

多組織間ネットワークにおけるエンドユーザ向き 通信方式とソフトウェア構成法の研究

中 所 武 司 明治大学理工学部情報科学科教授

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、「すべての日常の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」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。

本研究では、 m 種類の異なる組織と n 種類の異なるエンドユーザ端末の間でやり取りされる電子書類に関する $m \cdot n$ 種類のユーザインタフェースと通信プロトコルを共通化することにより、そのために開発すべきアプリケーションソフトウェアの数を $m \cdot n$ から $m+n$ に減じると共に、その電子書類を送付する一般のユーザのインタフェースを簡単にし、それを受ける取る業務の専門家の処理を自動化することを目的とする。

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。

このような観点から、本研究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 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

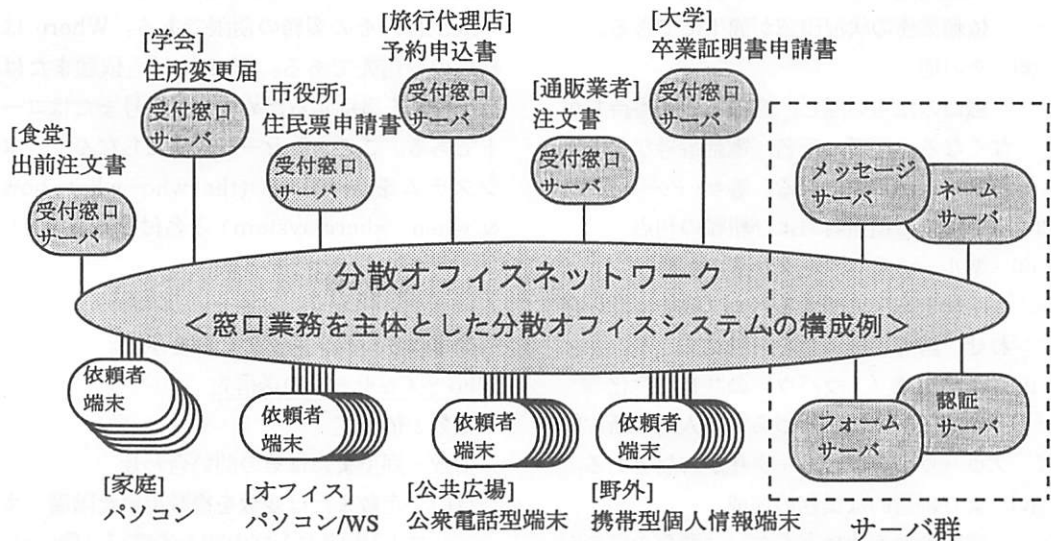


図1 多組織間ネットワークにおける分散オフィスシステムの例

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（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。

2.3 特長

このような応用システム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
業務の効率化にとめない、業務の質の向上により多くの時間をかけられる。

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について、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。

3.2 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。

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。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 (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。

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。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) また記号を使用する。

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, 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。)

(?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。

4. クライアントサイド wwHww システム

MOON システム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を図2に示す。ここで扱う電子フォームは、マシンやユーザインタフェースに依存しない表現にする必要があったので、テキスト形式の帳票定義言語 FDL を開発した。この FDL で記述された電子フォームはサーバーに格納され、クライアント側から要求されるとそのままの形で転送する。

ここでは、以下のシナリオを用いてクライアントサイド wwHww システムの処理方式を説明する。

(シナリオ)

クライアント側の初期画面として、Who, What, How, When, Where, Which の欄を含むウィンドウ (WIF) が開かれている。エンドユーザが Who の欄に、“学生事務室”、What の欄に、“在学証明書申込書”と記入し、How の欄をダブルクリックすることで、学生事務室の Form Server に在学証明書申込書を要求する。WIF に申込書が表示されると、必要事項を記入し、提出 (送信) する。

以下では、サーバから取り寄せた電子フォームがクライアント側に到着した後の処理を、図3に沿って説明する。

(1) 電子フォームの取り寄せ

サーバ側から取り寄せた電子フォームは、wwHww Protocol Sender が受け取り、そのまま FDL コンパイラに渡す。FDL コンパイラは、この電子フォームの字句解析/構文解析を行ない、内部表現 (木構造) 形式に変換し、それを wwHww Model に渡す。wwHww Model は、この内部表現形式の電子フォームを格納する。この内部表現形式 (UI 非依存) のデータは、wwHww UIMS によって UI 依存のデータに変換され、WIF に渡される。WIF はこのデータに基づき、電子フォームを表示する。

(2) 電子フォームの記入

エンドユーザは、電子フォームの“氏名”、

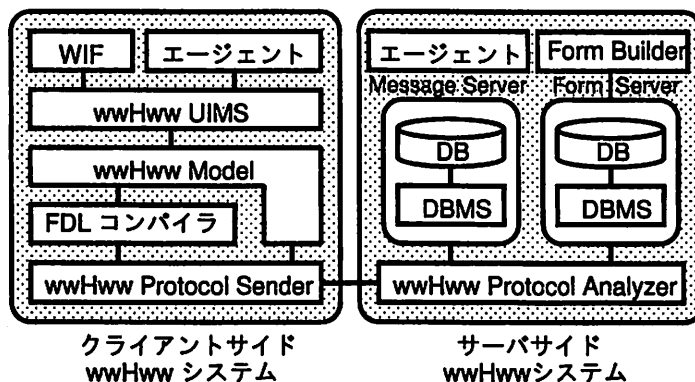


図2 wwHww システムの構成

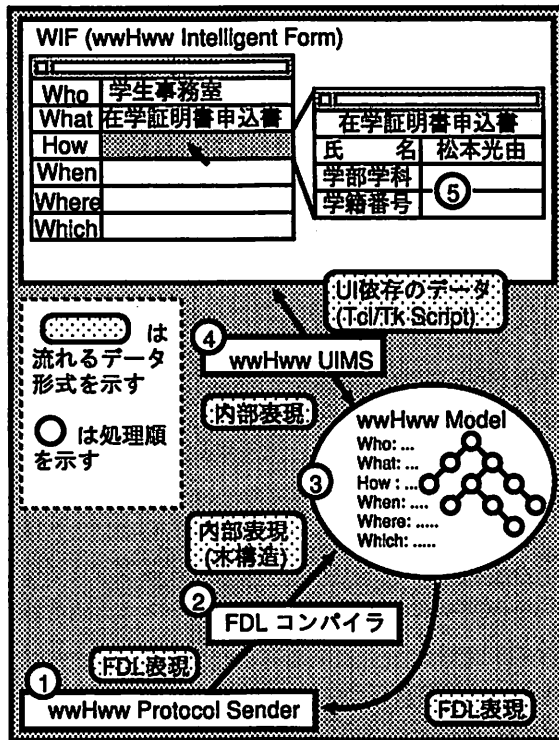


図3 クライアント側の電子フォームの処理

“学部学科”，“学籍番号”の各欄の記入を行なうことにより How の欄の入力を終了すると、必要に応じて When, Where の欄にも記入して、提出する。

(3) 電子フォームの提出

電子フォームの提出の処理は、取り寄せとは逆順で行なわれる。但し、wwHww Model が内部表現から FDL への変換を行なう。

クライアントサイド wwHww システムは、エージェント部分を除いて、Sun ワークステーション上で試作した。クライアント/サーバ間の通信は RPC により実現した。WIF が扱う UI 依存データには、Tcl/Tk スクリプトを用いた。FDL コンパイラの実装には、lex/yacc を、wwHww UIMS 及び wwHww Model の実装には、オブジェクト指向言語 C++ を用いた。

5. サーバサイド wwHww システム

次に、図2のサーバサイド wwHww システムについて説明する。

(1) 受付処理

wwHww Protocol Analyzer は、一般の依頼者に対して、その依頼に必要となる申込書を渡したり、記入された書類を受け取りたりする処理に対応するもので、クライアント側からの要求を解析する役割を担う。

(2) 申込書用引き出し

Form Server は、扱っている業務に必要な申込書を入れてある引き出しに対応するもので、申込書を蓄えたデータベースである。

(3) 記入済み書類用トレイ

Message Server は、依頼者から提出された書類を入れておくためのトレイに対応するもので、記入済み書類を蓄えておくデータベースである。

(4) 申込書作成

Form Builder は、業務に必要な申込書を作成するためのツールである。

ここで、具体的な応用システムの例として「図書管理システム」を挙げる。このシステムは、研究室内の図書を管理するためのアプリケーションである。具体的な機能としては、一覧表示、貸出、返却、立替購入、検索、登録、削除、および購入依頼がある。研究室内の図書を借りる時に行なう。「貸出処理」の実行例を図4に示す。このように、「一覧表示」以外の機能に関しては、その処理に必要な申込書をウィンドウに表示し、それに記入することによって依頼するようになるので、これらの処理部分に wwHww システムを適用した。

その結果、以下のような wwHww システムの機能が有効であることを確認した。

- (1) 入力方法に関しては、図4の例ではキーボードからの打ち込みのみであるが、入力可能項目のメニュー選択方式も可能である。
- (2) 実社会での業務の専門家は一般の依頼者から、申込書の記入に対して同じような質問を何度も受けるが、このようなよく受ける質問に対する回答も、ヘルプ機能として申込書に組み入れることができる。
- (3) 頻繁に記入する情報、例えば氏名、住所、生年月日等はエージェントによって自動記入させることも可能である。図4では、「貸出日」の欄が自動記入されている。

(4) これらのナビゲーション機能は、業務の専門家が From Builder によって申込書を作成する際に埋め込むことができる。

6. 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 を開発した。

大学での各種証明書の申請、書籍の通信販売の申込、自治体の各種登録、などの身近な例を参考に、基本的な通信プロトコルとユーザインタフェースの方式を決定し、アプリケーションフレームワークの設計と試作を終えた。

Message Server に蓄えられた記入済み書類の利用方法は業務によって異なるが、ワークフローシステムなどの業務システムと連携するようなケースが増えるものと思われる。

今後は、多組織間共通プロトコル詳細化とその処理プログラムの開発を行う予定である。

参考文献

- (1) Booch,G.: Object-Oriented Design with Applications, Benjamin/Cummings (1991).
- (2) 中所武司：使いやすいソフトウェアと作り

図書貸出依頼		
図書名	情報処理ハンドブック	
著者名	情報処理学会	
出版社	オーム社	
貸出日	1996/1/12	
依頼	クリア	キャンセル

図4 図書管理システムへの適用例
(貸出処理の申込書)

- やすいソフトウェアオブジェクト指向概念とその応用一, 電気学会雑誌, 110, 6, 465-472 (1990)。
- (3) 中所武司: エンドユーザコンピューティング, 情報処理, 32, 8, 950-960 (1991)。
- (4) 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)。
- (5) Coad, P. and Yourdon, E.: Object-Oriented Design, Prentice Hall (1991).
- (6) Fichman, R.G. and Kemerer, CF.: Object-Oriented and Conventional Analysis and Design methodologies, IEEE Computer, 25, 10, 22-39 (1992).
- (7) 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).
- (8) Lai, K., Malone, T., and Yu, K.: Object Lens: A "spreadsheet" for cooperative work, ACM Trans. Office Information Systems, 6, 4, 332-353 (1988).
- (9) Malone, T., Lai, K. and Fry, C.: Experiments with Oval: A radically tailorable tool for cooperative work, Proc. CSCW92, 289-297 (1992).
- (10) Monarchi, D.E. and Puhr, G.I.: A Research Typology for Object-Oriented Analysis and Design, Comm. ACM, 35, 9, 35-47 (1992).
- (11) Rumbaugh, J. et al.: Object-Oriented Modeling and Design, Prentice Hall (1991).
- (12) 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)。
- (13) Shaler, S. and Mellor, S.J.: Object-Oriented Systems Analysis: Modeling the World in Data, Prentice Hall (1988).

〈発表資料〉

題名	掲載誌学会名等	発表年月
wwHww: An Application Framework for End-User Computing in Multi-organizational Office Network Systems	The 12th annual conf. of Japan Society for Software Science and Technology, B8-3	1995年9月
wwHww:分散協調型アプリケーションフレームワーク-電子帳票の分散処理方式-	情報処理学会第52回大会論文集, 7Bb-6	1996年3月
wwHww:分散協調型アプリケーションフレームワーク-応用システムへの適用-	情報処理学会第52回大会論文集, 7Bb-7	1996年3月
wwHww: An Application Framework of Multi-organizational Office Network Systems	Memoirs of the Institute of Science and Technology, Meiji University, Vol.35, No.1	1996年6月
多組織間オフィスネットワークシステムにおける分散アプリケーションフレームワーク wwHww	「利用者指向の情報システム」シンポジウム論文集, 情報処理学会	1996年12月

電気通信普及財団研究調査報告書 No. 11

平成9年2月24日<非売品>

編集発行 財団法人 電気通信普及財団
〒105 東京都港区西新橋1-6-11
西新橋光和ビル5F
電話 03-3580-3411

印刷 至誠堂印刷株式会社

(記事の無断転載をしないでください。)