

語彙の階層構造に基づく申請書の自動記入

Auto Complete Method for Web Application form Based on Term Hierarchy

三木崇史^{*1} 小川修史^{*1} 松田憲幸^{*1} 三浦浩一^{*1} 瀧寛和^{*1} 堀聡^{*2} 安部憲広^{*3}

Takashi MIKI Hisashi OGAWA Noriyuki MATSUDA Hirokazu MIURA Hirokazu TAKI Satoshi HORI Norihiro ABE

^{*1} 和歌山大学大学院システム工学研究科 ^{*2} ものつくり大学 ^{*3} 九州工業大学情報工学部^{*1} Graduate school of Systems Engineering, Wakayama University^{*2} Monotsukuri Institute of Technologists^{*3} Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology

Recently, Web online application, such as member registration, is getting increase. Most online applications require information like name, affiliation and address. Though inputting contents are similar, we have to write differently by each application's format. Web browser's auto complete function and Cookie are useful when past inputted data need no conversion. These browser's function are useless when past inputted data is not match for application format. This paper describes auto complete method that convert past inputted data into formatted data for an application. Our method consists of label and format conversions with application's term hierarchy.

1. はじめに

インターネット上におけるコンテンツやサービスは多様化し、各種サービスの会員登録などのオンライン申請が行える Web ページを利用する機会も増加している。しかし、各申請書によって書く内容は似ているがその書き方はそれぞれ異なる。例えば、誕生日の表記について、西暦と和暦といった内容の違い、半角 4 桁や全角 2 桁といった書式の違いがある。これは各申請書によって求める情報やその優先度、利用意図が異なるためである。このように、申請書の記入者は同じ内容にも関わらずそれぞれの書式に応じて書き分けなければならない、面倒な作業となっている。

現在 Web ブラウザにはこれらの負担を軽減する Cookie やオートコンプリート機能がある[Microsoft]。これらは記入内容を保存することでユーザーの入力支援などが行える。しかし、内容や書式のの違いに対応することはできない。

申請書の自動補完の実現には、暦や桁数のように、各申請書が要求する内容や書式に変換しなければならない。そこで本稿では、申請書に記述されたラベルや書式間の関係を表す語彙階層を用いた自動補完手法を検討する。

2. 申請書語彙階層の構築と申請書へのタグ付け

2.1 申請書語彙階層

オンライン申請書に関する語彙階層を、大学生 5 名で合意形成を図り、オントロジーエディタ「法造」を用いて構築した [法造]。

概念“オンライン申請書”は、何について記入させたいのかを表す 1 つの“ラベル”とそのラベルに対する 0 か 1 つの“入力欄”を組とする“項目”、0 個以上の“但し書き”、さらに 0 個以上の“オンライン申請書”を構成概念としていたと考えた(図 1)。

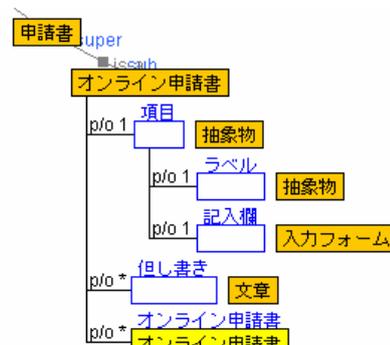


図 1 オンライン申請書の構成

簡単な例として図 2 を用いて説明する。まず全体を“オンライン申請書”と考え、「会員登録」が“ラベル”となり、「会員登録」に対する“入力欄”はなく、「*印は入力必須項目です。」が“但し書き”、残りの部分が“オンライン申請書”となる。その“オンライン申請書”では「会員情報の入力」が“ラベル”となり、「名前」「フリガナ」「メールアドレス」がそれぞれ“オンライン申請書”になると順に考えていく。

会員登録	
*印は入力必須項目です。	
■会員情報の入力	
名前 *	<input type="text"/> (全角)
フリガナ *	<input type="text"/> (全角)
メールアドレス	<input type="text"/> (半角英数字)

図 2 オンライン申請書の例

申請書においてラベルとなりうる概念は、概念“ラベル”の下位概念として“個人情報”“家族情報”“仕事情報”など、さらにその下位概念に“氏名”“住所”“職業”などのように定義した。

そして、概念“書式”の下位概念として“ひらがな”などの“表記”についての定義などを行っており、3.2 節で後述する記入履歴保存時のデフォルトの書式についても定義しておく。

また、“ひらがな”と“カタカナ”の概念間の表記変換関係など、関係概念を定義した。これらの関係概念の利用については 3.3 節で詳しく述べる。

連絡先: 三木崇史, 和歌山大学大学院システム工学研究科,
〒640-8510 和歌山市栄谷 930, TEL:073-457-8122,
Email:s071048@sys.wakayama-u.ac.jp

2.2 申請書へのタグ付け

以上のように構築した申請書語彙階層に基づき、申請書の各項目にタグ付けを行った。ここでのタグ付けとは、申請書語彙階層に基づいてラベルや書式を連結したものをその項目のフォームタグの name 属性に与えることをいい(図 3)、各申請書の策定時に行う。つまり申請書間で表記やデザイン上違う項目であっても、その項目に与えられているタグが同じであれば同じ意味の項目であることを意味するようになる。

また、現時点では手作業でタグ付けを行ったが、申請書語彙階層を読み込んで容易にラベルや書式のタグ付けを行えるエディタを想定しており、現在試作を検討中である。

電話番号(全角) - -

```
<input type="text" name="個人情報/電話番号/市外局番-全角"> -
<input type="text" name="個人情報/電話番号/市内局番-全角"> -
<input type="text" name="個人情報/電話番号/加入者番号-全角">
```

図 3 申請書語彙階層に基づく項目へのタグ付け例

3. 項目データの自動補完手法

3.1 自動補完システムの概要

システムの概要を述べる。申請書語彙階層の構築者がオントロジーエディタを用いて申請書語彙階層を構築し、各申請書策定者は申請書エディタを用いて申請書語彙階層に基づいたタグ付けを行った申請書を作成する。そしてユーザーが申請書にアクセスした際に、自動補完モジュールは過去の記入履歴および申請書語彙階層を参照し、項目の補完を行う。そして自動補完されなかった項目をユーザーが記入して申請すると、新たに記入した内容を記入履歴に蓄積する。

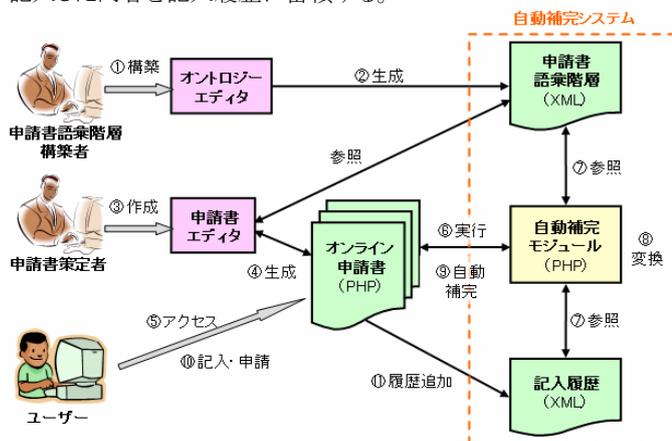


図 4 自動補完システム概要図

3.2 記入履歴の保存法

申請書を記入し申請が完了したとき、記入した内容は記入履歴として、各項目のタグのラベルに基づいて XML 形式で保存し蓄積する。

ただし、その際はユーザーが記入したデータをそのまま保存するのではなく、申請書語彙階層に定義されたデフォルトの書式に変換した値を保存しておき、履歴データを利用する際はそのつど必要な書式に変換して用いる。

例えば“全角数字”の“電話番号”を記入した場合であっても、保存時のデフォルトの書式が“半角数字”の場合は一旦、半角数字に変換を行った後に記入履歴として保存される(図 5)。

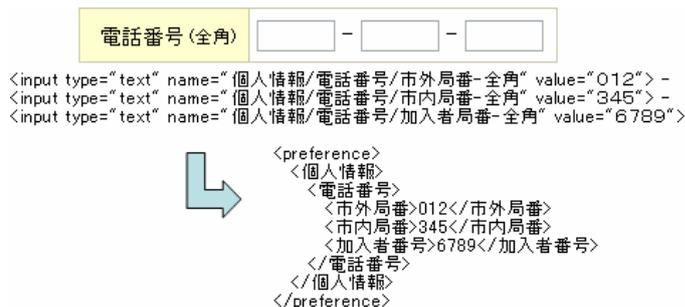


図 5 タグに基づく記入履歴への保存例

3.3 申請書語彙階層に基づく変換による自動補完

申請書の自動補完を行う際には、申請書内の全ての項目について以下の処理をする。ここでその項目 $item_e$ のタグと同一の記入履歴があった場合はその履歴データをそのまま補完すればよい(現在のブラウザのオートコンプリート機能に相当)。同一タグの記入履歴がなかった場合は、図 6 の手順で申請書語彙階層に基づいた自動変換が行われ、変換が成功した場合はその値を記入欄に補完する。

以下に、ラベル変換および書式変換について詳細を述べる。

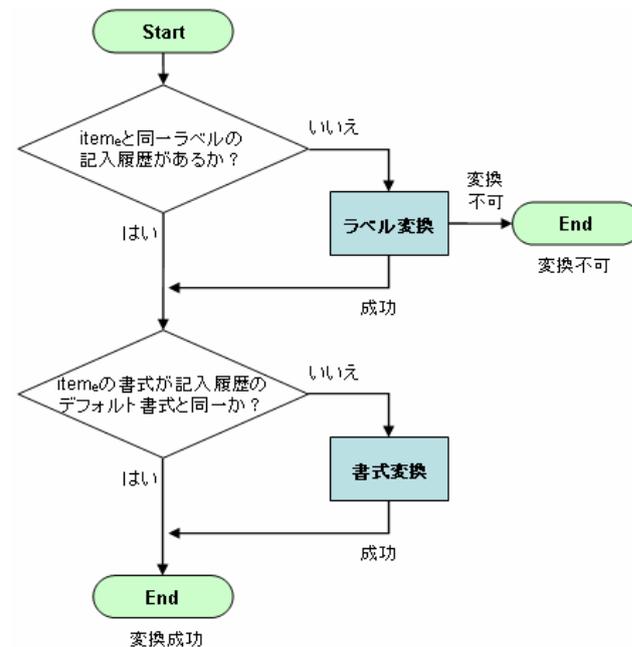


図 6 変換処理のフローチャート

(1) ラベル変換

ラベル変換は、項目 $item_e$ と同一ラベルの記入履歴がない場合に行われ、他のラベルの記入履歴から値を求める。この変換は、 $item_e$ のラベルが申請書語彙階層でどのように定義されているかに応じて行われ、任意の関係や part-of 関係、is-a 関係を用いた変換がある。また、項目によって単一の変換のみ行われるのではなく、その値が求められるまで可能な限りの変換が試みられる。

- 任意の関係を用了ラベル変換

申請書語彙階層には概念間に任意の関係を定義している。これらの関係に変換式を記述しておき利用することで、多様な自動変換へ対応が可能となる。

例えば、“西暦”と“和暦”の対応関係を用いた「2006 年」と「平成 18 年」との相互変換や、“生年月日”と“申請日”から“年齢”の算出などの変換を行う。

• part-of 関係を用いたラベル変換

項目 $item_e$ にタグ付けされていたラベルの概念に part-of 関係による部分概念が定義されていた場合は、それら部分概念の記入履歴を参照して連結したものを値とする。

例えば“電話番号”の場合、part-of 関係で“市外局番”“市内局番”“加入者番号”の 3 つの部分概念から構成されている。よって“電話番号”の記入履歴がない場合は“市外局番”“市内局番”“加入者番号”の記入履歴を順に連結したものが値となる。

これとは逆に、“電話番号”から“市外局番”のように全体概念の記入履歴から部分概念に自動変換するにはその区切り位置を判別することが困難なため、本手法では扱わない。

• is-a 関係を用いたラベル変換

項目 $item_e$ にタグ付けされていたラベルの概念に is-a 関係による上位概念が定義されていた場合は、その上位概念の記入履歴を値とする。

例えば“職業”について“正社員”と以前に選択した履歴があり、以降の申請書で職業を“会社員”“公務員”“自営業”などの中から選択させられる場合は、“正社員”の上位概念が“会社員”と定義されているので“会社員”が値となる。

またこれとは逆に、“会社員”から“正社員”のように上位概念の記入履歴から下位概念を自動選択することは出来ない。なぜなら上位概念は 1 つに定まるのに対し、下位概念は複数存在することがあり特定することが困難なためである(図 7)。

(2) 書式変換

記入履歴やラベル変換により求めた値は、記入履歴のデフォルト書式となっているため、 $item_e$ の書式と異なる場合には書式を変換する必要がある。

例えば、表記の変換が行われる。申請書の氏名や住所などの項目において、通常の漢字表記以外にひらがななどの表記での記入が求められる場合がある。そこで“ひらがな”“カタカナ”“ローマ字”間の相互変換の関係とその変換に用いる関数を申請書語彙階層に記述しておき、それを参照することにより表記の自動変換を可能とした(図 8)。

例えば、ひらがなの氏名に「みきたかし」との記入履歴があった場合、カタカナの「ミキタカシ」およびローマ字の「MIKI TAKASHI」へ自動変換が行える。

またその他にも、“西暦”の桁数を 4 桁から 2 桁にするといった自動変換などにも利用できる。

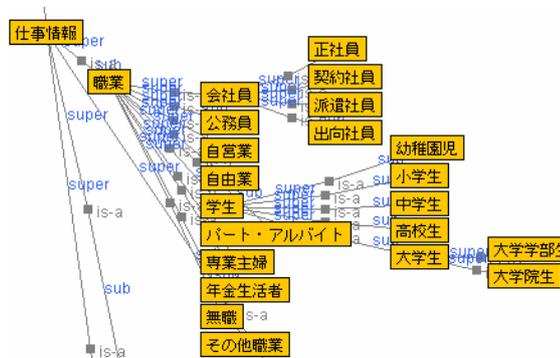
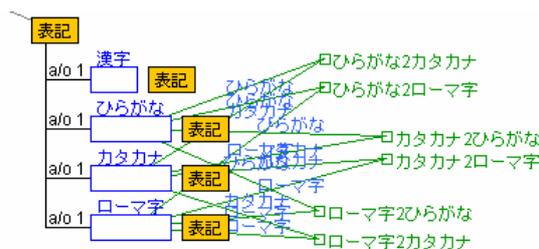


図 7 is-a 関係例



```

##### 表記変換用関数 #####
function h2k($hiragana){//ひらがな→カタカナ
    $katakana = mb_convert_kana($hiragana, 'C');
    return($katakana);
}
    
```

図 8 表記関係の定義例

4. システム実装例

4.1 実装の概要

本稿では図 3 のシステム概要図に基づいて実装を行った。ただしここでの申請書語彙階層構築者および申請書策定者は筆者らである。各申請書は実在する会員登録 Web ページ 4 サイトを基に、総入力項目 127 個にタグ付けを行い作成した。

各申請書は PHP、申請書語彙階層および記入履歴は XML で記述している。

また、本研究で対象としているのはオンライン申請書であるため、各申請書と申請書語彙階層は Web サーバー上に置いている。そして、記入履歴は各ユーザーの PC 内に保存する。

4.2 申請書読み込み時

ユーザーが申請書を読み込んだ際には、各項目において自動補完を行うために記入履歴および申請書語彙階層を参照する。ここでこれらのファイルは XML 形式で保存されているため、XML データ処理用 API である SimpleXML を用いて探索している。

4.3 申請完了時

ユーザーが各項目の記入を済ませ申請を行った際には、新たな記入内容を記入履歴に追加する必要があるため、同じく XML データ処理用 API である DOM を用いて、3.2 節で述べたように保存する。

5. 評価

5.1 評価方法

本学の学生 6 名に本手法で作成したオンライン申請書 4 枚を順に記入してもらった。申請書の記入順は、申請書が 4 枚であるためその全順序である 24 通り行った。

各申請書を作成する際において、項目の自動補完率と申請書作成にかかった時間を計測し、また、自動補完された内容が正しかったかどうかを評価してもらった。

以下に項目自動補完率と申請書 1 枚あたりの作成平均時間の計算方法を示す。

- 項目自動補完率の計算方法

$$\text{項目自動補完率(\%)} = \frac{\text{自動補完項目数}}{\text{全項目数}} \times 100$$

- 申請書 1 枚あたりの作成平均時間の計算方法

$$\text{作成平均時間(秒)} = \frac{\text{申請書作成合計時間(秒)}}{\text{サンプル数}}$$

5.2 評価結果

(1) 項目自動補完率

4 枚の申請書を順に記入してもらい、その記入順ごとに項目自動補完率を求めた。その結果を以下に示す。

表 1 項目自動補完率の評価結果

記入順	サンプル数	全項目数	自動補完数	自動補完率(%)
1 枚目	24	762	0	-
2 枚目	24	762	388	50.91
3 枚目	24	762	475	62.33
4 枚目	24	762	502	65.87

以上の結果から、1 枚目の申請書記入時には過去の記入履歴が無い項目の自動補完率は 0 となるが、2 枚目では約 5 割の項目を補完できており、それ以降も申請書の記入枚数を増すごとに自動補完率が向上していることが分かる。これは申請書作成ごとにその記入履歴が蓄積され、より多くの項目の補完に対応できるようになっていくためである。

(2) 自動補完内容の正確さ

各申請書を作成する際に自動補完された内容が正しかったかどうかを被験者に評価してもらった結果、自動補完された全 1365 項目においてその内容が正しかったとの回答を得ることができた。

(3) 申請書作成平均時間

4 枚の申請書を順に記入してもらい、申請書作成にかかった時間を計測し、その記入順ごとに申請書 1 枚当たりの作成平均時間を求めた。その結果を以下に示す。

表 2 申請書作成平均時間の評価結果

記入順	サンプル数	全項目数	作成合計時間(秒)	作成平均時間(秒)
1 枚目	24	762	2922	121.75
2 枚目	24	762	1418	59.08
3 枚目	24	762	848	35.33
4 枚目	24	762	728	30.33

以上の結果から、1 枚目の申請書の作成時間に比べ 2 枚目では約半分、4 枚目では約 4 分の 1 というように大幅に作成時間が短縮できていることが分かった。

当然利用する申請書によってこれらの結果に誤差は生じるが、申請書作成を重ねるごとに同様に自動補完率は高まり、作成平均時間も短縮できると考えられる。

6. おわりに

本稿では、オンライン申請書における項目記入時のユーザーの負担を軽減を目的に、申請書語彙階層に基づく項目自動補完手法を提案した。本手法はオンライン申請書に関する申請書語彙階層に基づいて申請書の各項目にタグ付けを行い、過去の記入履歴および申請書語彙階層を参照することで申請書の要求する書式への自動変換を可能とし、項目の自動補完を行った。

そして本手法を実装し評価実験を行った結果、本手法によりユーザーの負担が軽減できることが分かった。

しかし、より多くの申請書への対応や自動補完率の向上には、申請書の専門家による標準的な語彙階層(オントロジー[溝口 99])による申請書の策定が必要になると考えられる。

今後の課題としては、申請書を作成するエディタの作成である。本稿で実装した申請書は筆者が実在の申請書を基に作成したが、申請書の各項目へのタグ付けが行えるエディタの必要性があると考えられる。

また、本稿の実装例では個人情報のセキュリティは考慮していなかったため、セキュリティ問題への対応も課題となる。

現在、筆者の研究室内での運営を検討中である。

参考文献

- [Microsoft] Microsoft Internet Explorer : オートコンプリートによる時間の節約,
http://www.microsoft.com/windows/ie_intl/ja/using/howto/customizing/autocomplete.mspx
 [法造] オントロジーエディタ 法造, <http://www.hozo.jp/>
 [溝口 99] 溝口理一郎: オントロジー研究の基礎と応用, 人工知能学会誌, Vol.14, No.6, pp.977-988, 1999.