

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*

斉藤 裕樹 松本 光由 安齋 恵 中所 武司†

明治大学 理工学部 情報科学科‡

{hiro-s, matumoto, satoshi, chusho}@cs.meiji.ac.jp

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。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、窓口業務の自動化を取り上げ、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アプリケーションフレームワーク wwHww を開発している。

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も高まっている [1]。

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、「コンピュータによる、より良い生活の実現」という意味での情報化を“CS-life”(Computer-Supported Life) と呼ぶ。

すなわち、「すべての日常的業務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。

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 [2] 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である。

1. オブジェクト指向技術のメッセージ機能を拡

*wwHww : An Application Framework For Multi-Organizational Office Network

†Hiroki SAITO, Mitsuyoshi MATSUMOTO, Satoshi ANZAI, Takeshi CHUSHO

‡Department of Computer Science, Meiji University

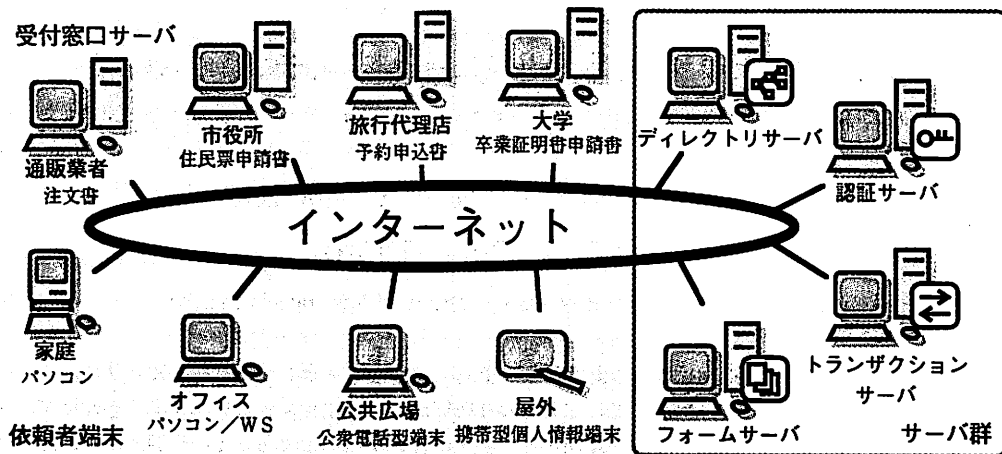


図 1: MOON システム

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(検索)に対応するサーバ側のサービス管理方式

既に、1, 2 のプロトプログラムの試作を終え、現在、インターネット / イン트라ネットをベースに 3, 4 の方式設計中である。

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。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。

- 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

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。

さらに、以下の3種類の業務を代行するエージェントを導入する。

- クライアント側の書類記入
- クライアント側が必要とする適切な窓口および関連書式の検索
- サーバ側からの書類記入誘導（ナビゲーション）

2.3 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、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

- (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) 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。

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。

- 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）。
- 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エンドユーザ（窓口への依頼者）が増大する（誰でも簡単に使えるようになる）。

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。

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) 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) 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

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。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。

現在のところ、標準的な端末としては WWW ブラウザ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ことも考えられる。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。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, Which)

また次のような記号を使用する。

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, 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。入力形式は依頼者側の
エージェントに依存する (以下同様)。

(a, b, c, 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 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。

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 ディレクトリ・サービス

本章では、クライアント側の多様な問い合わせと、
サーバ側のサービス管理を実現するディレクトリ・
サービスについて述べる。

4.1 要求機能

4.1.1 現状の問題点

現状のアプリケーションでは、窓口依頼者がイン
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[9] 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を行う
ことになる。

URL による場合は、表記法がわかりにくく意味
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。

4.1.2 必須機能

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。

そのために必要となる要件は次のとおりである。

(1) 実世界に対応した検索

業務を行う組織や部署の階層構造に基づく検索、
業務内容のキーワードを元にした検索、および、業
務種類別のカテゴリ分けに基づく検索を行えること。

(2) 業務専門家自身によるフォーム管理

業務専門家が作成したフォームを業務の専門家自
身が登録したり変更できること。

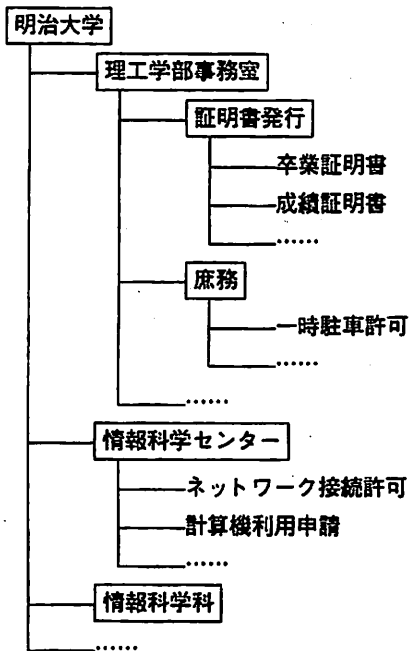


図 2: ディレクトリ・モデルに基づく名前空間

4.2 ディレクトリ・モデル

4.2.1 イン트라ネットのディレクトリ・モデル

このような必須機能を可能にするサービス管理方式を実現するために、名前空間のモデル(ディレクトリ・モデル)は、実世界の組織を反映したツリー構造とした。

ディレクトリ・モデルの例を図2に示す。最上位のノードはイントラネットとインターネットの境界になる部分であり、この名前空間は「明治大学」であるということを表している。「明治大学」以下は業務を行っている部署の階層構造であり、最下位は行われている業務である。

このモデルでは、例えば、現実の窓口利用での「理工学部事務局の証明書発行窓口で卒業証明書を申し込む」という概念と、図2のルートに当たる「明治大学」から「卒業証明書」までのパスが一致するので、エンドユーザ(窓口依頼者)にとってわかりやすい。

一方、エンドユーザ(業務の専門家)にとっても、「卒業証明書の保守管理は理工学部事務局にある証明書発行係」としての自分の業務がまとまった所に

あるのでわかりやすく、業務の専門家自身による保守管理が容易である。

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必要がある。

4.2.2 ディレクトリ・サーバ

ディレクトリ・モデルに則して各ノードにディレクトリ・サーバを配置する。ディレクトリ・サーバは直下のノードに対して定義を行う。

例えば、「明治大学」のディレクトリ・サーバは直下の「理工学部事務局」、「情報科学センター」、「情報科学科」、.....のノードとディレクトリ・サーバの存在を定義するのみで、さらに下位のノードについて定義する必要はない。「理工学部事務局」以下の構造は「理工学部事務局」のディレクトリ・サーバで定義することになる。

葉の「卒業証明書」は行われている業務を表すノードであり、このディレクトリ・サーバは業務「卒業証明書」の実際のネットワークアドレス、プロトコルなどを定義しておく。

このように、1つのディレクトリ・サーバの管理すべき範囲は、現実の部署や業務担当者の管理すべき範囲と一致していることになる。

4.3 検索機能

ここでは、先程と同様に図2での使用例を示すことにより検索機能を説明する。

なお、説明のために用いるパス表現は内部仕様であり、実際のユーザインタフェースはキーワードの誘導やメニュー選択などの検索支援方法を考慮する。

パス表現は“.”をディレクトリの区切りとし、“*”を任意の文字を表すワイルドカードとして表記している。

(1) 明治大学:理工学部事務局:*

この検索パターンは、明治大学の理工学部事務局の下にどのような窓口や業務が存在するか一覧表示させるものである。

検索結果は

明治大学:理工学部事務局:証明書発行

明治大学:理工学部事務局:庶務

である。

(2) 明治大学:*事務局*:~*証明*

この検索パターンは、事務室のどこかの窓口で行われている「証明」を含む業務を調べるものである。

検索結果は

明治大学:理工学部事務局:証明書発行:卒業証明

明治大学:理工学部事務局:証明書発行:成績証明

である。

(3) 明治大学:*~*駐車*

この検索パターンは、明治大学のどこかで行われている「駐車」を含む業務を調べるものである。
“*”は間に任意のディレクトリが何階層あっても適合するというワイルドカードである。この場合

明治大学:理工学部事務局:庶務:一時駐車許可

が検索結果である。

4.4 サービス管理方式

業務専門家はフォームの登録変更、担当部署の変更を管理マネージャを用いて行う。図3に示すように管理マネージャは業務専門家と各サーバ間に存在し、これらの作業を代行する。

フォームの登録変更を行う場合、管理マネージャはフォームをフォームサーバに登録するとともに、フォーム名、キーワードをディレクトリ・サーバに登録する。

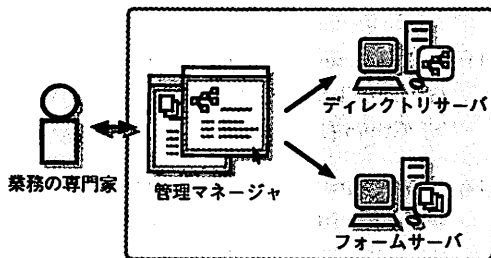


図3: サービス管理方式

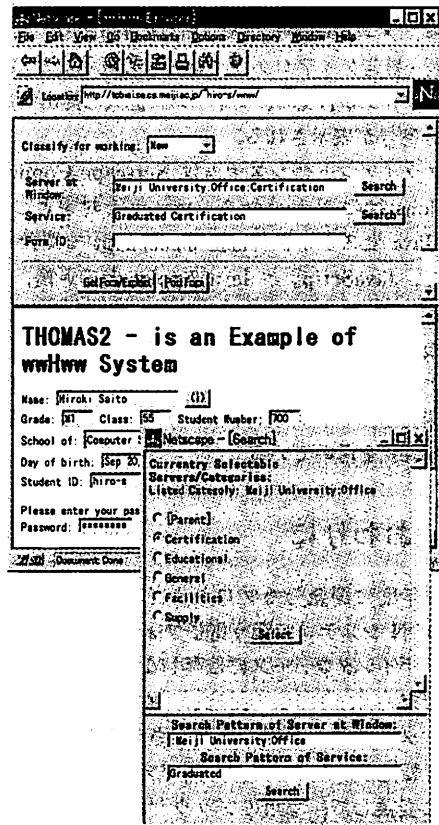


図4: 試作した wwHww プロトタイプ

また、担当部署の変更を行う場合は、新たな担当部署に該当する業務のディレクトリ・サーバの定義を作成し、旧部署から定義を削除する。

5 プロトタイプ試作

フォームの取り寄せと提出、訂正などの基本的かつ重要な機能に絞り、インターネット環境で動作するプロトタイプを試作した。

このプロトタイプでは WWW ブラウザを利用し、言語 Java で開発することにより、依頼者端末側のマルチプラットフォーム対応の問題を解決している。

図4は、検索ウィンドウで目的の窓口と業務を選択し、WWW ブラウザにフォームが表示されたところである。

フォームはハイパーテキスト形式の関連書式への誘導、メニュー選択、入力項目のチェック、および確認・警告ダイアログウィンドウの表示などのナビゲーションが行える。フォームに記入し提出すると、受付番号が表示される。

クライアント側システムは Netscape Communications 社の Netscape Navigator を対象とし、言語 Java とスクリプト言語 JavaScript で開発した。また、JavaScript と Java Applet との連携、および WWW ブラウザのコントロールを LiveConnect の技術を用いることで実現している。

サーバ側システムは Java アプリケーションとして開発し、トランザクション・サーバの DBMS に Oracle7 を用いている。

6 おわりに

昨年度に開発したプロトタイプ版 [6, 7] では、MOON システム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テキスト形式の帳票定義言語 FDL で記述し、サーバに格納しておき、クライアント側から要求されるとそのままの形で転送し、クライアント側でプラットフォーム対応に変換して、専用のブラウザに表示するようにした。

今年度は、インターネットと WWW ブラウザを標準的な環境としたアプリケーションフレームワーク wwHww を開発しているが、対象としている業務は、今のところ窓口への書類提出までである。しかしながら実際の適用段階では、トランザクションサーバに蓄えられた記入済み書類を媒介として、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。現在このような連携についても方式を検討している。

参考文献

- [1] 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).
- [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 (1994).
- [3] 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).
- [4] Lai, K., Malone, T., and Yu, K.: Object Lens: A "spreadsheet" for cooperative work, ACM Trans. Office Information Systems, 6, 4, 332-353 (1988).
- [5] Malone, T., Lai, K. and Fry, C.: Experiments with Oval: A radically tailorble tool for cooperative work, Proc. CSCW92, 289-297 (1992).
- [6] 安齋 恵, 松本 光由, 中所 武司: 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 回大会講演論文集 (1), pp.289-290 (1996).
- [7] 松本 光由, 安齋 恵, 中所 武司: 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 回大会講演論文集 (1), pp.287-288 (1996).
- [8] 斉藤 裕樹: 広域分散情報システムのためのアプリケーション向き資源管理フレームワーク, 明治大学理工学部情報科学科 1995 年度卒業論文 (1996).
- [9] T. Berners-Lee, L. Masinter, M. McCahill: Uniform Resource Locators(URL), RFC1738, Network Working Group (1994).
- [10] T. Berners-Lee, D. Connolly: Hypertext Markup Language — 2.0, RFC1866, Network Working Group (1995).
- [11] P. Mockapetris: Domain Names - Concepts and Facilities, RFC1034, Networking Working Group (1987).
- [12] P. Mockapetris: Domain Names - Implementation and Specification, RFC1035, Networking Working Group (1987).