

窓口業務アプリケーションフレームワーク wwHww における ルール生成を自動化した自動記入エージェントの実現方式

藤原克哉[†], 中所武司[†]

我々は、すべての日常的な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成し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の研究を行っている。本稿では、窓口利用者の電子フォームへの記入を支援する自動記入エージェントの実現方式について述べる。これまでに、自動記入のための記入ルールを定義し利用する自動記入方式を開発したが、記入ルールは業務の専門家が手動で定義する必要があった。そこで新たに、記入履歴から記入ルールを抽出する方式を確立することで、記入ルールの生成を自動化した。本システムは、自動記入と記入ルールの抽出を行うフォーム記入エージェントと、記入ルールを集計し管理するルール管理エージェントの 2 種類のエージェントからなる分散システムで Java と XML を用いて実現している。

Automation Method of Generating Rules for Filling in a Form on the Web Application Framework for End-users

KATSUYA FUJIWARA[†], and TAKESHI CHUSHO[†]

Enduser-initiative development of applications has become important for automation of their own task. As the solution based on the philosophy: "All routine work both at office and at home should be carried out by computers," this paper describes a agent which supports using web systems including window work in electronic commerce. We developed the agent for filling in a form for end-users. However, rules for filling in a form must be defined manually. Then, we developed the new system which generates a rule automatically, and the system was evaluated. This system is a Java application framework and includes a XML-based rule definition language.

1. はじめに

近年、パソコンやネットワークの普及とともにオフィス内外でエンドユーザが増加し、エンドユーザ主導によるシステム構築の必要性が高まっている。我々は、すべての日常的な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成し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の研究を行ってきた¹⁾。窓口業務のアプリケーションは、WWW (World Wide Web) 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

テムがすでにインターネットに実用化されている。また、最近では行政サービスの窓口を電子化する電子政府の実現が注目されている。

これらの既存の WWW システムにおいて、窓口利用者は、氏名や住所、メールアドレスなどの同じような項目の入力を求められることが多い。このような電子フォームへの記入作業を支援する自動記入機能がいくつか実現されている^{2)~6)}が、実際に自動記入できるフォームが少なかったり、記入を間違ったりするなどの問題がある。本研究では、記入内容から記入ルールを自動抽出し、その集計結果を用いて自動記入する方式を実現することで、これらの問題を解決する。

2. 従来の自動記入方式

現在利用されている自動記入方式を 3 つに分類した。各々の特徴は、以下のようなものである。
方式 A. 過去の記入内容の再記入 自動記入機能は、フォームの提出時に記入内容を保存し、次に同

[†] 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
現在、秋田大学工学資源学部情報工学科
Presently with Department of Computer Science and
Engineering, Faculty of Engineering and Resource Sci-
ence, Akita University

じフォームを利用する際に、前回の記入内容を自動記入する。

方式 B. 自動記入ルールの定義と利用 フォームに対して何を記入すればよいかというメタ情報を定義する。たとえば、フォーム X の項目 Y には「氏名」を記入するという自動記入ルールを定義する。自動記入機能は、その定義に基づき氏名を自動記入する。

方式 C. パターンマッチによる意味推論 フォーム中の文脈から、何を記入する場所なのかを推論する。たとえば、項目の近くに「名前」という単語がある(パターン)なら氏名を記入するというルールを定義しておく、あるフォームの記入項目の前に「お名前を入力してください:」と記されている場合「名前」という単語とこのルールから氏名を自動記入できる。

方式 A は、WWW ブラウザの Internet Explorer²⁾ や Netscape³⁾ で実用化されている。同じフォームには再度同じ内容を記入することが多いのでこの方式が有効であるが、初めて記入するフォームには適用できない。

方式 B を用いたものに Gator⁴⁾ や ShareStage⁵⁾ のサービスプロバイダがある。これらは記入ルールをサービスプロバイダが定義する。記入ルールを専門家が明示的に作成するため正確な記入ができるが、人手で作成するために対応するフォーム数が限られる。

方式 C を用いたものに Internet Explorer 5 Mac 版²⁾ や aiBAR2000⁶⁾、ShareStage がある。この方式は、適用可能なフォーム数が多いが、推論の精度に限界があり記入間違いが避けられない。Internet Explorer 5 の自動記入機能による氏名の記入調査⁷⁾ では、約 3 割が記入間違いか記入できないフォームであった。

3. wwHww における自動記入方式

上述のように、A、B は記入精度が優れているが、適用できるフォーム数が少ない。C は適用範囲が広いが、記入精度に劣る。自動記入機能には、記入精度が高く、適用範囲の広い方式が求められている。

wwHww システムでは、これらの 3 方式を組み合わせ、記入精度の高い方式から順に自動的に使い分けることで、記入精度が高く適用範囲の広い自動記入エージェントを実現する。3 つの方式は、記入精度の高い順に以下の優先順位で実行する。

- (1) 過去の記入内容の再記入(方式 A)
- (2) 自動記入ルールの定義と利用(方式 B)
- (3) パターンマッチによる意味推論(方式 C)

さらに各々の方式について、方式 B は適用範囲が広くなるように、方式 C は記入精度が高くなるように改良を行う。

方式 C については先述のような記入を間違える問題があるが、意味推論のルールを増やし、フォーム項目間の関連も判断材料にするとともにルールを実行時に変更可能にすることで記入精度を向上させる方式を研究中⁷⁾ である。

方式 B のような自動処理しやすい形式で意味情報を定義する Semantic Web 技術^{9),10)} は、自動記入をはじめとした処理の自動化・エージェント化に有効である。これまでに方式 B について、XML (Extensible Markup Language) を用いた記入ルールの定義方式を設計し、フォームナビゲーション機能の 1 つとして自動記入エージェントを開発した⁸⁾。しかしながら、これらの意味定義は人手で新たに用意する必要があり、対応するフォームが少ない。

本稿では、方式 B の欠点である記入ルールの手動定義の問題を解決する記入ルールの自動抽出方式について述べる。記入ルールが自動生成できれば、上記の問題は解決し、適用可能なフォーム数が大幅に増加する。

4. 記入ルール自動抽出による自動記入方式

4.1 実現方式の概要

本システムは、利用者端末でフォームへの記入を代行するフォーム記入ユーザエージェントと、ユーザエージェント間の対話を仲介するルール管理ブローカエージェントの 2 種類のエージェントから構成される。

本方式で自動記入するのは、多くのフォームに共通する氏名、住所、電話番号などの利用者の個人情報である。個人情報は、最初に一度、図 1 に示すような画面で、利用者からインタビューして利用者端末に保存しておく。本方式の主な特徴を以下に示す。

(1) 記入ルールの自動抽出

フォーム記入エージェントは、窓口利用者の各フォーム項目への記入内容を、利用者の個人情報と照合する。記入内容に対応する個人情報が見つかった場合、たとえばフォーム項目 A への記入内容が「太郎」で個人情報に「名は「太郎」とあれば「フォーム項目 A に名を記入した」という記入ルールが抽出できる。

(2) 記入ルールの集計

抽出した記入ルールは、ルール管理エージェントが収集する。ルール管理エージェントは、フォーム項目ごとに記入ルールを集計する。

(3) 記入ルールによる自動記入

フォーム記入エージェントは、記入ルールの集計結



図 1 自動記入用個人情報のインタビュー画面例
Fig.1 A profile registration window.

果を問い合わせ、最も記入例の多いルールを用いて、個人情報をフォームに自動記入する。この方式の記入精度は、どれだけ正しい記入ルールが集まるかによって決まる。少なくとも窓口利用者の半数以上がフォームに正確な内容を記入すれば、その記入履歴から抽出されたルールが最上位となり正確に自動記入できる。

なお、このような個人情報を扱うシステムにおいては、利用者が意図せずに個人情報が外部に渡らないように保護する必要がある。本システムでは、個人情報は利用者端末内に保存し管理する。ルール管理エージェントに送信される記入ルールには、個人情報の含まれた記入内容そのものは含まれず、「姓」を記入したなどの意味情報からなるため、個人情報の内容がフォーム記入エージェント外に流出することはない。また、ルール管理エージェントに送信される情報と、WWWサーバに提出される記入内容は、いつでも利用者が確認できるようにする。さらに、信頼できるルール管理エージェントを介することで、他のフォーム記入エージェントから IP アドレスや誰がどのフォームを利用したかという情報を隠蔽し保護する。

4.2 記入ルール定義形式の設計

エージェント間で交換する自動記入ルールは、どのフォーム項目にどのくらいの人が何をどのように記入したかという以下の 4 要素から構成される。

(1) 記入項目の識別子 (どこに) どのフォームのどの記入項目のルールかを示す。フォームの各記入

項目は、HTML フォームの URL、<FORM>項目の NAME 属性値 (省略時は通し番号)、<INPUT>項目などの NAME 属性値の 3 つからなる識別子によって表記する。

- (2) 個人情報の概念 (なにを) そこに記入した個人情報が何かを示す。個人情報の概念は概念名で識別され、ここではその概念名を記述する。自動記入可能な個人情報以外の記入があった場合は「不明」、記入が空の場合は「無記入」とする。
- (3) 記入書式 (どのように) 個人情報をどのような書式で記入したかを示す。大文字/小文字/カタカナ/全角文字への変換の有無と、項目間の区切り文字を記述する。
- (4) 累積ポイント数 (どの程度) ルール管理エージェントに、記入例としてこのルールが何件が集まっているかを示す。

(2) の自動記入を行う個人情報の概念名について、自動記入エージェントの共通概念となるオントロジを設計した。共通オントロジは、最小単位の基本項目と、基本項目の組合せによる複合項目の 2 種類に分類される。

基本項目は、日本語と英語表記の氏名の姓や名、住所の都道府県や市区町村、メールアドレスなどからなる 56 項目とした。基本項目一覧を表 1 に示す。また、複合項目は「姓名」や「都道府県 市区町村 番地」のような基本項目の組合せからなる約 100 項目とした。

(3) の記入書式について、記入項目には大文字や小文字での英字の記入を指定してあることがある。日本語圏のフォームでは、ひらがな/カタカナや全角/半角文字の区別がある。個人情報をこれらの書式に合わせて変換して自動記入することになる。ルール定義では以下の 6 種類の変換操作を定義する。

- (a) 英字 大文字
- (b) 英字 小文字
- (c) 半角文字 全角文字
- (d) ひらがな カタカナ
- (e) (a) と (c) の組合せ
- (f) (b) と (c) の組合せ

なお個人情報 DB には、かなひらがなで、英数字は半角でデータを登録しておく。これにより、(c) と (d) の逆変換を扱う必要がなくなる。

また記入書式には、複合項目の区切り文字を定義する。複合項目では、基本項目を続けて並べる場合と、項目間に空白や記号などの区切り文字を入れる場合があり、これらの区切り文字をここで記述する。

(4) のポイント数は、この記入例が何件のフォーム記入エージェントから集まっているかを 1 件 1 ポイ

表 1 記入用個人情報の基本項目
Table 1 Primitive items for user profile.

基本項目名	内容	
User.	Name.Last_ja	姓
	Name.First_ja	名
	Name.Last_ja_hira	姓(ふりがな)
	Name.First_ja_hira	名(ふりがな)
	Name.Last	姓(ローマ字)
	Name.Middle	ミドルネーム
	Name.First	名(ローマ字)
Name.Bdate	生年月日	
User.	Email	メールアドレス
	Address.Zip.Prefix	郵便番号(上位桁)
	Address.Zip.Suffix	郵便番号(下位桁)
	Address.State_ja	都道府県
	Address.City_ja	市区町村
	Address.Street_ja	番地
	Address.Street	番地(英語表記)
	Address.City	市区町村(英語表記)
	Address.State	都道府県(英語表記)
	Address.Country	国名
Home.	Phone.IntCode	電話番号(国番号)
	Phone.LocCode	電話番号(市外局番)
または	Phone.Prefix	電話番号(上位桁)
	Phone.Suffix	電話番号(下位桁)
User. Office.	Phone.Extension	電話番号(内線)
	Fax.IntCode	FAX 番号(国番号)
	Fax.LocCode	FAX 番号(市外局番)
	Fax.Prefix	FAX 番号(上位桁)
	Fax.Suffix	FAX 番号(下位桁)
	Fax.Extension	FAX 番号(内線)
	Mobile.IntCode	携帯番号(国番号)
	Mobile.LocCode	携帯番号(局番)
	Mobile.Prefix	携帯番号(上位桁)
	Mobile.Suffix	携帯番号(下位桁)

User.Home=自宅, User.Office=勤務先

ントとして加算したものである。ポイント数は、ルール管理エージェントが集計し管理する。フォーム記入エージェントはフォーム項目ごとに記入ルールを問い合わせるため、ルールはフォーム項目単位で管理する。この(1)のフォーム項目識別子が一致するルールの集合、つまり同じフォーム項目に対するルールの集合を項目別ルール群と呼ぶ。さらに、ある項目別ルール群の中で(2)の記入概念が同じルールの集合を概念別ルール群と呼ぶ。概念別ルール群は、ある1つのフォーム項目に(3)の記入書式は違っても同じ個人情報を記入するルールの集合である。

自動記入のためのルールの選択時に、まず項目別ルール群の中で最もポイント数の高い概念別ルール群1つに絞りこみ、記入する概念を確定する。そしてその概念別ルール群の中で最もポイント数の多いルールを選択し、記入する書式が確定する。これは、たとえば区切り文字について、フォームに区切り文字の指定がない氏名の記入項目には、「姓」と「名」の間を区

切りなしで記入するほかに、半角空白文字や全角空白文字で区切る利用者もいるが、どれも意味的には正しい。もし記入書式も区別したルール単位での集計結果を用いると、意味が同じものが別々にカウントされるため、本来上位の候補になるべきものが下位となり自動記入を誤ることがあり問題である。そこで先述のように先に概念別ルール群で絞りこむことでこの問題を解決する。

4.3 自動記入システムの実装

4.3.1 システム構成

分散協調型自動記入エージェントのシステム構成を図2に示す。窓口利用者端末でフォームへの自動記入と記入ルールの学習(抽出)を行うフォーム記入ユーザエージェントと、学習によって得られた記入ルールを集積してフォーム記入エージェント間の知識交換を仲介するルール管理ブローカエージェントの、2種類のエージェントからなる構成とした。また、ユーザエージェントとブローカエージェント間の対話のためのXMLベースのエージェント間通信言語 FACL(Form-based Agent Communication Language)³⁾を開発した。

フォーム記入エージェントは、利用者端末上で HTTP Proxy として動作する。WWW ブラウザと WWW サーバの通信はフォーム記入エージェント経由で行われる。フォーム記入エージェントは通信内容を観察し、取り寄せるデータがフォームの場合、自動記入機能を実行する。また、WWW ブラウザから WWW サーバへ記入内容の送信を行うときに、学習機能を実行する。

フォーム記入エージェントはプラットフォーム非依存の Java を用いて実装した。ほとんどの WWW ブラウザは HTTP Proxy に対応しているため、本システムはプラットフォームと WWW ブラウザに依存しない。

4.3.2 FACL プロトコルの概要

FACL 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分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した。

(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

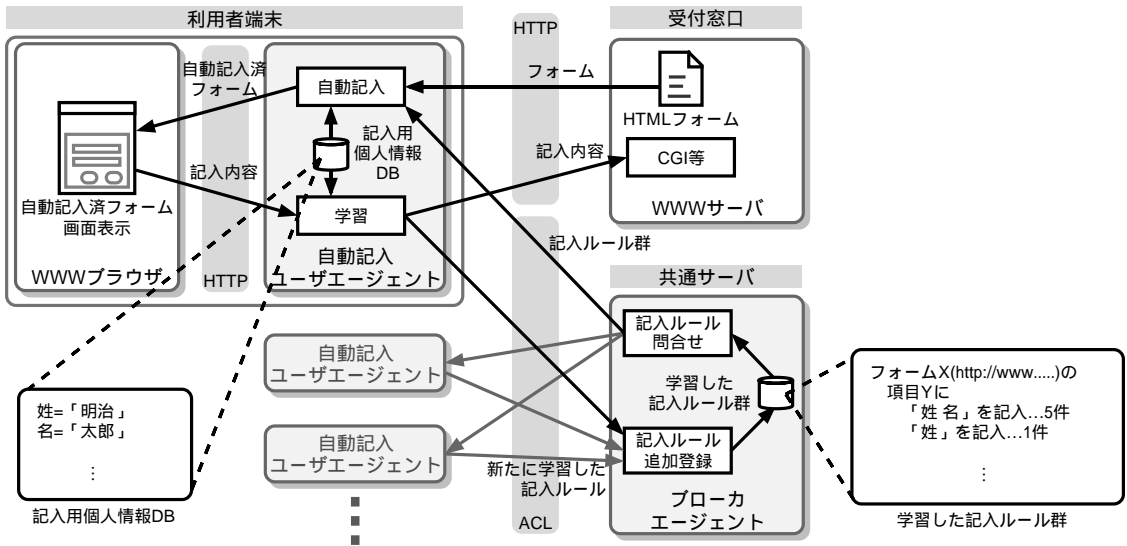


図 2 自動記入エージェントのアーキテクチャ

Fig. 2 Architecture of the agent for filling in a form.

FACL の詳細については文献 13) に詳しい。

ルール管理エージェントは、フォーム記入エージェントからの以下の 2 つの問合せに対応する。

- 記入ルールの集計結果の問合せ (検索)
- 新規記入ルールの追加登録

4.3.3 RDF ベースの記入ルール記述言語

記入ルールの外部記述のために、本システムでは、SemanticWeb 技術の 1 つである RDF (Resource Description Framework)^{11),12)} をベースとした、記入ルール記述言語を開発した。RDF は外部記述形式に XML を用いた WWW 向きのメタデータ定義方式である。HTML フォームへの記入方法を定義する記入ルールのように、HTML 文書を説明するメタデータを定義するのに適している。また RDF は、オブジェクト指向に基づくクラスの継承機能を用いてスキーマ (オントロジ) の拡張や再利用が可能になっている。この拡張機能を用いて、将来、個人情報のオントロジの追加が可能である。

記入ルール記述言語は以下のようなものである。

- <FormItem> : フォーム項目のメタデータ定義をこの要素内で行う。
 - rdf:about 属性: 前節 (1) のフォーム項目の識別子。
 - <history> : 記入履歴から自動抽出した記入ルールの内容を記述。
- <基本項目名> と <Profile> : 前節 (2) の個人情報の概念にあたる定義をこの要素内で行う。基本項目の場合、要素名を基本項目名とする。複合

項目の場合、要素名を Profile とする。

- format 属性: 前節 (3) の大文字/小文字/全角/カタカナへ変換する記入書式。
- amount 属性: 前節 (4) の累積ポイント数。
- <value> (複合項目の場合) : 前節 (2) 複合項目の構成要素の基本項目をこの要素内に記述。
- <separator> (複合項目の場合) : 前節 (2) 複合項目の構成要素の基本項目間の区切り文字。

HTML フォーム文書例と、そのフォーム項目の記入ルールの記述例の一部を図 3 に示す。FACL によるエージェント間の対話において、メソッドの引数 How に与える記入ルールの記述に、この記入ルール記述言語を用いる。この記入ルール記述は自動的に抽出し集計されるものでエンドユーザが直接意識する必要はない。

4.4 自動記入方式

フォーム記入エージェントは WWW サーバから取り寄せたデータがフォームの場合、以下の順に自動記入を実行する。

- (1) WWW サーバから取り寄せた HTML 文書にフォームが含まれている場合、まず、方式 A による自動記入を試みて、記入できなかった場合に以下を実行する。
- (2) ルール管理エージェントにフォームの URL を引数として記入ルールを問い合わせる。ルール管理エージェントは、先に方式 B の手動による記

```

<HTML>
<HEAD> ... </HEAD>
<BODY>
...
<FORM NAME="entry" ACTION="entry.cgi">
...
  Your name : <INPUT NAME="name">
    <INPUT TYPE="submit">
    <INPUT TYPE="reset">
  </FORM>
...
</BODY>
</HTML>

```

(a) HTML によるフォーム記述例の一部

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf=
"http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:o="http://www.org/schemas/userprofile/1.0/"
  xmlns:"http://www.org/1.0/">
<FormItem rdf:about="http://www.se.cs.meiji.ac.jp/libra
ry/entry/#form[entry].item[name]">
  <history>
    <Profile amount=10>
      <value><o:User.Name.First /></value>
      <separator> </separator>
      <value><o:User.Name.Last /></value>
    </Profile>
  </history>
  <history>
    <o:User.Name.Last
  format="http://www.org/datatype/en/1.0/#Lowercase"
  amount=1 />
  </history>
</FormItem>
</RDF>

```

(b) 記入ルール記述言語による記述例

図 3 記入ルールの外部記述例

Fig. 3 An example of the rule definition.

入ルール定義を検索し、見つかった場合はそのルールを送信する。ない場合は、自動抽出された記入ルール群の集計結果から該当するフォームの記入ルールを検索し結果を送信する。

- (3) (a) 方式 B の手動による記入ルールの場合、そのルールを用いてフォームの各記入項目に自動記入する。
- (b) 自動抽出されたルール群が見つかった場合、各々の記入項目について、その集計結果から利用者が設定した基準に基づき候補を 1 つ選択する。
- ルール候補が選択されたら、そのルールの基本/複合項目名をキーとして、記入用個人情報 DB からデータを取り出し、最も多く使用された書式(大/小文字、ひらがな/カタカナ、区切り文字)でフォー

```

<HTML>
<HEAD> ... </HEAD>
<BODY>
...
<FORM NAME="entry" METHOD="POST"
ACTION="http://localhost:3680/proxy.jsp">
...
  Your name : <INPUT NAME="name" VALUE="Taro Meiji">
    <INPUT TYPE="submit">
    <INPUT TYPE="reset">
  <!-- \ / -->
  <INPUT TYPE="hidden" NAME="___form" VALUE=
"http://www.se.cs.meiji.ac.jp/library/entry/">
  <INPUT TYPE="hidden" NAME="___action" VALUE=
"http://www.se.cs.meiji.ac.jp/library/entry/entry.cgi">
  <INPUT TYPE="hidden" NAME="___method" VALUE="GET">
  <!-- / \ -->
  </FORM>
...
</BODY>
</HTML>

```

図 4 自動記入済みの HTML 文書例

Fig. 4 An example of the filled HTML form.

ムの各記入項目に自動記入する。

- (c) 誰も記入したことがないフォームの場合や基準を満たすルール候補がない場合は、方式 C による自動記入を行う。

- (4) 記入済みのフォームを WWW ブラウザに渡す。窓口利用者は、自動記入されていない残りの項目に記入する。また、自動記入された内容に間違いがあれば修正して記入する。

フォーム記入エージェントによるフォームへの自動記入は、WWW サーバから取り寄せた HTML 文書を書き換えることによって行う。図 3(1) の HTML フォームに自動記入を行い書き換えた結果の HTML 文書を図 4 に示す。

フォームの記入内容の送信方式は、HTTP と SMTP がある。SMTP の場合、そのままでは HTTP Proxy を通らずに直接送信されてしまうため、記入ルールの抽出ができない。そこで、自動記入で HTML を書き換える際に、<FORM>要素の ACTION と METHOD で定義される記入内容の送信先と送信方式を、本システムのフォーム記入エージェントを呼び出すように変更しておく。また、元の送信先と送信方式はフォーム記入エージェントへのパラメータとしてフォームの非表示項目の ___action と ___method に記しておく。また、どのフォームに記入した内容なのかという情報は、HTTP ヘッダの ReferrerURL に含まれるが、WWW ブラウザによってはこの項目が省略される場合がある。このため、フォームの URL を非表示項目の ___form に明示的に記しておく。

4.5 記入ルールの自動抽出方式

フォーム記入エージェントは、利用者が記入を終えて送信したときに、その記入内容から新しい記入ルールを抽出する。まず各々の記入項目について、送信時の記入内容が自動記入した内容と同じで変更されていない場合は、自動記入に成功したものと認識し、自動記入に利用した記入ルールをそのままルール管理エージェントにフィードバックする。それ以外の、自動記入しなかった項目と、利用者が自動記入内容を修正した項目について、以下の手順で新たに記入ルールの自動抽出を行う。

- (1) 記入内容から「なにを」記入したのかを推論する。具体的には、記入内容に一致する記入用個人情報 DB の値を検索し、そのキーとなる基本項目または複合項目名を得る。
たとえば、「太郎」という記入内容から、個人情報 DB の「名=“太郎”」を見つけ、「名」という概念名を得る。
- (2) (1) の結果から「どこに」にあたる記入項目の識別子と、「なにを」にあたる概念名からなる新しい記入ルールを生成し、ルール管理エージェントに渡す。ルール管理エージェントは、記入ルール群に追加登録する。
- (3) 自動記入時に埋め込んだ `__action` と `__method` のパラメータにより、記入内容を本来の送信先である WWW サーバや SMTP サーバに送信する。

5. 実験と結果

5.1 実験方法

本方式のルール抽出から自動記入までの一連の流れを確認するために実験を行った。対象フォームには、実際に利用されている個人情報の入力項目があるフォームを 50 件選んだ。50 件の内訳は Gator と ShareStage の方式 B に対応する各 10 件と就職活動用リンク集から新着順に選択したエントリーシート 30 件である。自動記入の対象とする記入項目は、住所、氏名、電話・FAX・携帯番号、生年月日、メールアドレスの各個人情報を記入するテキスト入力項目とした。50 件のフォームの全 738 項目の中で、これらの個人情報の入力項目が合計 497 項目あった。フォーム記入エージェントは、実際の個人情報を登録した 3 人分を用意した。ルール管理エージェントは、ルールが何も登録されていない初期状態とした。

これらの条件により、以下の手順で実験を行った。

- (1) まず、1 人目のフォーム記入エージェントを用いて各フォームを取り寄せ、何も自動記入されてい

表 2 自動記入精度

Table 2 Precision of filling in a form.

実験手順	自動記入した項目数		記入精度 (正しい結果/全項目数)
	正	誤	
1 回目	0	-	0% (0/738)
2 回目	531	497	95% (704/738)
3 回目	501	497	99% (734/738)

ないことを確認し、すべての項目に手動で記入を行い記入内容を送信した。

- (2) 次に、2 人目のフォーム記入エージェントを用いて各々のフォームを取り寄せ、自動記入された結果を確認した。記入内容の間違ひがある項目を修正し残りの項目にも手動で記入を行い記入内容を送信した。
- (3) さらに、3 人目のフォーム記入エージェントに、最上位のルールの累積ポイント数がそれまでの記入件数の 50% 以下なら自動記入しないという閾値を設定して、(2) と同様の作業を 3 人目のフォーム記入エージェントを用いて行った。

5.2 実験結果

実験結果を表 2 に示す。記入精度は、全記入項目に対する正しく自動記入できた記入項目の割合とした。対象の個人情報以外の記入項目には記入しないのが正しい結果である。

(2) で 531 項目に自動記入したのは、(1) において合計 531 の記入ルールが抽出されたからである。結果として個人情報を記入すべき 497 項目にはすべて正しく記入できた。記入精度が高かったのは利用者の記入が正確であり適切な記入ルールが抽出できたためと考えられる。一方個人情報以外を記入する 241 項目のうち 34 項目に間違って自動記入してしまったため、全体の記入精度は約 95% であった。

つまり 34 項目について誤ったルールを抽出してしまったことになる。このほとんどは、桁数の少ない数字を記入する項目に間違って自動記入してしまうものであった。たとえば、留学経験回数のような数値や、入学年月、卒業年月、就職説明会の希望月日の指定などの、和暦年、西暦年の下 2 桁、月、日などの 1~2 桁の数値が、個人情報の誕生日などと偶然一致してしまい、誤ったルールが抽出された。またほかには、(1) でユーザ ID として姓のローマ字表記を用いていたため、(2) ではそのルールによりユーザ ID の記入項目に誤って姓が記入された。

(3) では、閾値を設定することで間違って自動記入する項目数が減り、結果として記入精度が 99% に向上することを確認した。誤記入が 30 件減った理由は、

(2)において誤って自動記入された内容を修正して送信したことで、その30項目について(1)で抽出したルールが誤っていると判断できたからである。その結果、誤ったルールのポイント数の割合が50%に減ったため、閾値以下となり自動記入されなかった。一方、4件が間違っ て記入されたのは、先述の桁数の少ない数字を記入する4項目について、偶然に1人目と2人目で同じ誤ったルールが生成されたためである。たとえばそのうちの1つは入学和暦年の記入項目に誕生日の日を記入してしまうものであった。また、(3)でも個人情報を入力すべき497項目すべてが正しく記入できたのは、(2)において、記入内容の修正が必要なかったため(1)と同じ適切なルールが抽出されたからである。

6. 考 察

6.1 個人情報 DB の値重複

本方式では、記入用個人情報 DB の値から個人情報の基本/複合項目を検索するが、個人情報 DB の値は必ずしもユニークではないため、複数のキー候補が見つかることがある。実験結果では、1, 2桁程度の数字が記入された場合、個人情報以外の数値と偶然一致していることがあったが、1人の個人情報 DB の中でも年月日のどれかや、電話番号の一部などの1, 2桁の数値が重複する可能性があり、区別しにくい。短い数値以外にも、たとえば、利用者が電話番号とFAX番号に同じ番号を登録していることがよくある。そこで、まず値が重複した場合、重複しない場合は1.0ポイントだった登録時のポイント数を、候補が n 個の場合はポイントを等分し、各々を $1/n$ ポイントとして登録する。

個人情報 DB の重複の有無や重複の組合せは利用者によって異なる。つまり、自動記入ではこのポイント数の累計の最も多いものを採用するため、十分な記入ルールが集まれば正しい結果が上位となり、正しく記入できると考えられる。

6.2 サンプル不足とルールのばらつきによる誤差

本方式では、過去の最も多く記入されたルールを採用するが、ある程度のサンプル数が集まらなると誤差が大きくなってしまふ。また前述のように、個人情報 DB の値が重複した際には、ルールの集計結果のばらつきが大きくなる傾向がある。これらの場合、自動記入を誤ってしまう可能性が大きくなり問題である。

そこで、実験ではある項目に対して累積ポイント数が50%以下なら記入しないとしていたが、それに加えて、その累積ポイント数が10件(10ポイント)以下の場合にもこの方式で自動記入しないこととした。こ

れにより、この方式での適用範囲は多少狭まるが記入精度の低下を防ぐことができる。ここで記入しない場合、方式Cを用いることになる。この閾値は利用者により変更できるようにした。

6.3 ユーザインタフェース設計

自動記入する内容は個人情報であり、利用者が気付かないうちに外部に流出することがあってはならない。エージェントが自動記入した箇所は、記入項目の背景色を変更することでそのことが利用者にはっきりと分かるようにした。ただし、HTMLは記入項目を利用者の見えない位置に配置することもできる。このため、フォームとは別画面に自動記入した内容の一覧を表示し、確認できるようにした。

また、自動記入の間違ひは、窓口利用者が見つけて修正することになる。本システムでは自動記入の内容にどの程度の確信を持っているかの目安として、採用した候補のサンプル数によって記入項目の背景色を段階的に変えることで窓口利用者に分かりやすくした。

6.4 フォームの更新の検出

4.2節で述べたように、記入ルールでは記入項目の識別子の一部としてフォームのURLを用いているが、同じURLでもフォームの内容が変更される場合がある。HTML文書の更新は、以下の4通りが考えられる。

- (1) フォーム以外の部分の変更
- (2) フォーム項目の追加
- (3) フォーム項目の削除
- (4) フォーム項目は不変で、記入内容のみ変更

(1)の場合は何もする必要がない。(2)の場合、新しいルールが自動的に抽出されるだけである。(3)の場合、これまでの古いルールが使われなくなるだけである。(4)の場合、同じ名前のフォーム項目が違う目的で使われるもので、たとえば、これまでユーザを識別するために名前を書いてもらっていたところが、メールアドレスを書くように変更されたものがあった。

変更前：お名前：<INPUT NAME="user">

変更後：メールアドレス：<INPUT NAME="user">

しかしながら、実際には(1),(2),(3)などの少しの修正や追加のみでルールの削除が不要な場合の方が多くと考えられるため、HTML文書が変更されたからすべてを破棄するのではなく、(4)の場合のみこれまでの該当項目のルールを破棄する仕組みが必要である。

ルール管理エージェントでは、ルールの管理を容易にするために、ルールの登録日時を記録し、ある程度の量のルールが蓄積されると古いルールから順次削除する。これにより、時間とともに変更前の古いルールは無効になるが、それまでは記入ミスを連発すること

になる。そこで、フォームの変更と同時に記入間違いが連発した場合に、それまでに蓄積されたルールを無効にすることとした。

7. 関連研究・技術との比較

7.1 従来の自動記入方式との比較

2章で述べたように、以前利用したフォームには前回と同じ内容を記入すればよい場面がある。このような場合、フォームの意味を理解しなくても、方式Aのように単純に前回の内容を自動記入すればよい。しかし、実験で用いた就職用エントリーシートのように、1利用者が1度しか利用しない用途のフォームも多く、これらには方式Aは適用できない。本方式では、他の利用者の記入内容から抽出した、どこに何を記入するというルールからなる意味情報を利用することで自動記入を行う。本方式が適用できるのは、本システムを利用する誰かが記入したことのあるフォームであり、範囲が広い。

また、従来の方式Bでは、記入ルールを手作業で構築する必要があり、構築作業の負担が大きかった。5章の実験において、本方式では最初の利用者の記入内容から、記入ルールを自動抽出したため、手作業によるルールの構築はまったく必要なく、すべての対象フォームに自動記入できることを確認した。

方式Cは適用範囲は広いが記入精度が約7割程度と低い問題があった。本方式では5章の実験結果の記入精度が9割以上と高かった。また、2回目よりも3回目の記入精度が高くなったことから、記入ルールが多く集まることで記入精度が高くなることを示した。

7.2 個人情報オントロジの粒度

本方式では、個人情報オントロジは56項目とした。個人情報のオントロジには、vCardやLDAP/X.500などがあるが、本方式の56項目に相当するこれらのオントロジの数は各々29, 22項目である。本方式のオントロジ数が多いのは、オントロジを細かく分割したからである。自動記入の分野では、たとえば住所が都道府県から市区町村、番地まで1つの記入項目に記入するもの、都道府県と、市区町村番地に分かれているものなど、様々な書式があるため、これらの多様な書式に対応するために、個人情報を最小単位で管理し、大きい粒度の個人情報を、最小単位の情報の組合せで表現する本方式が適していると考えられる。

7.3 認証サーバ方式との比較

窓口利用者による記入の手間を省くための、別のアプローチとして、Passport¹⁴⁾、Liberty¹⁵⁾、TeaFlavor¹⁶⁾などの個人認証技術が研究・実用化されている。

これらは、個人情報を集中管理する認証サーバを置き、窓口利用者が、窓口で個人情報をフォームに記入する代わりに、窓口に対して認証サーバから個人情報を取得することを許可することで、窓口は認証サーバから個人情報を取得するものである。

この方式では、窓口側システムが各々の認証サーバに対応する必要がある。窓口利用者にとっては、利用する窓口が自分が登録している認証サーバに対応している必要があり、利用範囲が限定される。窓口サービス提供者側にとっては、認証サーバのインタフェースに対応するために既存のアプリケーションを再構築する必要がある。また、多くの利用者に利用してもらうためには複数の認証サーバに対応する必要があるが、各々の認証サーバとのインタフェースが標準化されていないため、開発コストが大きい。さらに、認証サーバを利用しない利用者のために従来のフォームを利用した方式も並行して提供しなければならないため保守コストが大きくなる。

本研究の方式は、既存のWWWシステムのフォームを対象に自動記入できるため、窓口利用者は利用できる窓口が多いメリットがある。また、窓口サービス提供者は、既存のシステムに変更を加える必要がない。

7.4 他の記入支援方式との比較

GatorとaiBAR2000は2章で述べた自動記入機能のほかに、コピー&ペーストによる記入支援機能がある。これらは、利用者が、個人情報を記入したいフォーム項目をペースト先として指定し、個人情報一覧からその項目に記入するものをコピー元として指定する、という2つの操作を行うことでそのフォーム項目に個人情報が入力される。

また、Internet Explorerのオートコンプリート機能は、フォーム項目に数文字を途中まで入力すると、過去の入力履歴から先頭文字が一致するものが一覧表示され、残りを補間し入力できるものである。

住所に限った入力支援機能として、多くの日本語インプットメソッド(かな漢字変換ツール)には、郵便番号を入力するとそれに対応する住所が変換候補に出てくる機能がある。さらに、インプットメソッドのユーザ辞書に、住所などの入力量の多い個人情報を短い読みを設定して登録することで、入力作業を軽減する利用例が知られている。

これらの半自動記入方式は、すべて手動で記入するよりは手間を軽減することができるが、いずれも1フォーム項目ずつ入力していくため、自動記入機能と比較して手間がかかる点で劣っている。

8. おわりに

本稿では、フォームへの記入内容から自動的に記入ルールを抽出し交換する分散協調型自動記入エージェントの実現方式を確立した。記入ルールを自動抽出することで、手動で行っていた記入ルールの定義が不要になることを示した。これにより、従来の適用範囲が限られていた問題が解決され、自動記入可能なフォーム数が大幅に増加した。

今後は広範囲な実験により、各パラメータの設定が妥当であるかを検証するとともに、より最適なパラメータと方式を検討していく。

謝辞 本稿をまとめるにあたり、有益なアドバイスをいただいた査読者の方々に感謝いたします。

参 考 文 献

- 1) 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, Vol.41, No.4, pp.1202-1211 (2000).
- 2) Microsoft Internet Explorer 5 Macintosh Edition, Microsoft Corp. (2000).
<http://www.microsoft.com/mac/products/ie/>
- 3) Netscape 6 Release Notes, Netscape Communications (2001).
<http://home.jp.netscape.com/eng/mozilla/ns6/relnotes/6.0.html>
- 4) Gator, Gator.com (2000).
<http://www.gator.com/>
- 5) ShareStage, NTT Communications (2001).
<http://www.sharestage.com/>
- 6) aiBAR2000, (2000).
<http://www.mfi.or.jp/miisan/>
- 7) 南谷圭持, 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—自動記入エージェントの実現方式, 情報処理学会第 62 回大会講演論文集(1) 1W-5, pp.273-274 (2000).
- 8) 藤原克哉, 中所武司: 分散アプリケーションフレームワーク wwHww におけるフォームナビゲーション機能の XML による実現方式, 情報処理学会ソフトウェア工学研究会ウィンターワークショップ・イン・金沢, pp.77-78 (2001).
- 9) Berners-Lee, T., Hendler, J. and Lassila, O.: *The Semantic Web*, Scientific American (2001).
- 10) Fensel, D., Horrocks, I., van Harmelen, F., McGuinness, D.L. and Patel-Schneider, P.F.: OIL: An Ontology Infrastructure for the Semantic Web, *IEEE Intelligent Systems*, Vol.16, No.2, pp.38-45 (2001).

- 11) 浦本直彦, 武田浩一: インターネットでの情報の記述と交換方式の最近の動向, 人工知能学会誌, Vol.13, No.4, pp.519-527 (1998).
- 12) Lassila, O. and Swick, R.R.: Resource Description Framework (RDF) Model and Syntax, *W3C Recommendation*, World Wide Web Consortium (1999).
- 13) Chusho, T. and Fujiwara, K.: FAFL: A Form-based Agent Communication Language for Enduser-Initiative Agent-Based Application Development, *COMPSAC 2000*, pp.139-148, IEEE Computer Society (2000).
- 14) .NET Passport SDK Documentation version 2.1, Microsoft Corp. (2001).
- 15) The Liberty Alliance Project (2001).
<http://www.projectliberty.org/>
- 16) TeaTray 技術情報, NTT Comware (2001).
<http://www.nttcom.co.jp/teatray/>

(平成 13 年 10 月 1 日受付)

(平成 14 年 3 月 14 日採録)



藤原 克哉 (学生会員)

1974 年生。2002 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後期課程修了。同年 4 月より秋田大学工学資源学部情報工学科助手、現在に至る。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, インターネット環境におけるアプリケーション開発技法に興味を持つ。電子情報通信学会, 日本ソフトウェア科学会, IEEE Computer Society 各会員。



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, 日本信頼性学会, IEEE Computer Society, ACM 各会員。