

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—

藤原克哉† 中所武司†

本研究では、全ての日常的仕事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を例題としたアプリケーションフレームワークを開発している。本稿では、現在インターネット上に実用化されている窓口業務システムを対象に、窓口利用者のための知的ナビゲーション機能を実現する方式を取り上げる。具体的には、ナビゲーションに必要な窓口のメタデータを定義する XML ベースの記述言語を開発し、これらの定義を管理するサーバエージェントと、ナビゲーション機能を実行するユーザエージェントからなるフレームワークを提供する。ユーザエージェントは、XML ベースの共通プロトコルによりサーバエージェントと協調してナビゲーション機能を実現する。

Development of XML-based Navigation Services 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.

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、全ての日常的仕事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を例題としたアプリケーションフレームワーク^{4)~6)}を開発している。

本研究の対象とする窓口業務アプリケーションの典型的な利用手順は以下のようなものである。

- (1) 窓口とフォームの検索
- (2) フォームへの記入
- (3) 書類の提出と処理状況の確認

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、旅行予約等のシステムが既にインターネット上に次々と実用化されているが、(1) について、現在の WWW システムで用いられている HTML は、書類の表示レイアウトのための形式であり、窓口やその記入フォームに限った検索等の意味情報に基づく検索はできない。窓口利用者のためにこれらの多組織にわたる窓口を縦断して検索するための仕組みが必要である。そこでこのような窓口の意味情報の定義のために、コンピュータの処理しやすい XML ベースの記述言語を設計した。

(2) のフォームへの記入においては、窓口利用者は名前や住所など同じような項目の入力を求められることが多い。また、記入フォームを提出した際に記入の不備を指摘されて記入し直す場面も多い。これらを自動的に記入したりチェックする機能があれば便利である。これらの実現のためには何を記入するのかと言った意味情報の定義が必要になる。ここでも (1) と同様に自動記入などの意味情報の定義のための XML ベースの記述言語を設計した。

(3) について、現状のシステムでは処理を依頼した

† 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

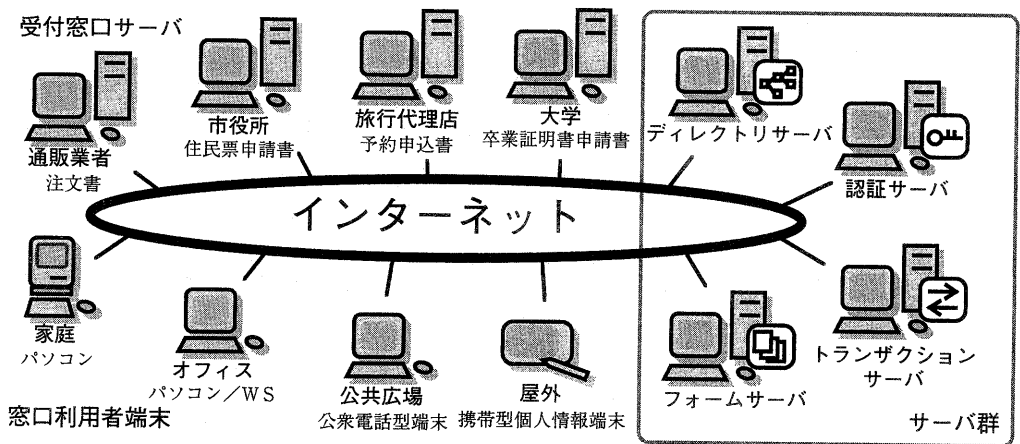


図1 多組織間ネットワーク上の分散オフィスシステム MOON の構成例
 Fig. 1 A MOON(multi-organizational office network) system

きりでその処理状況を知る手段がない場合が多い。依頼後に処理状況を確認する統一された方式があれば便利である。そこでこれらの問い合わせに対応する共通のトランザクションインタフェースを設計した。

本システムでは窓口の利用手順に対応するこれらの3つのナビゲーション機能を実現する。

また、これらの機能を実現するための窓口利用者側のクライアントと受付窓口側のサーバの通信プロトコルとして、XML ベースのメッセージ記述言語による共通プロトコルを設計した。最近では、インターネットの電子商取引分野で、XML ベースのメッセージ記述言語を用いたインターネット EDI¹⁾ が注目されている。しかし、これらは主にカタログ等の情報交換やトランザクションの定義に重点が置かれており、従来の WWW システムとの位置づけが明確ではない。また、既存の WWW システムからこれらのシステムに移行するためには、アプリケーションの再構築が必要になる。本システムでは、(1) と (2) については既存の WWW システムにメタデータ定義を追加することで、エージェントによる自動化やナビゲーション機能を実現するためのフレームワークを提供する。なお、(3) の機能はアプリケーションが対応している場合のみ利用できる。

2. 応用システムの概要

2.1 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、一つのブラウザから「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。

- 窓口の問い合わせ(検索)
- 窓口からのフォーム取り寄せとフォームへの記入
- 記入済書類の窓口への提出

2.2 対話インタフェース

2.2.1 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。

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 同上またはその問い合わせ

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

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。

2.2.2 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。?は値の問い合わせを意味する。

A. 業務依頼の例

- (1) (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) (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。
- (3) (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。

B. 依頼先、業務種別、書式の問い合わせ例

- (1) (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)。
- (2) (a, ?x,,) 窓口 a が担当する処理一覧が表示される。
- (3) (?x, b,,) 処理 b を担当する窓口がすべて表示される。
- (4) (?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)。
- (5) (?a,,) 窓口 a の業務内容が説明される。
- (6) (a, ?b,,) 窓口 a が担当する処理 b の内容が説明される。

3. 知的ナビゲーション機能の実現方式

3.1 例題システムの構築

本研究では、ナビゲーション機能を評価するために、まず窓口業務の例題システムを構築した。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。

例題システムの全体構成を図2に示す。例題システムは2.1で述べた多組織間ネットワーク環境を元に、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。

クライアント側の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) フォームの取寄せとフォームへの記入
- (3) 記入した書類の提出と依頼処理の追跡

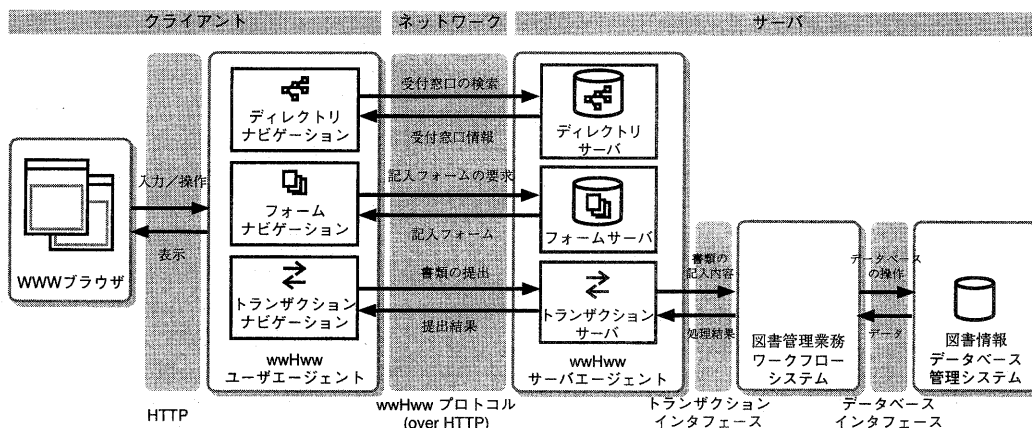


図 2 例題システムの構成

Fig. 2 Structure of library system

ユーザエージェントは、この利用手順を支援する3つのナビゲーション機能からなる構成とした。

- ディレクトリナビゲーション
- フォームナビゲーション
- トランザクションナビゲーション

受付窓口側の wwHww サーバは、3種類の窓口利用者からの問合わせに対応する以下の3つのサブシステムからなる構成とした。

- ディレクトリサーバ
- フォームサーバ
- トランザクションサーバ

インターネット上で 2.2 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。窓口の利用手順に対応する 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。

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、Solaris7, Windows98, 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。ユーザエージェントは WWW ブラウザをプレゼンテーション層とした WWW アプリケーションとして構築した。これにより、窓口利用者は一般の WWW ブラウザから本システムを利用できる。

3.2 窓口検索方式 (ディレクトリナビゲーション)

3.2.1 要求機能

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り。

現状のアプリケーションでは、窓口利用者がインターネットを経由して申込書を提出する場合、その窓口を

呼び出すために、URL によるネットワークリソース表記やハイパーリンクによる検索、全文検索を行うことになる。

URL を直接入力する場合は、表記法がわかりにくく窓口利用者にとって意味を持たないホスト名やプロトコル名を扱わなければならない。

ハイパーリンクを利用する場合、リンク構造が複雑で目的のところに簡単にたどり着けないことが多い。

現状の WWW 全文検索システムは、窓口と記入フォームに限定した検索はできない。また、検索条件を工夫しなければ件数が多くなり、目的の窓口を見つけるのが難しくなる。

以上のような窓口依頼者側の問題を解決するとともに、業務の専門家が自ら管理、保守できるようなサービス管理方式が必要となる。

3.2.2 ディレクトリモデル

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。ディレクトリの構成例を図 3 に示す。図の明治大学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。

なお、このように組織内で行われる業務とそれを提供する部署の階層構造をディレクトリモデルに適用できるのは、イントラネットの場合である。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、より概念的なサービスのカテゴリ分けを考慮する。

3.2.3 窓口のメタデータ定義方式

業務の担当者は、担当する窓口の、所属部署、窓口名、検索用のキーワード、サービスの説明、サービスの

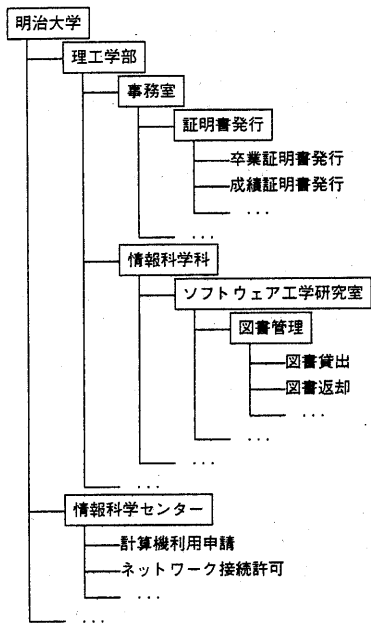


図3 組織の階層構造に基づくディレクトリの構成例

位置情報（物理アドレス）を定義した窓口のメタデータを受付窓口サーバに登録する。

このような窓口のメタデータを受付窓口の WWW サイトに用意することで、組織の階層構造に基づく検索や、キーワードによる検索などの多様な窓口検索が可能になる。

窓口のメタデータの記述には、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)²⁾ 形式の記述言語を用いる。RDF は、WWW で提供される情報に関する情報（メタデータ）を記述するための XML ベースの記述言語である。

窓口のメタデータの記述例の一部を以下に示す。

```

<!DOCTYPE RDF [
  <ENTITY site 'http://inside.se.cs.meiji.ac.jp'>
]>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:d="http://purl.org/dc/elements/1.1/"
  xmlns:w="http://wwhw.org/1.0/">
  <w:Agent rdf:about="&site;/library/">
    <d:Title>図書管理</d:Title>
    <d:Description>ソフトウェア工学研究室の図書管理システム</d:Description>
    <d:Text>ソフトウェア工学研究室の図書管理システム.&lt;br&gt;
    図書の研究室外への持ち出し/返却手続き等を行う（自己申告制）。</d:Text>
    <w:name>/明治大学/情報科学科/ソフトウェア工学研究室/図書管理</w:name>
  
```

```

<w:service rdf:resource="&site;/library/takeout"/>
<w:service rdf:resource="&site;/library/return"/>
<w:service rdf:resource="&site;/library/entry"/>
<w:service rdf:resource="&site;/library/modify"/>
<w:service rdf:resource="&site;/library/search"/>
</w:Agent>
<w:Form rdf:about="&site;/library/takeout">
  <d:Title>図書貸出</d:Title>
  <d:Description>図書の研究室外への持ち出しのための貸出し手続き</d:Description>
</w:Form>
</rdf:RDF>
  
```

w:Agent 項目内が「ソフトウェア工学研究室の図書管理」窓口のメタデータ定義である。w:Form 項目内がその窓口で提供している「図書貸出」サービスの定義である。

ディレクトリサーバは、窓口のメタデータを定期的に収集し分散管理するとともに、ユーザエージェントからの問い合わせに対応する。

3.2.4 ディレクトリ関連の共通プロトコル

クライアントからディレクトリサーバへ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) キーワード検索のプロトコル
(?x, ?y="図書", ?x,,)
- (2) 窓口の処理一覧のプロトコル
(/明治大学/理工学部/情報科学科/ソフトウェア工学研究室/図書管理, ?x,,)
- (3) 窓口の業務内容の説明のプロトコル
(?/明治大学/理工学部/情報科学科/ソフトウェア工学研究室/図書管理,,)

(1) は、2.2.2 B(4) の(?x,?y="k",,) に対応するもので、キーワード「図書」に関連した窓口と処理を検索して表示する。(2) は、2.2.2 B(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。(3) は、2.2.2 B(5) の(?a,,) に対応するもので、「図書管理」窓口の業務内容の説明を要求するメッセージである。

ユーザエージェントによる「図書管理」窓口の詳細表示の例を図4に示す。図は(2)と(3)の結果を組み合わせたもので、業務内容の説明文、処理一覧、業務内容の詳細説明へのリンク等からなる。

3.3 記入支援方式(フォームナビゲーション)

3.3.1 要求機能

窓口利用者は、記入フォームに名前や住所、連絡先等の同じような項目の入力を求められることが多い。また、以前利用したフォームに記入する際は、再び同じ内容を記入することが多い。このような、決まりきった

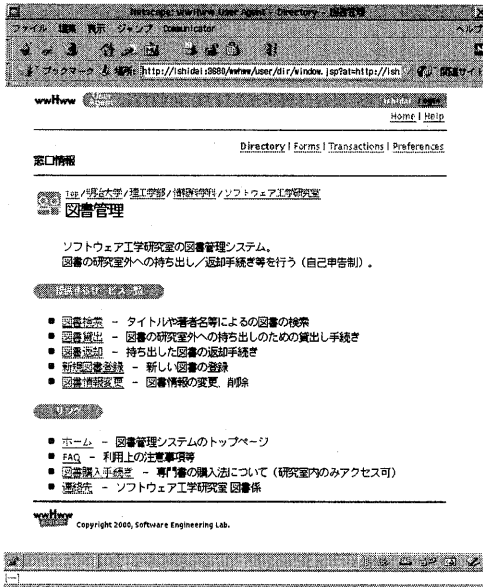


図 4 ディレクトリ検索結果の表示画面例

た単純作業をエージェントが支援し代行してくれれば便利である。

また、窓口利用者は、記入フォームを提出する際に記入の不備を指摘されて記入し直すような場面も多い。提出前のフォームへの記入時に、分かりやすく誘導してくれれば便利である。

本システムでは、前者の記入を代行する機能として、自動記入エージェントを導入する。また、後者を支援するために、オンラインヘルプ機能と記入チェック機能を導入する。

3.3.2 フォームモデルとフォーム記述方式

フォームへの自動記入のためには、フォームの各記入項目に何を記入すればよいかという、フォームの意味情報が必要になる。

現在、インターネット上のほとんどのフォームが HTML で記述されている。しかし、HTML 文書の書類の表示レイアウトのための言語であり、意味情報の定義には向いていない。HTML 文書は、人間は表示された文書を読んで理解できるが、コンピュータがその意味を理解するには複雑な自然言語処理が必要になる。

本システムでは、フォームの定義には従来の HTML を用い、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 (HTML と別に) 定義する。

フォームのメタデータの記述例の一部を以下に示す。
<RDF

```
xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:d="http://purl.org/dc/elements/1.1/"
xmlns:w="http://www.w3.org/1.0/">
<w:Form about="http://inside.se.cs.meiji.ac.jp/library/takeout/">
  <d:Title>図書貸出</d:Title>
  <d:Description>図書の研究室外への持ち出しのための貸出し手続き</d:Description>
  <w:input>
    <w:FormItem>
      <w:name>usr</w:name>
      <w:datatype resource="http://imc.org/vCard/3.0#FN"/>
      <w:value resource="urn:userprofile:@User.name.fullname"/>
      <w:help>http://inside.se.cs.meiji.ac.jp/library/takeout/help.html#usr</w:help>
    </w:FormItem>
  </w:input>
</w:Form>
</RDF>
```

w:Form 項目内が、図書管理フォームのメタデータ定義である。w:FormItem 項目内は、フォームの記入部品のメタデータ定義である。ここではこの記入部品について、w:datatype 項目では記入するデータの型が「氏名」であることを、w:value 項目では「利用者の氏名」を記入することを定義している。

ユーザエージェントはこのようにメタデータとして定義された自動記入や記入チェックのナビゲーション機能について、それを実際に実行する JavaScript 形式のロジックを自動生成し HTML フォームに埋め込む。生成されたフォームナビゲーションのロジックは窓口利用者の WWW ブラウザ上で実行される。

すでにインターネット上にあるフォームに、これらのフォームの意味定義を追加することで上記のフォームナビゲーション機能が実現できる。メタデータの定義は、業務の専門家がビジュアルな構築ツールを用いて自ら構築することを考えている。これらのナビゲーション機能により業務の専門家には、自動記入によって人為的な記入ミスがなくなり、より正確な情報が得られるメリットがある。

フォームナビゲーションの「自動記入」「自動記入チェック」「オンラインヘルプ」の3つの機能で定義するのは以下のようなものである

A. フォームへの自動記入

本システムで提供するフォームへの自動記入機能は、以下の3つに分類できる。

- (1) 窓口利用者(クライアント)側の情報の自動記入
住所、氏名などの窓口利用者のプロフィール情報をユーザエージェントに学習させて、自動記入を行う。

なお、自動記入では個人情報扱うためプライ

パシーの保護に配慮する必要がある。本システムでは、記入する内容をユーザエージェントがあらかじめ窓口利用者に明示し、確認する。

- (2) 窓口（サーバ）側の情報の自動記入
サーバ上にある情報を取寄せて記入する。
- (3) 他の記入内容と連動した自動記入
フォーム上の他のフォーム部品への記入内容に基づいて計算などを行い記入する。

B. 記入内容の自動チェック

フォームの記入内容チェックには、書類の提出前のクライアント側でのチェックと、提出後のサーバ側でのチェックと2種類考えられるが、ここで述べているものは、クライアント側での事前チェックである。

クライアント側記入内容チェックは大きく以下の3つに分類できる。

- (1) 個々部品の記入内容のチェック
書式が正しいか、値が正しいか。
- (2) 複数部品の記入内容のチェック
記入内容によって他の記入項目へ自動記入（値の計算など）や表示／非表示の変更を行う。
- (3) 送信時のチェック
記入漏れがないか。全体として記入の矛盾がないか。

C. オンラインヘルプ

個々の部品の記入例等を記述した文書をHTML形式で用意する。ヘルプは、ヘルプボタンによって呼び出され、ポップアップウインドウ上に表示する。

3.3.3 フォーム関連の共通プロトコル

クライアントからフォームサーバへのメッセージに対応する wwHww プロトコルは以下の通りである。

● フォーム取寄せのプロトコル

(/明治大学/理工学部/情報科学科/ソフトウェア工学研究室/図書管理, 図書貸出, ?x_i)

これは、2.2.2 B(1) の(a,b,?x_i)に対応するもので、「図書管理」窓口に「図書貸出」フォームを要求するメッセージである。

3.4 処理依頼支援方式（トランザクションナビゲーション）

3.4.1 要求機能

窓口利用者からの記入済み書類は、トランザクションサーバが受け付ける。さらに、処理依頼では書類を受け付けるだけでなく、即時に結果が出る処理を除いて、依頼した処理をキャンセルしたり、処理が終了したのか等の処理状況を確認できる仕組みがあれば便利である。

本システムでは、処理状況の確認と依頼内容の修正・

キャンセルのための、共通プロトコルと、バックエンドシステムとのインタフェースを用意する。

これにより、窓口利用者は、共通のインタフェースでユーザエージェントに処理が終了しているか定期的にチェックさせたり、終了した処理結果を自動的に取寄せることができる。

3.4.2 トランザクションインタフェースの概要

トランザクションサーバが受取った提出書類は、図2に示すようにトランザクションインタフェースを通して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処理される。

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るが、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。

3.4.3 トランザクション関連の共通プロトコル

クライアントからトランザクションサーバへ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) 書類提出のプロトコル

(/明治大学/理工学部/情報科学科/ソフトウェア工学研究室/図書管理, 図書貸出, { 記入内容 }, x)

(2) 処理キャンセルのプロトコル

(/明治大学/理工学部/情報科学科/ソフトウェア工学研究室/図書管理, 図書貸出,, -123)

(1)は、2.2.2 A(1)に対応するもので、「図書管理」窓口へ「図書貸出」の書類を提出するメッセージである。結果として、受付番号が返ってくる。(2)は、2.2.2 A(3)に対応するもので、「図書管理」窓口の「図書貸出」に提出した、受付番号123の書類の処理をキャンセルするメッセージである。

3.5 XMLを用いた wwHww メッセージ記述方式

(Who, What, How, Which)の基本形式で示される窓口利用者側のクライアントと受付窓口側のサーバの通信プロトコル(wwhwwプロトコル)の実現のために、XMLベースのRDF形式のメッセージ記述言語を開発した。そのメッセージのネットワーク転送プロトコルには、既存のWWWシステムとの相互運用性の確保のため、インターネット標準のHTTPを用いる。

以下に、3.2.4 (2)に対応する「図書管理」窓口の担当する処理一覧を要求するメッセージの記述例を示す。

<rdf:RDF

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

```

xmlns:w="http://wwhww.org/1.0/">
<w:Message rdf:about="">
  <w:who>/明治大学/理工学部/情報科学科/ソフトウェア工学研究
室/図書管理</w:who>
  <w:what><w:query/></w:what>
</w:Message>
</rdf:RDF>

```

このXML 文書中の w:who, w:what の各項目が、wwHww プロトコルの who, what の各パラメタに対応する。

窓口に書類を提出するメッセージの文書型は業務の専門家が設定する。例えば、XML を用いた代表的なメッセージ記述言語の SOAP を設定した場合、3.4.3 (1) に対応する「図書管理」窓口に「図書貸出」の申請書類を提出するメッセージは以下のように記述できる。

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlso
ap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/so
ap/encoding/">
  <SOAP-ENV:Header>
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#"
      xmlns:w="http://wwhww.org/1.0/">
      <w:Message rdf:about=""
        w:who="/明治大学/理工学部/情報科学科/ソフトウェア工学研
究室/図書管理"
        w:what="図書貸出">
        <w:how rdf:resource="" />
      <w:which/>
    </rdf:RDF>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <takeout
      xmlns="http://wwhww.org/schemas/lifecycle/takeout">
      <itemid>19991220</itemid>
      <user>fujiiwara</user>
    </takeout>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

文書中の SOAP-ENV:Envelope 項目が wwHww プロトコルの how パラメタに対応する。how 以外の who や what のパラメタは、SOAP-ENV:Header の rdf:RDF 項目内に定義している。このように本システムの wwHww メッセージ記述言語は他の XML 文書内や HTML 文書内に埋め込むことができる。なお、業務の専門家はこのような埋め込み方式以外に、必要に応じて先述の窓口の処理一覧メッセージの記述例のような w:Message の w:how 項目内に how パラメタを記述する方式も選択できる。ただしこの場合、w:how 項目内に記述できるのは XML 形式のデータに限られるため、HTML 文書を利用する場合等にはその HTML 文書内に wwHww メッセージ記述を埋め込む方式が適している。

4. ま と め

今回、例題として図書管理システムを開発し、窓口業務システムの知的ナビゲーション機能を構築した。

ダイレクトリナビゲーションでは、窓口利用の前段階である窓口利用者による窓口の検索を支援した。フォームナビゲーションでは、窓口から取寄せた記入フォームへの記入という、窓口利用者にとって煩雑な作業を自動記入や記入チェック機能で支援した。トランザクションナビゲーションでは、記入済み書類の提出と、提出後の処理状況の追跡のための機能を提供した。

また、ダイレクトリナビゲーションとフォームナビゲーションについては、既存のアプリケーションを変更せずに、新たにメタデータを追加する形で用意することで本システムのナビゲーション機能が利用できるようになることを示した。

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、および窓口利用者によるエージェントのパーソナライズ手法や業務の専門家によるアプリケーション構築法の研究を行っていく。

参 考 文 献

- 1) Robert J. Glushko, Jay M. Tenenbaum, and Bart Meltzer: An XML Framework for Agent-based E-commerce, Comm. ACM, 42, 3, 106-114 (1999).
- 2) Ora Lassila, Ralph R. Swick: Resource Description Framework (RDF) Model and Syntax, W3C Recommendation, World Wide Web Consortium (1999).
- 3) Simple Object Access Protocol (SOAP) 1.1 W3C Note, World Wide Web Consortium (2000).
- 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) 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, 朝倉書店, 116-123 (1998).
- 6) 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, 41, 4, 1202-1211 (2000).