九州大学学術情報リポジトリ
Kyushu University Institutional Repository

複雑な検索サイトにおける入力フォーム情報の自動抽出

大森, 敬介
九州大学大学院システム情報科学府

中藤, 哲也
九州大学情報基盤センター

廣川, 佐千男
九州大学情報基盤センター

https://hdl.handle.net/2324/2950

出版情報：2005-03
バージョン：
権利関係：

DEWS2005 4B-i12

複雑な検索サイトにおける入力フォーム情報の自動抽出

大森　敬介† 　中藤　哲也†† 　廣川佐千男††

† 九州大学大学院システム情報科学府　〒 812–8581 福岡市東区箱崎 6–10–1
†† 九州大学情報基盤センター　〒 812–8581 福岡市東区箱崎 6–10–1
E-mail: †keisuke@matu.cc.kyushu-u.ac.jp, ††{nakatoh,hirokawa}@cc.kyushu-u.ac.jp

あらまし　近年，単純なキーワード検索でなく，複数の入力フィールドを用いる事で，より複雑な検索が可能な専門検索サイトが増えている．我々は，検索サイトの複雑化の動向についての調査を行い，このような複雑な検索サービスを統合する重要性を報告してきた．本論文では，検索において入力画面を構成する入力フィールドの一般的な構成を示し，その利用のために抽出すべき項目を明らかにする．また，これらのフォーム情報抽出ツールのためのアルゴリズムと，それを実装したツールの応用例を示す．

キーワード　Web 利用技術, 情報検索, 情報統合, 検索サイト，検索エンジン，Complex Query

Automatic extraction of input form information in complex search site

Keisuke OHMORI†, Tetsuya NAKATOH††, and Sachio HIROKAWA††

† Department of Informatics, Kyushu University
6–10–1 Hakozaki, Higasi-ku, Fukuoka 812–8581, Japan
†† Computing and Communications Center, Kyushu University
6–10–1 Hakozaki, Higasi-ku, Fukuoka 812–8581, Japan
E-mail: †keisuke@matu.cc.kyushu-u.ac.jp, ††{nakatoh,hirokawa}@cc.kyushu-u.ac.jp

Abstract　Increasing number of complex search sites is available on the Web. The query for such sites is not just a keyword but a complex query. We have been investigating the trend of the complex search sites, and reported the importance of integrating such complex search service. In this paper, we explain the general composition of the input fields in the input screen of the search sites, and clarify items to be extracted to use the sites. Moreover, we explain the outline of our extraction tool of the form information and the actual data extracted by the tool.

Key words　Metadata, Meta Search, Web Service, Search Engine, Deep Web

1. はじめに

WWW 上で提供されている情報検索のサイト（以下，**検索サイト**と呼ぶ）には，Google などの一般検索エンジンを持つサイトの他に，自サイト内のデータベースに対する検索機能を提供するサイトも数多く存在する．それらは，Web のインターフェースを持つデータベースという意味で Web データベースと呼ばれ，一般的な検索エンジンと区別される．それらのデータベース中の情報は直接参照する事ができず，検索によって動的に生成される Web ページによってのみ参照可能である．そのため，それらのページは Invisible Web [10], [11]，Deep Web [1], Hidden Web [3], [4] などと呼ばれている．

それらの検索サイトは特定のテーマに限定した質の高い情報やサービスを提供している事が多く，またその情報量は直接参照可能な Web ページの情報量よりも非常に多い（一説には 500 倍とも言われている）．それらのデータの自動的な取り扱いは，情報抽出の重要なテーマの一つである．

我々は，そのような検索サイトを自動的に解析する事で，情報の入出力を自動化し，いわゆるメタサーチシステムを動的に構築するシステム **DAISEn** [18] を提案している．DAISEn では，特定の分野に関する検索サイトを選び，キーワード検索を自動的に統合し，結果をユーザにまとめて提示する事が可能である．

近年検索サイトには，これまでとは異なる新しい方向性がみられるようになって来た．複数の項目を用いた複雑な質問が行われ，URL の単純なリストの代わりに，幾つかの項目から構成された情報の集まりのリストを返す検索サイトが増えている．例えば，Amazon.com [16] は本のリストを返す．kakaku.com [21] は PC のリストと共にそれらの価格を返す．Travelocity [23] は指定されたエリアのホテルのリストを返す．これらの専門的な検索サイトの入力形式は，一般的なサーチ・エンジンのものよ

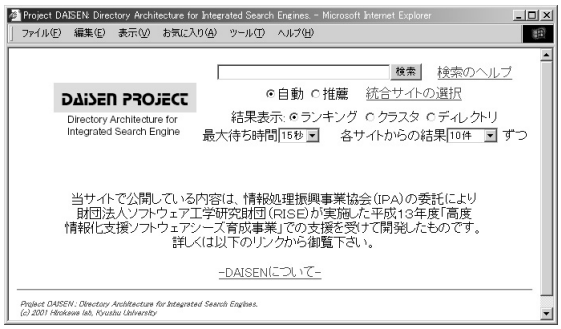

図 1　自動構築型メタサーチシステム DAISEn

り複雑である．入力としていくつかのキーワードを組み合せて指定することを必要とし，それぞれのキーワードが異なった属性を表す．乗り換え検索の Jorudan [20] では，出発と到着駅，日時が必要である．また，ホテルのための検索サイト Mytrip [22] では，チェックインの日付，チェックアウトの日付，人数，部屋数，価格の上限と地域が必要である．

これらの専門的な検索サイトを統合することは，利用者にとってより使いやすいシステムを構築することに他ならない．例えば，複数の PC パーツの検索サイトを統合することで最も安い PC パーツを扱う店を探すことができる．また，論文検索と翻訳サイトを統合することで，翻訳された論文を見ることもできる．

ネットワーク上における情報サービスの新しい形として Web サービスが注目されており，Web サービスの連携として Web Composition に関する研究が行われている [13]．しかしながら，今のところイントラネット内での運用が主であり，現在公開され利用可能な Web サービスは限定的である．今後，公開される Web サービスの増加には期待が持てるが，それを上回る多数のサイトで人間に対するユーザインターフェースを用いたサービスが提供され続けると思われる．

本研究は，それら複雑な検索サイトが持つ機能やサービスを動的に結合，連携し，新たなサービスを構築するという長期プロジェクトの一環である．その実現のためには，各検索サイトで扱われるデータベースのレコードが何であるかが予め分からなければならない．従来のデータベース，あるいは Web サービスであれば，データスキーマの明示的に与えられている．しかし，検索サイトにおいてはブラウザ経由での利用しか想定されていない．従って，各検索サイトが扱うデータスキーマは，入出力のページのフォーム情報や検索結果の出力情報から抽出，推定しなければならない．

複数の入力項目を持った検索サイトに対する統合的な検索サービスの自動的な取り扱いを実現するために，我々は，そのような検索サイトの収集，分析を続けている．本論文では，まずそれらの検索サイトの現状を報告し，入力フィールドを統合するための方法について説明する．更に，入力フィールドのデータスキーマを自動的に抽出するアルゴリズムを提案し，それを実装したツールについて述べる．最後に，そのツールによる情報を用いて作成した検索サービスの連携のプロトタイプに述べる．

2.　複雑化する検索サイト

複雑化の原因として，複数の入力項目を用いた複雑な質問が行われ，URL の単純なリストの代わりに幾つかの項目の集まりを返す検索サイト（図 2，図 3）が増えていることが挙げられる．

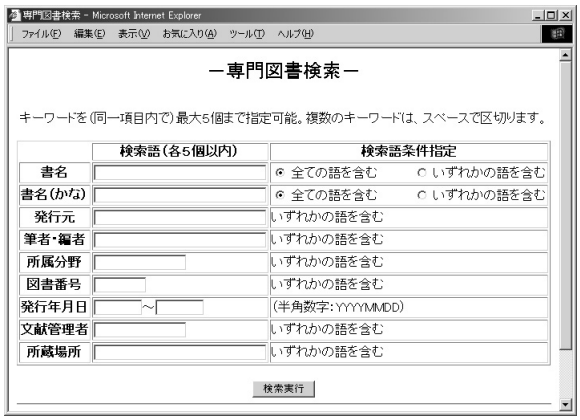

図 2　図書検索サイトの例

図 3　検索結果の例

我々は，複雑化する検索サイトの現状を調査するため，国立国会図書館関西館のデータベースナビゲーションサービス **Dnavi** [19] に登録されている検索サイトを対象データとしてその全体像を調査した．また，複雑な入力項目を持つ典型的な検索サイトに限定して詳細な調査を行った．Dnavi（図 4）では，WWW 上に存在するデータベースへのリンクとデータベース検索サービスを提供している．ただし，Dnavi における検索機能は本研究の目的であるメタサーチを提供するものではなく，目的とするデータベースの所在の検索サービスである．Dnavi の収録データベース数は 2005 年 2 月 1 日時点で約 9,000 件あり，このデータベース中には検索機能を提供している検索サイトが数多くある．

我々は，Dnavi に掲載されているサイトから専門検索サイト 2,880 件を抽出し，その全体像を調査した [7], [14].

表 1 は，複数のテキスト入力フィールドを持つサイトが 2,880 件中どのくらいあるかを表し，表 2 は，プルダウンメニュー，ラジオボタン，チェックボックスを用いて複雑な検索を行うサ

図 4　データベース・ナビゲーション・サービス “Dnavi”

表 1　テキストボックスの数

number of textbox	number of sites	number of textbox	number of sites
0	780	14	92
1	559	15	14
2	168	16	6
3	236	17	2
4	252	18	18
5	163	19	1
6	127	20	1
7	92	21	5
8	82	23	2
9	50	24	1
10	42	26	1
11	56	74	1
12	78	100	1
13	50		

表 2　# of Components

# of components	# of sites		
	pulldown menu	checkbox	radio button
0	1,227	2,263	2,036
1	310	49	5
2	221	52	169
3	271	111	156
4	144	81	117
5	163	26	78
6	88	34	70
7	61	77	80
8	63	24	44
9	32	13	25
10	21	20	15
＞10	279	130	88

イトがどのくらいあるかを表す．表 1 から，2,880 件の内，テキスト入力フィールドを 2 個以上持つサイトは 1,541 件あることが分かる．また，表 2 から，プルダウンメニューを持つサイトは 1,653 件，ラジオボタンをもつサイトは 617 件，チェックボックスをもつサイトが 844 件あることが分かる．これらのことから，複数の入力項目を持つ複雑な検索サイトが多くあることが分かる．

さらに，複数の入力項目を持ち特定の種類の検索サービスを提供する典型的な検索サイトとして図書検索サイト 938 件に対象を限定し，詳細な調査を行った [9]．この調査は入力項目に関するメタデータを自動で生成するための基礎情報を得る調査である．

図書検索サイトの典型的な一例を図 2 に示す．図書検索サイトの多くは，この例のように図書に関する複数の項目の 1 つあるいは複数を指定することで，データベース中の情報から指定にマッチする図書の一覧をユーザに提示する．

図書検索サイトにおけるテキスト入力フィールド(注1)数を図 5 に示す．この図から，テキスト入力フィールドが 2 個以上の複雑な図書検索サイトが約 8 割存在した．また，一見入力フィールドの少ないサイトも多いが，プルダウンメニュー(注2)等を用いてフィールドの属性を切り替えるサイトが多く見受けられる．

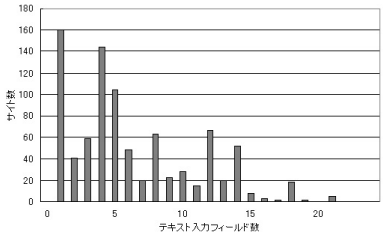

図 5　図書検索サイトの持つテキスト入力フィールド数

3.　検索サイトの入力項目

検索サイトの結合，より一般的には，検索機能を結合し新たな検索機能を構成するには，各検索サイトの入力情報と出力情報が何であるかを自動的に抽出しなければならない．検索結果の HTML ファイルに現われる反復パターンを発見し個別データ抽出するためのプログラムを自動あるいは半自動で生成する研究が多くなされてきた．このようなプログラムは狭い意味でラッパーと呼ばれる．一方，本論文のテーマは入力インターフェースとそのデータスキーマの自動抽出であり，まだ多くの研究はない．

複数の入力項目をもつ検索サイトを統合するシステムを自動で構築するには以下の 5 つの機能を実装する必要がある．

(A)　入力項目を持つ検索サイト URL の取得
(B)　検索サイトからのフォーム情報取得
(C)　入力項目の分類
(D)　入力項目の統合
(E)　検索結果から個別データの抽出

我々は，これら 5 項目のことについて研究を行ってきた．(A) については，すでに多くの検索サイトの URL を手動で収集している．(B) のフォーム情報については，これまでも多くの調査を行ってきたが，本論文ではそれらの調査結果を元に検索サイトからフォーム情報を自動で取得するアルゴリズムを提案し，それを実装したツールについて述べる．(C) については，図書検索サイトを分析し入力フィールド名に関するメタデータを作

(注1)：タグ <input type=text> で生成される要素
(注2)：タグ<select>で生成される要素

成した [8]．我々は，このメタデータやシソーラス等を用いることで入力項目の分類を行うことができると考えている．

(D)，(E) の技術は今後研究開発していく予定である．これらの技術を用いることで，統合検索システムを構成することができる．さらに，各検索サイトを Web サービスとして統一的に扱い，自由に組み合わせることでより複雑で有用な情報統合が可能と考えられる．

4. フォーム情報の自動抽出

検索サイトはブラウザ経由の利用しか想定されていない．従って，統合システムの構築時に利用できる情報は入力ページや検索結果の HTML ファイルのみである．本研究で統合システムを構築するために用いるデータは入力ページの HTML ファイルにおいて FORM タグ (<FORM>, </FORM>) で囲まれる部分から抽出される情報である．この情報のことを特にフォーム情報と呼ぶことにする．本論文では，複数の入力項目を持つ検索サイトの HTML ファイルから統合に必要なフォーム情報を取得する手法を考案した．本章では，統合対象となる入力ページの構造と入力項目の属性名について説明し，フォーム情報を取得するアルゴリズムを示す．

4.1 入力項目の属性

図 6 のような複数の入力項目を持つ検索サイトの入力ページから，検索の統合に必要なフォーム情報を取得する事を考える．一般に，各入力項目の直前には入力項目の属性名を示す文字列がある．たとえば，図 6 では「タイトル」,「著者名」,「出版者」,「出版年」,「件名」,「キーワード」,「分類」の文字列である．本論文では，これらを各入力項目のラベルと呼ぶ．ラベルはその検索サイトの機能的意味を示すもので，フォーム情報の中で最も重要なものである．

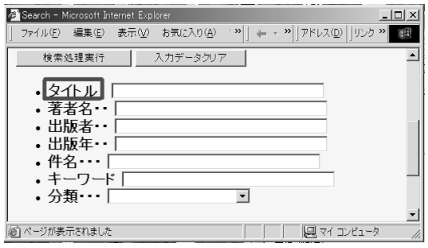

図 6　入力項目とそのラベル

従来の研究 [15] においては，このようなラベルとして各入力項目の直前の文字列を想定していた．しかし我々の調査の結果，複数の入力項目を持つサイトでは，多くの場合 TABLE タグが用いられていることが明らかになった．このため本研究では，TABLE タグで表される入力項目群からラベルを抽出するためのヒューリスティックスを考案した．我々のヒューリスティクスでは，左端，上端，直前にラベルが現われるとする．下記に，左端，上端，直前にラベルが現われている検索サイトの例を示す．

図 7 では，ラベルは入力項目の左端に現われている．この図では，プルダウンメニュー中の,「全て」,「標題」,「著者名」,「出版者」,「件名」,「フルタイトル」がラベル候補となる．

図 8 では，ラベルは入力項目の上端に現われている．この図

図 7　入力項目の左端に現われるラベル

では，プルダウンメニュー中の,「フリーワード」,「タイトル」,「フルタイトル」,「著者」,「出版者」,「件名」,「分類」,「ISBN」がラベル候補となる．

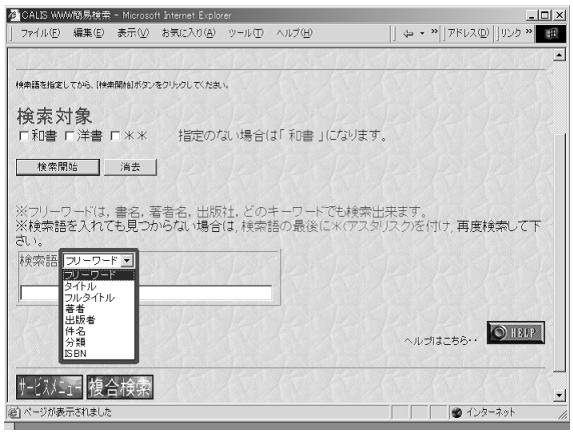

図 8　入力項目の上端に現われるラベル

図 9 では，ラベルは入力項目の直前に現われていることがわかる．この図では，プルダウンメニュー中の,「書名／タイトル」,「著者名／制作」,「出版社／発売者」がラベル候補となる．

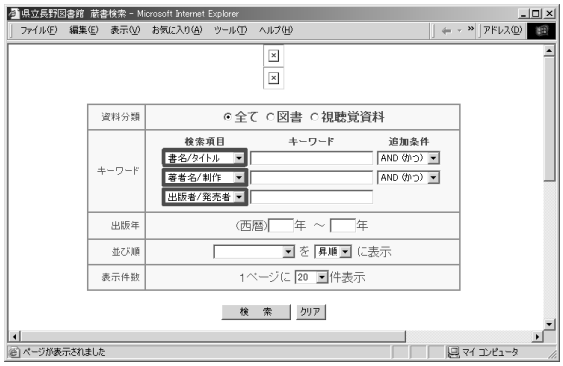

図 9　入力項目の直前に現われるラベル

4.2 フォーム情報の定式化

入力項目のラベル情報を含んだ，フォーム情報の定式化を行う．図 10 の例のような具体的におけるフォーム情報は，FORM タグの action 値 “./input.cgi” と method 値 “GET”，INPUT

タグの type 値 “text” と name 値 “te”，OPTION タグの value 値 “opt1” と “opt2” と “opt3” や OPTION タグ直後の文字列「属性名 1」と「属性名 2」と「属性名 3」である．

本論文では，ラベルが文字列である場合，INPUT タグと同様の構造に変換する．その type 値は “word”，term 値は「ラベル候補の文字列」である．これにより，他の要素と同様に扱う事が可能となる．

```
<FORM action="./input.cgi" method="GET">
    <SELECT name="select">
        <OPTION value="opt1"> 属性名 1 </OPTION>
        <OPTION value="opt2"> 属性名 2 </OPTION>
        <OPTION value="opt3"> 属性名 3 </OPTION>
    </SELECT>
    <INPUT type="text" name="te">
</FORM>
```

図 10　HTML ファイルにおける FORM タグ

BNF 表記で表したフォーム情報の構造を図 11 に示す．以下では，フォーム情報の構成要素について説明する．

フォーム情報　複数の form から構成される．

form　FORM タグ 1 つ分の情報を持ち，action, method と複数の input から構成される．

method　FORM タグにおける method 値であり，一般に “GET” か “POST” である．

action　FORM タグにおける action 値である．

input　INPUT タグや SELECT タグ 1 つ分の情報を持ち，type, name 及び複数の value, term, pointer, initial で構成される．

type　INPUT タグにおける type 値のことで，“text”, “radio” や “checkbox” である．SELECT タグは “select” という type 値を持った INPUT タグに変換する．

name　INPUT タグや SELECT タグの name 値である．

value　INPUT タグにおける value 値であり，SELECT タグの value 値は OPTION タグの value 値を用いることとし，OPTION タグが複数の場合 value 値も複数と定義する．

term　INPUT の type 値が “radio” か “checkbox” の場合は INPUT タグ直後の文字列とする．type 値が “select” の場合は OPTION タグ直後の文字列とし，OPTION タグが複数の場合 term 値も複数と定義する．また，type 値が “word” の場合は「ラベル候補の文字列」である．

pointer　この input のラベル候補が何番目の input であるかを示す．

initial　input の type 値が “radio” か “checkbox” の場合は何番目に “checked” が付いていたかを示す数字である．type 値が “select” の場合は何番目の OPTION タグに “checked” が付いていたかを示す．

4.3　フォーム情報抽出アルゴリズム

図 12 は，複数の入力項目を持つ検索サイトの HTML ファイルからフォーム情報を取得する手順である．まず，4.3.1 節〜4.3.4 節に示した手順で HTML ファイルを前処理した後に，

```
フォーム情報 := (form*);
form := (method, action, input*);
method := GET | POST ;
input := (type, name, value*, term*, pointer*, initial*);
type := text | radio | checkbox | select | word | etc ;
pointer := 整数;
initial := 整数
```

図 11　フォーム情報の BNF 表記

4.3.5 節〜4.3.9 節に示した手順でフォーム情報の抽出を行う．

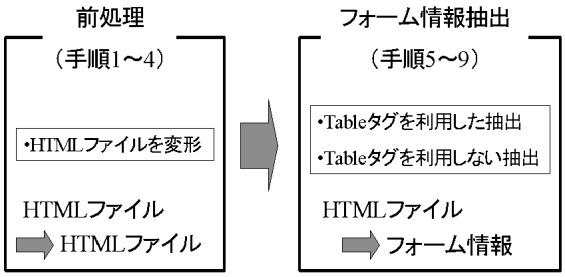

図 12　フォーム情報抽出の手順

このアルゴリズムによって，各テキスト入力フィールドに関して最大３つのラベルが取得される．それらのうち，どのラベルを採用するかは，検索結果のフィールド名の情報を用いて選択する方法 [8] などが考えられるが，今後の課題である．現在，このアルゴリズムによって得られる情報を評価するための正解例の作成，及びデータ収集を行っている．

4.3.1　無視するタグの除外

HTML ファイルから，<ADDRESS>, <SCRIPT>, <!>, <FONT>, <LABEL>, <B>, <I>, <U>, <S>, <TT>, <SUP>, <SUB>, <NOBR>, <CENTER>, <A> の各タグ（終了タグを含む）を除去する．

4.3.2　SELECT タグから INPUT タグへの変換

SELECT タグで囲まれた部分を INPUT タグへと変換する．変換方法は，INPUT タグの type 値を “select”，value 値を OPTION タグの value 値とし，OPTION タグ直後の文字列を INPUT タグの直後に配置する．OPTION タグが複数あり，OPTION タグの value 値と OPTION タグ直後の文字列が複数存在する場合は，その value 値と文字列をそれぞれコンマで区切り，INPUT タグ中の value と INPUT タグ直後の位置にそれぞれ配置する（図 13，14）．

4.3.3　連続する INPUT タグの統合

INPUT タグの type 値が “radio” か “checkbox” であり，name 値が同じものが連続して現われる場合，それら連続する INPUT タグを 1 つに統合する．統合方法は，それら連続する INPUT タグ中の value 値と INPUT タグ直後の文字列をそれぞれコンマで区切り，INPUT タグ中の value と INPUT タグ直後の位置にそれぞれ配置する（図 15，16）．

4.3.4 INPUT，SELECT タグへの番号付加

INPUT タグそれぞれに対し，何番目の FORM タグの INPUT タグかを示す “form-num”，及び，その FORM タグ中の何番目の INPUT タグかを表す “input-num” をそれぞれ INPUT タグ中に新たに付加する（図 17, 18）.

4.3.5 FORM タグ中の action 値と method 値の取得

各 FORM タグ中から，method 値として “GET” か “POST” を取得し，また，action 値も取得する．もし，action 値が “./input.cgi” のような相対パスの場合は，URL 等を補って絶対パスへと変換する（図 19）.

4.3.6 TABLE タグの内容の 2 次元配列化

TABLE タグで囲まれる部分を<TR>や<TH>と<TD>を考慮して 2 次元配列へと格納する．これは，<TR>が各行の区切りを表し，<TH>と<TD>が各列の区切りを表していることを利用する．もし，<TH>と<TD>に “colspan” や “rowspan” のように複数の行と列にまたがることを表す指示がある場合はこのことも考慮して 2 次元配列へと格納する（図 20，21）.

4.3.7 2 次元データの整形

TABLE タグで囲まれた部分を格納した 2 次元配列において，一列，または一行全てのデータが空の場合は 2 次元データの整形を行う（図 22，23）.

4.3.8 一般入力項目のラベル取得

TABLE タグで囲まれていない部分に入力項目がある場合，各入力項目の直前の文字列をラベルとして取得する.

文字列 1<input1> 文字列 2<input2><input3>

上図における入力項目のラベルとして,

・入力項目 <input1> は直前のラベルとして文字列 1 を取得する.

・入力項目 <input2> は直前のラベルとして文字列 2 を取得する.

・入力項目 <input3> は直前のラベルとして <input2> を取得する.

4.3.9 2 次元データからの入力項目ラベル取得

TABLE タグで囲まれた部分の 2 次元データを解析することで，データ中に含まれる入力項目のラベルを取得する.

入力項目からみて左端，上端，直前の三種類の文字列または入力項目をラベル候補として取得する．ラベル候補として入力項目を取得した場合は，その入力項目のラベルを再取得する.

文字列 11	···	文字列 12<input1> 文字列 13
···	···	···
文字列 41	···	文字列 42<input2> 文字列 43<input3>

上図における入力項目のラベル候補として,

入力項目 <input1> は左端のラベルとして文字列 11, 直前のラベルとして文字列 12 を取得する.

入力項目 <input2> は左端のラベルとして文字列 41, 上端のラベルとして <input1>，直前のラベルとして文字列 42 を取得する.

入力項目 <input3> は左端のラベルとして文字列 41, 上端のラベルとして <input1>，直前のラベルとして文字列 43 を取得する.

```
<SELECT name=na>
  <OPTION value=val1>セレクト 1
  <OPTION value=val2>セレクト 2
</SELECT>
```

図 13　SELECT タグから INPUT タグへの変換（変換前）

```
<INPUT type="select" name="na"
       value="val1,val2"> セレクト 1, セレクト 2
```

図 14　SELECT タグから INPUT タグへの変換（変換後）

```
<INPUT type=radio name=na value=val1>ラジオ 1
<INPUT type=radio name=na value=val2>ラジオ 2
```

図 15　連続する INPUT タグの統合（統合前）

```
<INPUT type=radio name=na value="val1,val2">
ラジオ 1, ラジオ 2
```

図 16　連続する INPUT タグの統合（統合後）

```
<INPUT type=text name=na>
```

図 17　INPUT，SELECT タグへの番号付加（付加前）

```
<INPUT type=text name=nam form-num=1 input-num=1>
```

図 18　INPUT，SELECT タグへの番号付加（付加後）

```
<FORM method="GET" action="./input.cgi">
```

図 19　FORM タグ中の action 値と method 値の取得

```
<TABLE>
 <TR><TD>A1</TD><TD>B1</TD><TD>C1</TD></TR>
 <TR><TD>A2</TD><TD>B2</TD><TD>C2</TD></TR>
 <TR><TD>A3</TD><TD>B3</TD><TD>C3</TD></TR>
</TABLE>
```

図 20　TABLE タグの内容の 2 次元配列化（2 次元配列化前）

A1	B1	C1
A2	B2	C2
A3	B3	C3

図 21　TABLE タグの内容の 2 次元配列化（2 次元配列化後）

空	空	空
空	A1	B1
空	A2	B2

図 22　2 次元データの整形（整形前）

A1	B1
A2	B2

図 23　2 次元データの整形（整形後）

5. フォーム情報抽出ツールの応用

我々は，フォーム情報取得ツールを実際に作成し，このツールを利用して情報収集を行う検索エージェントのプロトタイプを作成した．本プロトタイプは，次に示す各学会の論文検索サイトを対象に，論文の情報を収集する事を目的としている．

- 情報処理学会電子図書館(注3)
- 電子情報通信学会 和文論文誌(注4)，英文論文誌(注5)
- 人工知能学会論文誌(注6)
- 日本ソフトウェア科学会 J-STAGE(注7)

本システムは主に三つの機能から成り立っている．それらは，(1) 複数の検索サイトに対して同時に検索を行い，結果を統合してユーザに提示する機能，(2) 結果中の著者名を抽出し，リスティングする機能，(3) リスティングされた著者名をキーとした次のステップの検索を提供する機能，である．

(1) は，いわゆるメタサーチの機能である．個々の検索サイトに対するラッパーにより入出力の違いを隠蔽し，得られた複数の結果を組み合わせてユーザに提示する．(2) は，出力結果のページの解析により著者名及び共著者名を抽出し，それらを一覧表としてユーザに提示する．これは (3) の機能へのポインタともなっている．(3) は，得られた情報を元に繰り返し検索を行う機能である．得られた論文一覧中の著者名をクリックする事で再び新たな検索を行い，その著者に関する論文情報を提示する．

この 3 つの機能のうち，(2), (3) は文献検索システム DBLP で用いられているものと同等である．検索結果に対するこのような処理を含む機能は，利用している DB の直接的アクセスが必要なので，DBLP のように通常システム中に組み込まれなければならない．一方，我々の提案する方式では，独立した文献検索システムを統合するだけでなく，この (2), (3) の機能をそれぞれのシステムの外部に構成することができる．

これらの機能のデータ結合の模式図を図 24 に示す．

本システムの基本動作をみよう．最初に，著者名による検索か，キーワードによる全文検索を行う（図 25）．本システムは入力された条件（キーワード or 著者名）を各検索サイトの要求するフォームに変換する．そのフォームを各検索サイトへ送り，それぞれ検索を行う．得られた結果は，各検索サイト毎のラッパーでフィールド単位に分割し，全てのサイトからの結果を一つの表にまとめてからユーザに提示し，同時に次の検索へデータを渡すためのリンクを生成し，各著者名に関連付ける（図 26）．

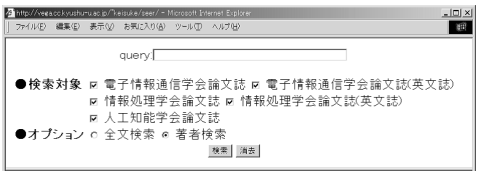

図 25　著者検索システム（プロトタイプ）

図 26　検索結果の例

ユーザは，参照したい著者名をクリックするだけで，順次関連情報を検索して行く事が可能である．我々は本システムを，http://matu.cc.kyushu-u.ac.jp/guruguru にて公開している．

6. 関 連 研 究

Web 上のサービスの連携に関するこれまでの研究 [2], [12] では，各 Web データベースの詳細情報を開発元からされること，あるいは共通形式のデータへの変換プログラムが提供されることを想定している．本論文で提案する手法は，各検索サイトの Web インターフェースだけから必要な情報を得るものであり，各サイトの開発，運用システムとは完全に独立に実現できる．

北村ら [5] は，WWW より情報を抽出し統合するスクリプト言語 MetaCommander を実装した．HTML ページから希望するデータを抽出する為の手順をスクリプトとして記述する事で，

(注3)：http://www.bookpark.ne.jp/ipsj/, 会誌，英文誌，研究報告，論文誌（ジャーナル），欧文誌，論文誌（トランザクション）を含む
(注4)：http://search.ieice.org/jpn/search-j.html
(注5)：http://search.ieice.org/search.html
(注6)：http://tjsai.jstage.jst.go.jp/ja/
(注7)：http://www.jstage.jst.go.jp/browse/jssst/-char/ja/

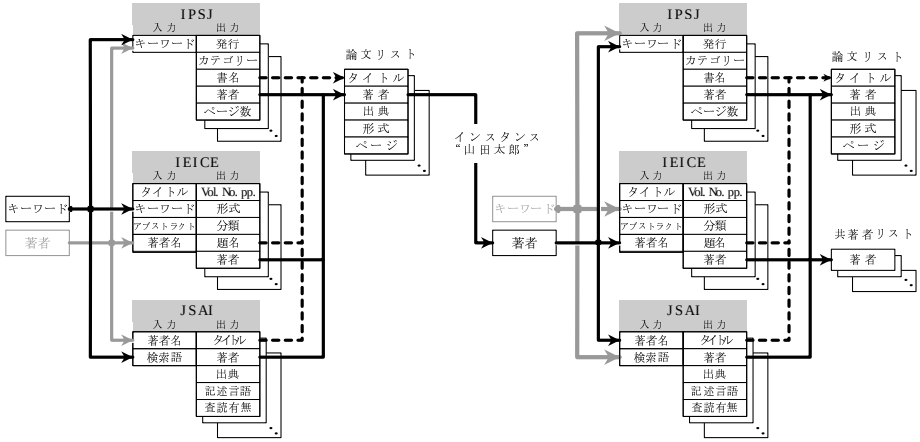

図 24　データ結合の模式図

目的のデータを入手するシステムである．しかし，タグや文字列として表された HTML 文書にどのようなデータ構造が含まれているかをスクリプトを書くユーザーが考え，そのデータ構造の表現形式をタグや文字列として記述する必要があり，データスキーマが自動的に抽出される訳ではない．

情報融合のエージェントについての関連研究としては，Knoblock らによる ARIADNE [6] がある．これは，学習に基づいた情報抽出エージェントを容易に構築するための枠組みと，それらを組み合わせるための枠組みを与えているが，対象は一般の Web(Visible Web) である．

Zhang ら [15] は，クエリーフォーム全般に存在する隠された共通の文法を想定し，それへのパージングを試みるもので，我々の研究に最も近い．しかしながら，想定された構造が我々のものとは異なっている．我々の想定する構造がより現実に即していると考えている．

7. ま と め

本稿では，専門検索サイトを調査することにより複雑な検索サイトを統合することの有用性を示し，複雑な検索サイトを統合する手順を紹介するとともに統合に必要な技術に触れた．特に，フォーム情報取得ツールについて詳細な解説を行い，最後に本ツールを使用して作成した情報収集を行う検索エージェントのプロトタイプを紹介した．現在，このフォーム情報取得ツールの評価のためのデータ収集を行うとともに，自動的に統合検索システムを構築することを試みている．本ツールの使用により，検索サイトを Web サービスとして統一的に扱う事が可能となり，それらを自在に組み合わせることによって，統合検索システムの自動構築のみならず本論文内で紹介した検索エージェントのプロトタイプのような，より複雑な情報統合を行うシステムの自動統合ができると考えている．

文　献

[1] BrightPlanet, The Deep Web: Surfacing Hidden Value, BrightPlanet White Paper, 2000.

[2] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. ”The TSIMMIS Project: Integration of Heterogeneous Information Sources”. In Proceedings of IPSJ Conference, pp. 7-18, Tokyo, Japan, October 1994.

[3] P. Ipeirotis, L. Gravano and M. Sahami, PERSIVAL Demo: Categorizing Hidden-Web Resources, JCDL2001, 2001.

[4] P. Ipeirotis, L. Gravano and M. Sahami, Probe, Count, and Classify: Categorizing Hidden-Web Databases, ACM SIGMOD 2001, 2001.

[5] Yasuhiko Kitamura, Tomoya Noda, and Shoji Tatsumi, Single-agent and Multi-agent Approaches to WWW Information Integration, Multiagent Platforms, Lecture Notes in Artificial Intelligence, Vol. 1599, Berlin et al.: Springer-Verlag, 133-147, 1999.

[6] Knoblock, C. A., S. Minton, J. L. Ambite, N. Ashish, I. Muslea, A. G. Philpot, and S. Tejada, The Ariadne Approach to Web-Based Information Integration, International Journal of Cooperative Information Systems, vol.10, no.1-2, pp.145-169, 2001.

[7] T. Nakatoh, K. Ohmori, Y. Yamada and S. Hirokawa, *COMPLEX QUERY AND METADATA*, Proc. ISEE2003, pp. 291-294, 2003.

[8] 大森 敬介, 中藤 哲也, 原 由加里, 廣川佐千男, 検索サイトにおける入力項目と検索結果のフィールド名の対応調査 FIT2004, pp. 89-90, 2004.

[9] 大森敬介, 中藤哲也, 山田泰寛, 原由加里, 廣川佐千男, 複雑な検索機能を持つ検索サイトの動向調査 DEWS2004, I-1-05, 2004.

[10] P. Pedley, The invisible web, ASLIB, 2001.

[11] C. Sherman and G. Pric, The Invisible Web, Infomation Today, Inc., Medfore, New Jersey, 2001.

[12] 菅坂 玉美, 益岡 竜介, 佐藤 陽, 北島 弘伸, 丸山 文宏. 知的エージェント環境 SAGE の EC への適用, 取引フェーズへの適用. 第 6 回マルチ・エージェントと協調計算ワークショップ (MACC), 日本ソフトウェア科学会, 1997 年 12 月.

[13] S. Thakkar, C. A. Knoblock, J. Ambite and C. Shahabi, Dynamically Composing Web Services from On-line Sources, Proc. of 2002 AAAI Workshop on Intelligent Service Integration, Edmonton, Alberta, Canada.

[14] 山田泰寛, 松永吉広, 野口正人, 中藤哲也, 廣川佐千男, 統合検索システム DAISEn での検索サイトフォーム分析, 情報処理学会研究報告 2003-DBS-131(II)(77)(夏のデータベースワークショップ DEWS2003), pp.311-318, 2003.

[15] Zhen Zhang, Bin He, Kevin ChenChuan Chang, Understanding Web Query Interfaces: BestEffort Parsing with Hidden Syntax, SIGMOD2004.

[16] Amazon.com, `http://www.amazon.com/`

[17] askOnce, `http://www.askonce.com/`

[18] 専門検索サイトの動的統合による次世代検索システム DAISEn, Directory Architecture for Integrated Search Engines, `http://daisen.cc.kyushu-u.ac.jp/`

[19] 国立国会図書館関西館データベース・ナビゲーション・サービス Dnavi, `http://dnavi.ndl.go.jp/`

[20] Jorudan, `http://www.jorudan.co.jp/`

[21] kakaku.com, `http://www.kakaku.com/`

[22] Mytrip, `http://www.mytrip.net/`

[23] Travelocity, `http://www.travelocity.com/`