

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

藤原克哉[†] 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ともに、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。また、今日の情報システム構築においては、フレームワークやデザインパターン、コンポーネントウェアなどの構成要素から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。本研究では、業務の専門家が自らのアプリケーションの構築に利用できるような窓口業務のアプリケーションフレームワークを開発した。窓口業務の例題システムとして図書管理システムを構築し、窓口業務に共通の部分と個々の窓口依存する部分を明確に分離することにより、窓口業務アプリケーションフレームワークを抽出した。さらに、そのフレームワークを利用して業務の専門家がアプリケーションを構築する方法を確立し、実際に別のシステムに適用し、その評価を行った。

Development and Evaluation of an Application Framework of Window Work for End-users

KATSUYA FUJIWARA[†] and TAKESHI CHUSHO[†]

The number of end-users increases on the inside and outside of offices. This paper describes an application framework for window work in banks, city offices, travel agents, mail-order companies, etc. based on the philosophy of "All routine work both at office and at home should be carried out by computers." We developed the application framework of the window work which the business experts were able to use for building application. The window work application framework has been extracted from the library system which was developed as an example of the window work. Then, the framework was applied to another system, and was evaluated.

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、限られた人たちが利用してきた。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ともに、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。

今日の情報システム構築においては、従来の構造化技法に代表されるトップダウン技法からコンポーネントウェアの組合せ（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、コンポーネントウェア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^{1)~3)}が追求されている。

特にあらかじめ用意されたテスト済みコード（クラスライブラリ）を再利用するオブジェクト指向フレームワーク^{4)~6)}に関して、GUIやネットワークなどのシステムレベルのフレームワークがすでに実用になっている。最近では、特定の問題領域に特化したアプリケーションフレームワークが開発、利用されはじめており、企業の基幹業務を対象としたエンタプライズアプリケーションフレームワークが注目されている^{7)~9)}。

しかしながら、これらのフレームワークを用いたアプリケーション構築技法はシステムエンジニア主導によるアプリケーション構築を前提としており、その具体的な開発プロセス^{7),8)}にはプログラミング言語を用いたソフトウェアのコーディングが含まれるので、エンドユーザによる構築に適さない。本研究では、エンドユーザ主導で利用できるフレームワークが今後重要になると考え、業務の専門家によるフレームワークのカスタマイズ方式をはじめとした具体的なアプリケー

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

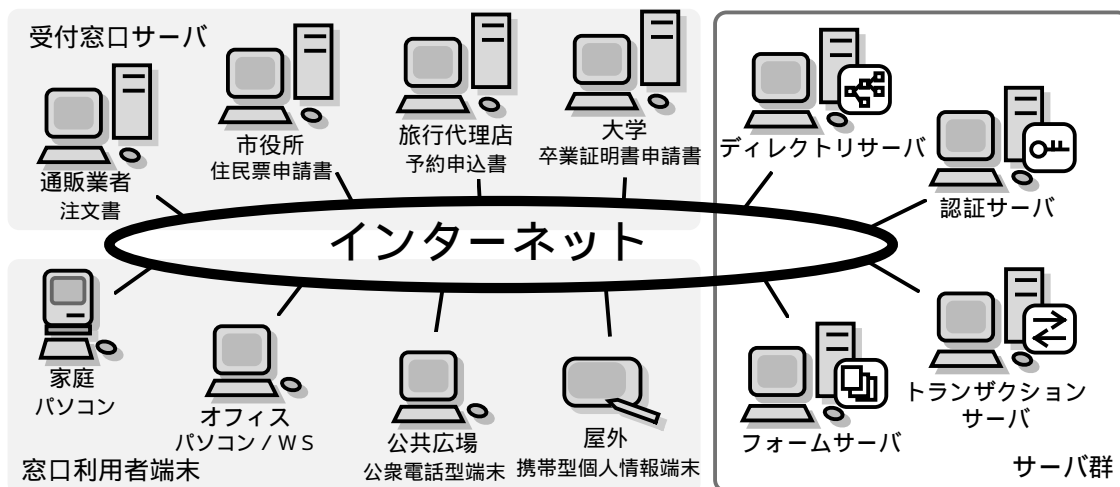


図 1 多組織間ネットワーク上の分散オフィスシステム MOON の構成例

Fig. 1 A MOON (multi-organizational office network) system.

ション構築技法¹⁰⁾の確立を目指す。

本論文では、業務の専門家が自らアプリケーションを構築する必要性の高い窓口業務のアプリケーションフレームワークを取り上げる。その例題システムとして図書管理システムを構築し、窓口業務に共通の部分と図書管理業務に固有の部分とを明確に分離することにより、窓口業務アプリケーションフレームワークを抽出する。さらに、そのフレームワークを利用して業務の専門家がアプリケーションを構築する方法を確立し、実際に別のシステムに適用し、その評価を行う。

2. 応用システムの概要

2.1 アプリケーションフレームワークの意義

本論文で取り上げるオフィスの窓口業務について、窓口利用者は直接窓口へ行く代わりに、自分のオフィスあるいは家庭のコンピュータから窓口への依頼を行えば便利である。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をチェックし、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、その分費用の低減になる。

このような窓口業務のアプリケーションは、WWWを利用したオンラインショッピングや銀行・証券取引、旅行予約などのシステムがすでにインターネット上に次々と実用化されている。これらのシステムは、申込みを受け付けるといった観点での窓口業務の処理に関しては、異なる業種間での共通性が高いが、個別に開発されているため開発コストが高く、実際に開発される

のはコストに見合った対象に限定されている。

窓口利用者にとっては、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、個別に操作方法を習得する必要がある。

そこで、窓口業務アプリケーションフレームワーク^{11),12)}を研究開発し、業務の専門家自身がそれを利用して自らの業務を電子化することにより、開発コストを削減できる。また、窓口利用者にとってはフレームワークで提供する統一されたユーザインタフェースにより操作性の問題が改善される。

2.2 応用システムの特徴

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。窓口利用者端末は、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。この窓口利用者もエンドユーザである。

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

図の窓口利用者端末では、以下のような共通のユーザインタフェースを提供することにより、窓口利用者にとっては、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。

- 受付窓口の問合せ(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 対話インタフェース

2.3.1 機能仕様

今回の応用システムでは、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。

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 同上またはその問合せ

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 同上またはその問合せ

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 同上またはその問合せ

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 同上またはその問合せ

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。

2.3.2 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。? は値の問合せを意味する。

(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 に記入する。入力形式は窓口利用者端末に依存する(以下同様)。

(2) 依頼先、業務種別、書式の問合せ例

(a, b, ?x,) 窓口 a に処理 b を依頼するための書式が表示され、入力が誘導される。表示形式は窓口利用者端末に依存する(以下同様)。

(a, ?x,) 窓口 a が担当する処理一覧が表示される。

(?x, b,) 処理 b を担当する窓口がすべて表示される。

(?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)。

(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。

3. 例題システムの構築

本研究では、まず窓口業務の例題システムを構築し、窓口業務に共通の部分と個々の窓口固有の部分とを明

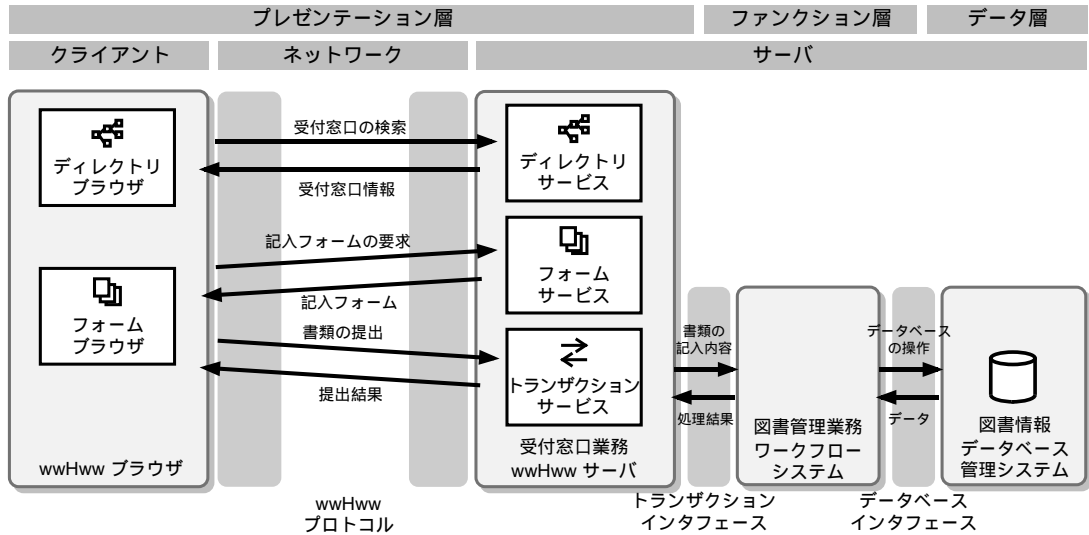


図 2 図書管理システムの構成
Fig. 2 Structure of library system.

確に分離することにより，窓口業務のアプリケーションフレームワークを抽出するというアプローチをとる．例題システムには身近な業務から研究室の図書管理業務を取り上げ，その一連の窓口業務を自動化した図書管理システムを開発した．

図書管理業務で提供する機能は，図書の登録・削除・貸出・返却・検索の 5 つである．検索機能はシステムの自動化により可能になった機能である．また，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．

3.1 設計方針

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 2.2 節で述べた多組織間ネットワーク環境を基に，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．

本システムは，マルチプラットフォームに対応するため 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みである．

3.2 クライアント側システム構成

窓口利用者による本システムの利用手順は以下のようになる．

- (1) 窓口と記入フォーム（書式）の検索
- (2) フォームの取り寄せ
- (3) フォームへの記入

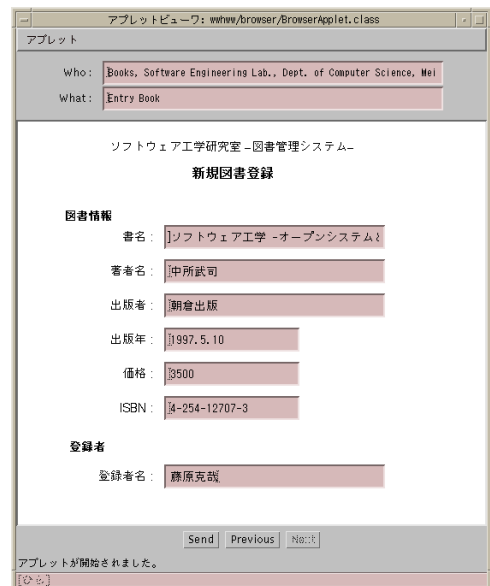


図 3 フォームブラウザの使用例

Fig. 3 An example of the wwHww browser at a client.

(4) 記入した書類の提出

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，手順 1 を支援するディレクトリブラウザと手順 2, 3, 4 を支援するフォームブラウザからなる．図 3 にフォームブラウザの使用例を示す．

これらのブラウザは，WWW ブラウザで利用するための Java アプレット版と，応答性を重視した Java アプリケーション版の 2 種類を構築した．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。アプリケーション版はプログラムをインストールしておく必要があるが、応答性が向上するためオフィス内や自宅から利用する場合に便利である。

3.3 サーバ側システム構成

図書管理システムのサーバ側システム構成は、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。3 層クライアントサーバシステムの観点では、各々、プレゼンテーション層、ファンクション層、データ層に相当する。

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は、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り、窓口業務単体のシステムも構築できる。しかし、実用上は今回の例のように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、先述の利用手順 1, 2, 4 に対応する次の 3 種類の要求を受け付ける。

- (1) 窓口とフォームの検索
- (2) フォームの取り寄せ
- (3) 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 つのサブシステムからなる構成とした。

- 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
- 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
- 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

3.4 共通プロトコル

インターネット上で 2.3 節で定義された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。クライアントからサーバへの 3 種類のメッセージに対応する具体的な wwHww プロトコルの例を以下に示す。

- (1) 受付窓口検索のプロトコル
(図書管理 , ?x, ,)
 - (2) 記入フォーム取り寄せのプロトコル
(図書管理 , 図書貸出申請 , ?x,)
 - (3) 書類提出のプロトコル
(図書管理 , 図書貸出申請 , { 記入内容 } , x)
- (1) は、2.3.2 項 (2) の (a, ?x,) に対応するもので、「図書管理」窓口が担当する処理一覧を要求するメッセージである。(2) は、2.3.2 項 (2) の (a, b, ?x,) に対応するもので、「図書管理」窓口へ「図書貸出申請」フォームを要求するメッセージである。(3) は、2.3.2 項 (1) に対応するもので、「図書管理」窓口へ「図書貸出申請」の書類を提出するメッセージである。

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。

4. 窓口業務フレームワークの抽出

4.1 フレームワークの構成

フレームワークは、ある問題領域のソフトウェアの設計と実装の再利用を目的としている。協調動作するクラス群として与えられる問題領域に共通なコードをあらかじめフレームワークとして用意しておくことで、アプリケーション開発者は、個々のアプリケーションに固有の機能のみを開発すればよい。

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あたるの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窓口業務アプリケーション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 4 に示す。wwHww ブラウザは、ディレクトリブ

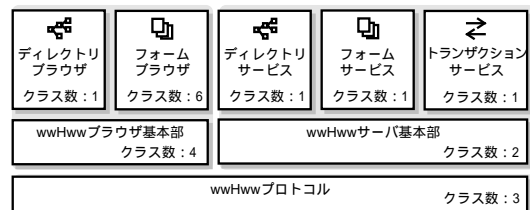


図 4 受付窓口業務アプリケーションフレームワーク

Fig. 4 The wwHww application framework.

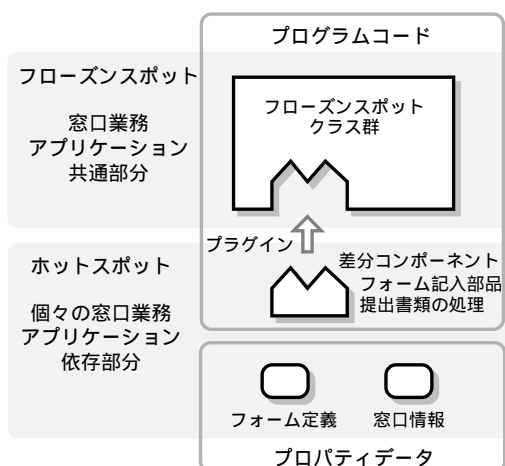


図5 窓口業務フレームワークの2種類のカスタマイズ方法
Fig. 5 Two types of customization on hot spots.

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。

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、窓口業務アプリケーションが 22 クラスで約 1700 ステップ、図書管理の業務ワークフローシステムが 7 クラスで約 600 ステップとなっている。

4.2 フレームワークのカスタマイズ方式

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¹⁾。アプリケーション開発者は、個々のアプリケーションに固有の処理をホットスポットに定義することで、フレームワークをカスタマイズしてアプリケーションを構築する。

フレームワークのカスタマイズ方式は以下の2種類に分類できる。その概要を図5に示す。

- (1) 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を定義するために、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る。
- (2) 差分コンポーネントのプラグイン アプリケーションに固有の機能をフレームワークの指定したインタフェースに沿って用意し、フレームワークにプラグインする。

(1)の方式は業務の専門家による構築に適しているが、この方式のみでは適用範囲がフレームワークで用意された機能に限定され、柔軟性に欠ける。

(2)でプラグインする差分コンポーネントは、基本的にアプリケーション開発者が既存のコンポーネントから個々の要求に適したコンポーネントを探し出すことになる。アプリケーションに適したコンポーネントが見つからない場合は、新たに開発する必要がある。本研究では、エンドユーザ主導によるアプリケーション開発を目的としているため、(2)の方式の場合は必要なコンポーネントが用意されていることが望ましい。

図書管理システムの場合は、業務の専門家がカスタマイズすべき固有の処理部分として、受付窓口の説明などの記述からなる窓口情報、電子フォーム、電子フォーム用の記入部品、提出書類の処理方式がある。このうち窓口情報と電子フォームの定義はプロパティデータとして用意すればよい。また、電子フォーム用の記入部品と提出書類の処理方式は差分コンポーネントをプラグインする。本システムでは、よく使われるフォーム部品と提出書類の処理コンポーネントをあらかじめ用意しておくことで、業務の専門家はその組合せにより個々の業務に固有の処理を構築することを想定している。

今回開発した窓口業務アプリケーションの 22 クラスのうち、ホットスポットにプラグインするコンポーネントが 3 クラスで約 200 ステップ、フローズンスポットが 19 クラスで約 1500 ステップとなっている。

4.3 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、今回の図書管理システムのような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

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、図書の登録・削除・貸出・返却・検索の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と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。

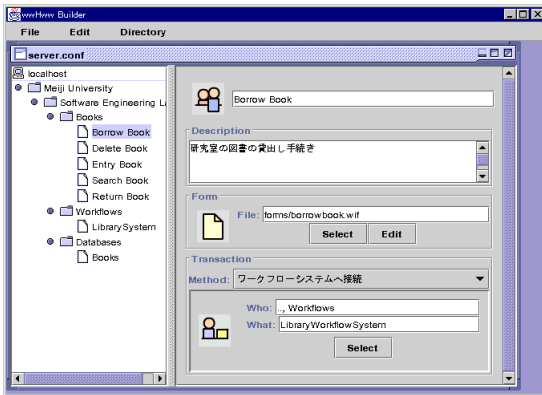


図 6 窓口業務アプリケーションの構築ツール画面

Fig.6 An example of the wwHww builder for domain experts.

(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、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。

(4) サーバへの登録

最後に上記の (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。

登録までの一連の手順は、図 6 に示す統一された構築ツールで支援する。

5. 他の例題への適用

このフレームワークを利用して、業務の専門家が窓口業務アプリケーションが作れることを確認するために、実際に他の例題システムへ適用した。例題には、以下の 3 つのシステムを選択した。

- (1) 研究室内の備品管理システム
- (2) 学科の会議室予約システム
- (3) ゼミ欠席届システム

備品管理システムは図書管理システムと物の貸出しの管理という機能仕様がほぼ同じ例である。まず類似のシステムでフレームワークが再利用できることを確認する。

会議室予約システムは記入フォームが複雑な例である。フォーム部品の種類が増えた場合の業務の専門家によるカスタマイズ方式を検証する。

ゼミ欠席届システムは業務ワークフローシステムとの連携がない例である。業務ワークフローシステムが

ない場合の提出書類の処理方式のカスタマイズ方式を検証する。

各々のシステムの構築は、前述の業務の専門家による構築手順に沿って以下のように行った。

5.1 備品管理システム

備品管理システムは、研究室の備品の管理とその貸出の管理を行う。図 2 に示す図書管理システムと同様に、窓口業務システムと業務ワークフローシステム、データベース管理システムの 3 つのサブシステムから構成される。

(1) サービスの定義

備品管理の受付窓口では、備品の登録・削除・貸出・返却・検索の 5 つのサービスを提供する。ここでは各々のサービスについて、サービスの名前や提供するサービスの説明などの窓口業務フレームワークが指定した項目を埋めていく形でプロパティデータを作成した。

(2) フォームの作成

次に、サービスごとの記入フォームについて、窓口利用者が記入する項目の設定と、その項目を配置したフォーム全体のレイアウトの定義を行った。

(3) 書類の処理方式の設定

備品管理システムでは、提出された書類の処理をワークフローシステムで行う。フレームワークで指定したインタフェースに沿って備品管理ワークフローシステムを構築した。

(4) サーバへの登録

備品管理の受付窓口は、図書管理システムと同じ受付窓口サーバに追加する形で登録した。

備品管理システムの構築では、フレームワークのコードに変更を加えることなく、窓口業務アプリケーションが構築できることを確認した。

5.2 会議室予約システム

学科内の会議室の予約は、現状では、掲示板に張り出してある会議室ごとの月間予定表に利用希望者が名前を記入することになっている。このシステムでは、この機能を電子化する。

(1) サービスの定義

会議室予約システムの受付窓口では、会議室の予約・予約のキャンセル・予約状況の表示の 3 つのサービスを提供する。

(2) フォームの作成

次に、各々のサービスに必要な記入項目を考えながら記入フォームを作成した。会議室の予約は、会議室名、予約期間、予約者、利用目的を指定して行う。予約期間の時間指定は、学校の授業時間

表 1 例題システムのプログラムサイズ
Table 1 Program size of example systems.

例題システム	窓口業務アプリケーション				業務ワークフローシステム	
	フローズスポット		ホットスポット		クラス数	ステップ数
	クラス数	ステップ数	クラス数	ステップ数		
図書管理システム	19	1468 (86%)	3	235 (14%)	7	353
備品管理システム	19	1468 (86%)	3	235 (14%)	7	367
会議室予約システム	19	1468 (70%)	6	615 (30%)	7	779
ゼミ欠席届システム	19	1468 (86%)	3	246 (14%)	-	-

単位で行う。ここでは日付専用の記入部品を新規に開発した。図書管理システムと備品管理システムでは、日付を記入する目的でテキストの記入部品を利用していたが、会議室予約システムでは日付の扱いが重要になるので、窓口利用者による記入を支援し、記入ミスを防ぐ目的で専用の部品を利用した。

(3) 書類の処理方式の設定

会議室予約システムも備品管理システムと同様に、提出された書類の処理をワークフローシステムで行う。

(4) サーバへの登録

会議室予約システムの受付窓口は学科に所属するので、備品管理窓口を登録した研究室の上位階層にあたる学科のディレクトリに登録した。

会議室予約システムでは、多くの種類のフォーム部品を利用したため、表 1 に示すようにフレームワークに占めるフローズスポットの比率が約 70%と最も低くなった。

5.3 ゼミ欠席届システム

ゼミナールの欠席届を先生に提出するシステムである。

(1) サービスの定義

ゼミ欠席届システムの受付窓口で提供するサービスは、ゼミ欠席手続きのみのシンプルな構成である。

(2) フォームの作成

次に、欠席届の記入フォームを作成した。欠席届には、欠席するゼミの日時、欠席者名、欠席理由を記入することにする。

(3) 書類の処理方式の設定

提出された書類は、電子メールで転送する。電子メールを選んだのは、勤務中も外出中もチェックができるなど、現状で最も確実な伝達手段であるからである。この電子メールで送信するコンポーネントは、フレームワークで指定したインタフェースに沿って新たに開発した。

(4) サーバへの登録

ゼミ欠席届システムの受付窓口は、在庫管理システムと同じ研究室の窓口サーバに登録した。

他の例題システムが、提出書類の処理を業務ワークフローシステムと連携して処理していたのとは違い、欠席届システムは、提出書類を電子メールで送信するのみで、複雑な業務ロジックがない。このため、他の例題が図 2 に示すような 3 層 C/S 型のシステム構成なのに対し、このシステムは窓口業務のみから構成される。

6. 評価

例題への適用結果を基にフレームワークの有効性を確認するために、以下の 3 つの評価基準について考察する。

- (1) フローズスポットとホットスポットの比率
- (2) 業務の専門家によるカスタマイズ方式
- (3) カスタマイズの作業量

6.1 フローズスポットとホットスポットの比率

図書管理システムを含めた 4 つの例題システムのプログラムサイズを表 1 に示す。今回適用した例題システムでは、窓口業務アプリケーション全体に占めるフローズスポットの割合は 70~86%、ホットスポットの占める割合は 14~30%である。ホットスポットのほとんどは、電子フォームの記入部品が占めている。

フローズスポットの 19 クラスの詳細は図 4 に示したとおりである。ステップ数で示すと、wwHww ブラウザ関連が 1114 ステップ、wwHww サーバ関連が 258 ステップ、wwHww プロトコル処理部が 96 ステップである。

小・中規模の業務アプリケーションを対象としたサンフランシスコフレームワーク^{8),9)}は、基本的なビジネス・ロジックおよび分散オブジェクトなどを提供する業務アプリケーションフレームワークである。このフレームワークのサポートするドメインの典型的なアプリケーションでは、全体の約 40%をフレームワークがカバーし、アプリケーション開発者は残りの 60%を開発している。フレームワークに占めるフローズスポットの割合が少ない理由としては、アプリケーション

ン開発者の用意するコードに、プログラムコードが大きくなりがちなユーザインタフェースが含まれることと、フレームワークの適用範囲を広く設定していることがあげられる。

窓口業務システムのフローズスポットの割合が大きい理由としては、業務の専門家による構築の作業量を少なくするために、問題領域を窓口業務に限定していることがあげられる。

例題システムの中で、会議室予約システムに占めるホットスポットの割合が比較的に大きくなった理由としては、記入フォームで利用したフォーム部品の種類が多かったことがあげられる。

6.2 業務の専門家によるカスタマイズ方式

3つの例題システムへの適用にあたっては、いずれも窓口業務フレームワークのフローズスポットはそのまま利用し手を加える必要はなかった。業務の専門家は4.2節で述べた2種類のカスタマイズ方式により、フレームワークのホットスポットに個々のアプリケーションに固有の処理を定義することで、アプリケーションを構築する。

差分コンポーネントのプラグインによるカスタマイズ方式において、アプリケーションの要求に適したコンポーネントが見つからない場合は、新たに開発する必要がある。しかしこのコンポーネントの新規開発は、業務の専門家による構築に適さないので、よく使われるコンポーネントをフレームワークとともに用意する必要がある。

今回の例題システムでは、会議室予約システムの日付記入部品と、ゼミ欠席届システムの電子メール送信コンポーネントを新たに開発する必要があった。しかし、これらのコンポーネントは、会議室予約システムやゼミ欠席届システムに限らず、他の窓口業務アプリケーションにも共通に利用できるものである。したがって、これらのコンポーネントは本来あらかじめ用意しておくべきコンポーネントであるといえる。

ただ、すべての用途に適應するコンポーネントをあらかじめ用意することはできないので、特別な機能が必要な際にはシステムエンジニアの支援による新規開発が必要となる。

6.3 カスタマイズの作業量

前述のように、ホットスポットの差分コンポーネントとして既存のコンポーネントを利用できる場合、業務の専門家にとっては、プログラムサイズの比率よりも、コンポーネントの選択とプロパティデータを設定する作業量が重要になる。

例題システムの各窓口の記入フォーム数を表2に示

表2 例題システムの各窓口の記入フォーム数
Table 2 Entry forms on example systems.

例題システム	フォーム数	部品総数	部品の種類
図書管理システム	5	108	2
備品管理システム	5	108	2
会議室予約システム	3	23	5
ゼミ欠席届システム	1	8	2

す。実際に、プロパティデータの設定作業量は、ホットスポットの比率の高い会議室予約システムよりも、フォーム数とその部品総数の多い備品管理システムの方が多かった。

6.4 その他の課題

手順(3)で設定する提出書類の処理方式では、欠席届システム以外の3つのシステムの書類の処理を業務ワークフローシステムとして自動化している。業務ワークフローシステムは本フレームワークの対象ではないが、実際に窓口業務の自動化を考える際には、このように提出書類の処理も自動化したい場合が多いと思われる。

このようなワークフローシステムの開発方法としては、本研究における窓口業務の場合と同様に業務ワークフローシステムを対象ドメインとしたフレームワークを利用することが考えられる。ただし、業務ワークフローシステムのような対象ドメインの広いフレームワークでは、ホットスポットが多くなるものと思われる。

7. ま と め

今回、例題として図書管理システムを開発し、窓口業務アプリケーションフレームワークを抽出した。そして、そのフレームワークを実際に他のシステムに適用することで、業務の専門家が窓口業務アプリケーションを構築できることを確認した。また、2種類のホットスポットのカスタマイズ方式を検証し評価を行った。

その結果、業務の専門家によるアプリケーション構築においては、ホットスポットのプログラムサイズの比率よりも、プロパティデータの設定などのホットスポットの作業量を少なくすることが重要であることを示した。

今後は、ホットスポットのカスタマイズをさらに容易にするためのコンポーネントモデルの研究を行っていく。

謝辞 本研究の一部はEAGL事業推進機構の支援を受けて実施したものである。

参 考 文 献

- 1) Pree, W.: *Design Patterns for Object-Oriented Software Development*, ACM Press, (1995). 佐藤, 金澤 (訳): デザインパターンプログラミング, トッパン (1996).
- 2) Johnson, R.E.: Frameworks = (Components + Patterns), *Comm. ACM*, Vol.40, No.10, pp.39-42 (1997).
- 3) Mellor, S.J. and Johnson, R.: Why Explore Object Methods, Patterns, and Architectures?, *IEEE Software*, Vol.14, No.1, pp.27-30 (1997).
- 4) Sparks, S., Benner, K. and Faris, C.: Managing Object-Oriented Framework Reuse, *IEEE Computer*, Vol.29, No.9, pp.52-61 (1996).
- 5) Islam, N.: Customizing system software using OO framework, *IEEE Computer*, Vol.30, No.2, pp.69-78 (1997).
- 6) Fayad, M. and Schmidt, D.C.: Object-oriented application frameworks, *Comm. ACM*, Vol.40, No.10, pp.32-38 (1997).
- 7) Jacobson, I., Ericsson, M. and Jacobson, A.: The Object Advantage, *Business Process Reengineering with Object Technology*, ACM Press (1995). 本位田 (訳): ビジネスオブジェクトユースケースによる企業変革, トッパン (1996).
- 8) Bohrer, K., Johnson, V., Nilsson, A. and Rubin, B.: Business Process Components for Distributed Object Applications, *Comm. ACM*, Vol.41, No.6, pp.43-48 (1998).
- 9) 吉田武稔, 志村 功, 田中一郎: ビジネスオブジェクトとその実現のための基盤技術, 情報処理, Vol.39, No.2, pp.81-85 (1998).
- 10) 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).
- 11) 藤原克哉, 斉藤裕樹, 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).

- 12) 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 の開発と適用評価, 情報処理学会オブジェクト指向'98 シン

ポジウム, オブジェクト指向最前線'98, pp.116-123, 朝倉書店 (1998).

(平成 11 年 3 月 10 日受付)

(平成 12 年 1 月 6 日採録)



藤原 克哉 (学生会員)

1974 年生. 1999 年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博士前期課程修了. 現在同大学院博士後期課程在学中. オブジェクト指向分析・設計技法, アプリケー

ションフレームワークに興味を持つ. 日本ソフトウェア科学会会員.



中所 武司 (正会員)

1946 年生. 1969 年東京大学工学部電子工学科卒業. 1971 年同大学院修士課程修了. 同年 (株) 日立製作所入社. 同社システム開発研究所主任研究員を経て, 1993 年から明治大

学理工学部情報科学科教授, 現在に至る. ソフトウェア工学の研究に従事. コンポーネントベースのアプリケーション開発方法論に関心を持つ. 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) 等. 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 Computer Society, ACM 各会員.