、Solaris2、Windows95、WindowsNTのプラットフォーム上で動作することを確認済である。

4.2 サーバ側システム構成

4.2.1 概要

図書管理システムのサーバ側システム構成は、受付窓口業務の基本フレームワークに対応するwwHwwサーバとそのバックエンドに位置する業務ワークフローシステムおよびデータベース管理システムで構成した。

業務ワークフローシステムおよびデータベース管理システムは、受付窓口業務のアプリケーションフレームワークの観点では本来取り扱わない部分であるが、実用上は連携して利用されることが多いと思われる。なお、データベース管理システムにはOracleを利用している。

wwHwwサーバは窓口利用者側から主に次の3種類の要求を受け付ける。

- ・ 受付窓口と記入フォーム（書式）の検索
- ・ 記入フォームの取り寄せ
- ・ 書類の提出

そこで、図2に示すように、それぞれの要求に対応する次の3つからなる構成とした。

(1) ディレクトリサービス

フォームの入手先、書類の提出先などを含む受付窓口情報をディレクトリ構造で管理し、窓口利用者からの多様な検索に対応。

(2) フォームサービス

記入フォームの書式データを管理し、受付窓口利用者からの要求に応じて記入フォームを提供。

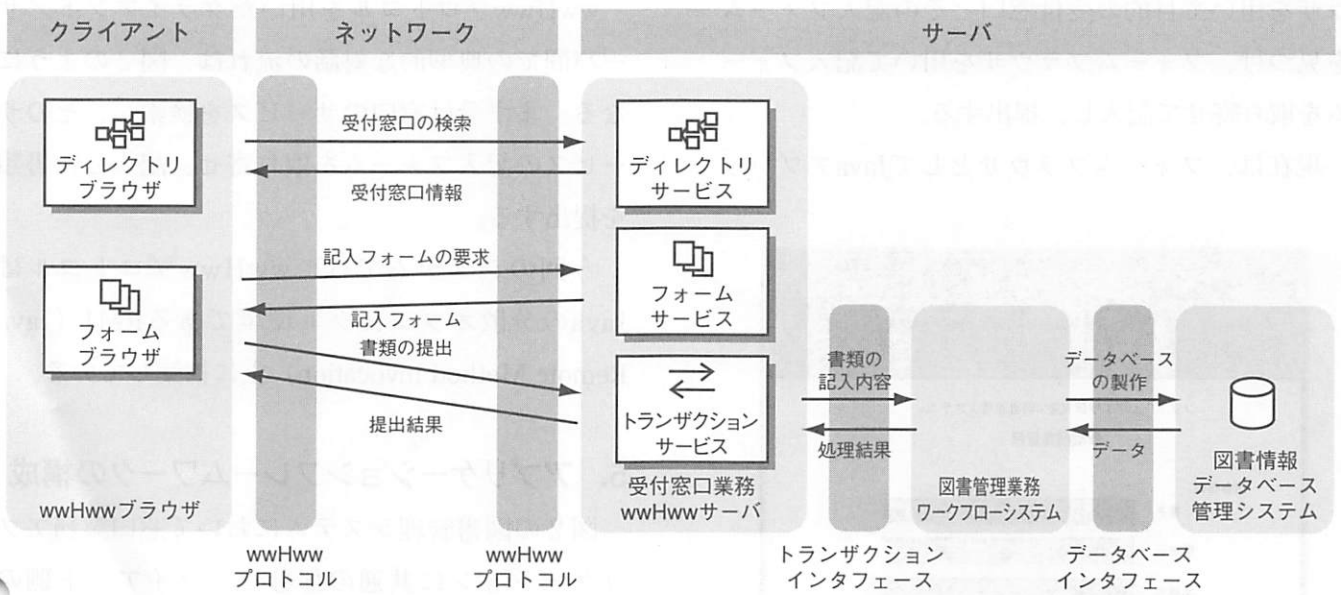


図2 図書館管理受付窓口システムの構成

(3) トランザクションサービス

窓口利用者から提出された書類を受け取り、業務ワークフローシステムと連携してトランザクション管理機能を提供。

4.2.2 サービス管理方式

受付窓口の処理方式は、利用者に分かりやすく、業務の担当者に管理しやすくする必要がある。こ

のような機能を実現するために、本システムでは受付窓口の名前空間を実社会の組織の構造に則したツリー構造のディレクトリとした。ディレクトリでは、受付窓口名、検索用のキーワード、サービスの説明、サービスの位置情報（物理アドレス）を管理する。

ディレクトリ構成の例を図3に示す。図のMeiji Univ.以下のノードは組織の階層構造を、末端のノードはその部署で行われている窓口のサービスを表している。このようなディレクトリモデルは、組織内の階層構造をそのまま反映できるイントラネットに適している。

しかし、多組織間にわたるインターネット上の名前空間を扱う場合は更に概念的なサービスのカテゴリ分けを考慮する必要がある。

4.3 クライアント側システム構成

図書館管理システムと受付窓口利用者とのインタフェースとなるwwHwwブラウザは、図2に示すようにディレクトリブラウザとフォームブラウザからなる。受付窓口利用者は、ディレクトリブラ

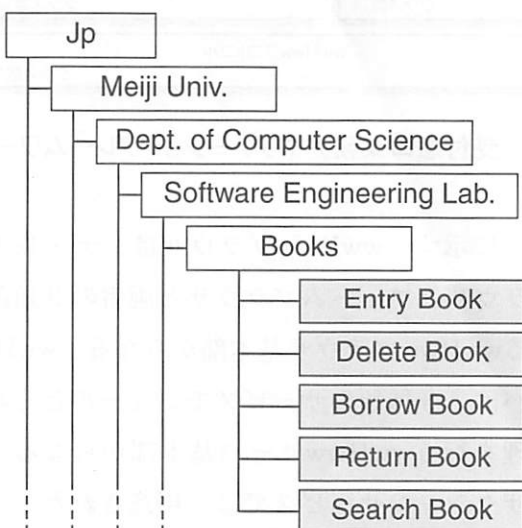


図3 ツリー構造のディレクトリ構成の例

ウザを用いて目的の受付窓口とその記入フォームを見つけ、フォームブラウザを用いて記入フォームを取り寄せて記入し、提出する。

現在は、フォームブラウザとしてJavaアプレッ

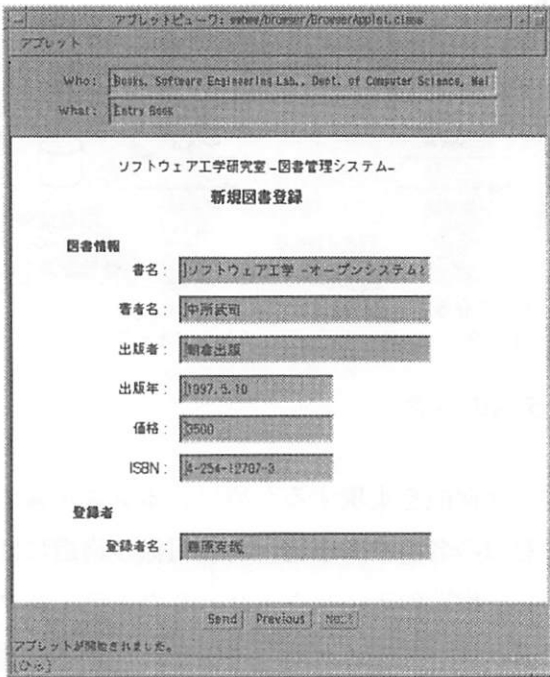


図4 フォームブラウザの使用例

ト版とJavaアプリケーション版を開発した段階である。図4にフォームブラウザの使用例を示す。アプレット版はプログラムをネットワークからロードして実行するため、Java対応のWWWブラウザがあればネットワーク上のどこからでも利用可能であり、外出先での利用に便利である。しかし、プログラムのロードに時間がかかる欠点がある。一方、オフィス内や自宅から利用する場合には、アプリケーション版をあらかじめ用意しておくことによって応答性が向上する。

4.4 共通プロトコル

インターネット上で先に定義したシステムモデルの対話インタフェースを実現するためにwwHwwプロトコルを設定した。

wwHwwプロトコルを用いたクライアント/サーバ間での典型的な対話の流れは、図2のようになる。まず受付窓口のサービスを検索し、そのサービスの記入フォームを取り寄せ、記入した書類を提出する。

今回のシステムでは、wwHwwプロトコルはJavaの分散オブジェクト環境であるRMI (Java Remote Method Invocation) 上に構築している。

5. アプリケーション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



図5 受付窓口業務アプリケーションフレームワーク

を図5に示す。wwHwwブラウザは、ディレクトリブラウザとフォームブラウザと両者の共通部分であるwwHwwブラウザ基本部からなる。wwHwwサーバは、3種類のサービスモジュールとそれらの管理を行うwwHwwサーバ基本部からなる。トランザクションサービスでは、用意されたインタフェースを用いて業務ワークフローシステムとの連携が可能である。

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5のアプリケーションフレームワークには含まれない。

今回、Javaを使用して開発したアプリケーションのコードサイズは、受付窓口業務のフレームワークが19クラスで約1500ステップ、図書管理システム固有の部分が7クラスで約600ステップとなっている。

6. 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、今回のような図書管理システムを構築する場合の手順を次にあげる。

(1) サービスの定義

まず、受付窓口で提供するサービスについて定義する。図書管理システムでは、5つのサービスを提供しており、それぞれについて窓口の名前や窓口の検索の際に利用するキーワードやカテゴリ分けなどの情報を定義する。

(2) フォームの作成

それぞれのサービスに必要な記入項目を定義し、その記入を支援するナビゲーション情報の設定を行う。記入フォームは、GUIによるフォーム作成ツールを利用して作成する。図書管理システムでは、5つのサービスに対応した5種類の記入フォームが必要となる。ナビゲーション機能には、記入内容チェック、オンラインヘルプ、エージェントによる自動記入などがある。

(3) 書類の処理方式の設定

提出された書類の処理方式を設定する。書類の処理方式には、業務ワークフローシステムなどの業務アプリケーションに受け渡す方法、データベースへ蓄積する方法、紙に印刷する方法などが考えられる。業務の専門家による業務アプリケーション

の開発方法としては、M-base[5]等のエンドユーザ向け開発環境を利用した開発を想定している。

(4) サーバへの登録

最後に(1)から(3)で定義した内容をwwHwwサーバのディレクトリサービス、フォームサービス、トランザクションサービスに登録する。

7. おわりに

今回、例題として研究室内での図書管理システムを開発し、そのソフトウェアアーキテクチャに基づいて受付窓口業務アプリケーションフレームワークの基本部分を確立した。また、それを用いて業務の専門家が自ら窓口業務を自動化するための方式について述べた。

今後は、業務の専門家および一般の窓口利用者という2種類のエンドユーザの視点での使い勝手を向上させるツール群の開発、および多組織間オフィスネットワークシステムの視点でのディレクトリサービスの分散管理方式やサーバ及びクライアントの認証機能について検討していく。

参考文献

- [1] 中所武司：ソフトウェア工学、朝倉書店 (1997)
- [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, pp.33-40 (1994)
- [3] 藤原克哉、斎藤裕樹、中所武司：多組織間オフィスネットワークにおける分散アプリケーションフレームワークwwHwwのアーキテクチャと例題への適用、情報処理学会ソフトウェア工学研究会資料97-SE-115-9, pp.65-72 (1997)
- [4] M. E. Fayad and D. C. Schmidt (Ed.) : Object-Oriented Application Frameworks, Commun. ACM,

40, 10, pp.32-87 (1997)

[5]T. Chusho, Y. Konishi and M. Yoshioka : M-base
: An Application Development Environment for End-
user Computing based on Message Flow, APSEC'
96, IEEE Computer Society, pp.366-375 (1996)

プロフィール



中所 武司 (ちゅうしょ たけし)

1946年 丸亀市生まれ

1969年 東京大学電子工学科卒

1971年 同大学院修士課程修了

1971年～1993年 (株)日立製作所システム開発研究所勤務

1993年から明治大学工学部情報科学科教授、現在に至る。工学博士 (東京大学)。1982年度情報処理学会論文賞、1986年度大河内記念技術賞受賞。著書「ソフトウェア工学」(朝倉書店)、「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)、「プログラミングツール」、「人工知能」(昭晃堂、共著)。情報処理学会、電子情報通信学会、日本ソフトウェア科学会、人工知能学会、IEEE、ACM各会員。