フォーム定義にもとづくPHPデータ入力プログラムの自動生成

吉田　彩子†　　遠山　元道††

† †† 慶應義塾大学理工学部情報工学科　〒 223–8522 神奈川県横浜市港北区日吉 3–14–1
E-mail: †aya@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

あらまし　近年，インターネットの爆発的な普及により，データベースと Web の統合的な利用が増えてきた．しかし，そのためにはプログラミングの技術などの専門的な知識が必要であった．そこで本研究では，Web を用いたアンケートの結果の収集等の応用に，専門的な知識がなくても対応できるよう，独自のデータ格納言語を定義し，フォーム定義にもとづいて PHP データ入力プログラムを自動生成するシステムを開発した．
キーワード　データベース, Web, フォーム, PHP

Automatic Generation of Data Input Program in PHP Based on Form Definition

Ayako YOSHIDA† and Motomichi TOYAMA††

† ††Department of Information and Computer Science, Faculty of Science and Technology,
Keio University
Hiyoshi3–14–1, Kouhoku-ku, Yokohama-shi, Kanagawa, 223–8522 Japan
E-mail: †aya@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

Abstract　Recently, due to widespread use of Internet, the integrated use of Database and Web is increasing. But to do that, we need a technical knowledge such as a programming. So in this paper, we define the original data input language, and develop the system which generate PHP program automatically based on Form definition.
Key words　Database, Web, Form, PHP

1. はじめに

近年，インターネットの爆発的な普及により，データベースと Web の統合的な利用が増えてきた．Web を用いたアンケート調査や企業のエントリーシート，オンラインショッピングなどその用いられ方は多岐にわたっている．それに伴い，Web ベースのデータベースアプリケーションを構築したいというニーズも増えてきている．それによって，初心者から上級者まで幅広い層からの Web コンピューティングに対する関心も高まってきた．しかし，Web コンピューティングにはプログラミングの技術などの専門的な知識が必要であり，専門的な知識のない人々にとってそのようなアプリケーション構築の実現はやはり困難なものであった．

そこで本研究では，Web を用いたアンケートの結果の収集等の応用に，専門的な知識がなくても対応できるよう，独自のデータ格納言語を定義し，フォーム定義にもとづいて PHP データ入力プログラムを自動生成するシステムを提案する．

2. 章でまずデータベースと Web との連携プログラミング言語である PHP とフォームの定義について述べ，3. 章で本研究で提案するデータ格納言語の仕様を示した後システムの実現方法を述べる．4. 章では本システムの使用例を示し，5. 章では本研究の評価を行う．最後に 6. 章で結論を述べる．

2. データベースと Web の連携

近年，データベースに対してインターネットから情報の検索，登録を行うといったデータベースと Web の統合的な利用が増えてきた．データベースと Web を連携させるための技術として，PHP や Java，Perl などがある．一般的にユーザは，Web ブラウザから必要事項を入力し，入力を受け取ったこれらのプログラムが，データとデータベースとのやりとりを行う．

本研究では，PHP を用いてシステムを作成する．

2.1 PHP

PHP とは Hypertext Preprocessor の略で，HTML 埋め込み型のサーバサイドスクリプト言語である．HTML ファイル内に記述することで，サーバサイドで動作する Web アプリケーションを効率的に開発することができる．

2.2 FORM

フォームとは，ブラウザから Web サーバへデータを送るための手段を提供するための HTML タグである．

<form [method=post|get] action={"URL"|"mailto:メールアドレス"}>から</form>までがひとつのフォームとしてブラウザに認識される．ブラウザからの入力が行われると，このフォームで入力された値が変数=値のリストとしてサーバ側に送信される．

method は変数をサーバに渡すための方式を指定し，action は変数リストをどこに渡すのかを指定する．

フォームの要素には様々な種類がある．

（1） テキストボックス

<input type=text name=変数名 size=数値>

1 行で入力できる短い文字列を入力するためのもの．

size を指定して見た目の幅を変更することもできる．

（2） テキストエリア

<textarea name=変数名 rows=数値 cols=数値></textarea>

複数行の入力ができる領域であり，数行にわたる内容を入力するためのもの．

入力領域のサイズは rows(行数) と cols(列数) で指定する．

（3） ラジオボタン

<input type=radio name=変数名 value=文字列>

選択可能なラジオボタンを表示する．

同一変数名に対して複数の radio 属性を指定することにより複数のボタンを表示してどれかひとつを選択させる．

（4） チェックボックス

<input type=checkbox name=変数名 value=文字列>

選択可能なチェックボックスを表示する．

各エントリに異なった変数名をつけることで複数の選択を可能にする．

（5） リストボックス

<select name=変数名>

<option>候補文字列 1</option>

....

<option>候補文字列 n</option>

</select>

リストボックスを表示する．

指定した候補文字列からなる選択リストができる．

（6） 送信ボタン

<input type=submit value=ボタン名>

送信ボタンを表示する．

送信ボタンはそのフォームに入力されたデータをまとめてサーバ側に送信するためのもの．

（7） リセットボタン

<input type=reset value=ボタン名>

リセットボタンを表示する．

リセットボタンはそのフォームに入力中のデータをすべて破棄 (クリア) するためのもの．

これらのフォームの要素を用いて HTML ファイルを作成する．フォームに入力し，送信ボタンを押すと，“action=” で指定した PHP スクリプトにフォームで指定した変数の値として，入力したデータが渡される．

3. 本研究による処理方法

本研究では，データベースと連携した Web アプリケーションを簡単に構築できるようにするために，具体的には，データ収集を行いデータベースに格納するということを目標としてシステムの開発を行う．そのデータ収集を行うための手段としてフォームを用いる．

まずは，どのようなフォームを出力させ，そこから得られるデータをどのようなスキーマ設計でデータベースに格納するのかというユーザ指定を行わせるためのツールとして，データ格納言語というものを定義する．つまり，システムがデータ格納言語から得る情報は，ユーザが作成したいフォームや，そこから得られるデータを格納するデータベースのテーブルのスキーマ設計である．

システムは，それらの情報に従ってまずはユーザが指定したフォームを作成するための HTML タグを表記し，データベースに接続させるための PHP プログラムも同じファイル内に埋め込んで記述する．そして，フォームから得られるデータを指定されたデータベースのテーブルの属性に挿入するための PHP プログラムも同様に同じファイルに記述する．このようにして，データ格納言語で書かれたファイルから得られるすべての情報を用いてデータ入力プログラムを自動生成する．

3.1 データ格納言語の定義

データ格納言語の一般形は以下の通りである．

ID シーケンス名 初期値 (1)

$TITLE$ タイトル (2)

$\{''$見出し$'', form(text >$ 属性名$)\}$,
$\{''$見出し$'', form(textarea >$ 属性名$)\}$,
$\{''$見出し$'', form(radio($文字列$,$文字列$,...) >$ 属性名$)\}$,
$\{''$見出し$'', form(check($文字列$,$文字列$,...) >$ 属性名$)\}$,
$\{''$見出し$'', form(listbox($文字列$,$文字列$,...) >$ 属性名$)\}$ (3)

$WITH\ MULTIPLE($属性名$)\ TO$ テーブル名 $ON\ id$ (4)

$WITH\ DUPLICATE($属性名$,...)\ TO$ テーブル名 $ON\ id$ (5)

TO テーブル名　　(6)

(1) データベースにデータを格納する際に自動的に id をふるために，データベース中にそのファイル独自のシーケンスを作成しなければならない．そのシーケンス名と初期値をこの ID 文で指定する．

(2) ブラウザのタイトルバーとページのトップに表示させる共通の語をこの $TITLE$ 文で指定する．

(3) ここが本文であり，実際に作成したいフォームの要素の指定をここで行う．

データ格納言語で用いることができるフォームの要素はテキストボックス，テキストエリア，ラジオボタン，チェックボックス，リストボックスの 5 つである．

本文は以下のような “見出し” と $form$ 文のペアからなる．

$\{''$見出し$'', form()\}$

見出しにはそのフォームの左側に表示させたい語を，そして $form$ 文中では上で述べた 5 つの要素を示す $text, textarea, radio, check, listbox$ のいずれかを指定する．

それぞれの記述方法を以下に示す．

フォーム	$form$ 文
テキストボックス	$form(text >$ 属性名$)$
テキストエリア	$form(textarea >$ 属性名$)$
ラジオボタン	$form(radio($文字列$,$ 文字列$, ...) >$ 属性名$)$
チェックボックス	$form(check($文字列$,$ 文字列$, ...) >$ 属性名$)$
リストボックス	$form(listbox($文字列$,$ 文字列$, ...) >$ 属性名$)$

属性名には，フォームに記入されたデータを格納したいテーブルの属性名を記述する．

$text, textarea$ を指定するときには，装飾子というフォームの大きさを指定する演算子を用いることができる．

ソースは以下のように一行ずつ改行して記述する．

$\{''$見出し$'', form()\},$
$\{''$見出し$'', form()\},$
　　:
　　:
$\{''$見出し$'', form()\}$

(4) チェックボックスは複数チェックすることができるので，一つの属性に対して値が複数存在する多値属性が生じてしまう可能性がある．そこで，id とその属性だけからなるテーブルを別に作成しなければならない．そのテーブル名と対象属性名をこの $WITH\ MULTIPLE$ 文で指定する．

(5) テーブル名と属性名をこの $WITH\ DUPLICATE$ 文で指定することによって，ここで指定した属性からなるテーブルに冗長的にデータを格納することができる．必要に合わせて好みの属性からなるテーブルを作成することができる．

$WITH\ DUPLICATE$ 文は書いても書かなくてもどちらでもよい．

(6) チェックボックスを除いたすべてのフォームに入力されたデータを格納するためのテーブルをこの TO 文で指定する．

＜注意すべき点＞

- 必ず一行ずつ改行して記述する
- 番号の順番にしたがって記述する
- (1),(2),(3),(6) は必ず表記する
- (4) はチェックボックス $form(check($文字列$, ...) >$ 属性名) を用いる際には必ず表記する

3.2 システムの実現

3.2.1 アルゴリズム

3.1 で定義したデータ格納言語で書いたファイルを一行ずつ順番に読み込んでいき，構文解析を行う．構文解析を行った後，得られた情報をもとにフォームを作成するための HTML タグを表記し，HTML ファイルを作成する．その際データベースに接続させるための PHP プログラムも同じファイル内に埋め込んで記述する．このようにしてできたデータ入力プログラムをブラウザを通して見ることによってフォームができる．ユーザがフォームにデータを入力し，送信ボタンを押すと，読み込みファイル (データ格納言語で書いたもの) で指定したテーブルにデータが挿入されることになる．

3.2.2 構文解析

構文解析の方法を簡単に以下に示す．

- ’ID’ を見つけると，すぐ右隣りの空白から 2 番目の空白までにある語をシーケンス名，2 番目の空白から 3 番目の空白 (最後) までにある語を初期値として判断する．そして，“$create\ sequence$ シーケンス名 $start$ 初期値” という sql 文を作る．
- ’TITLE’ を見つけると，すぐ右隣りの空白から 2 番目の空白 (最後) までにある語をタイトルとして判断する．
- ’{’ を見つけると，すぐ右隣りの’”’ から’”’ までにある語を見出しとして判断し，同じ行に’form’ を見つけると，すぐ右隣りの’(’ から’>’ または’(’ までにある語をフォームタイプとして判断する．フォームタイプには text,textarea,radio,check,listbox の 5 種類あり，フォームタイプに応じてその後の構文解析のプロセスが異なる．’>’ のすぐ後にはフォームから得られたデータの挿入先の属性名が書かれているので，’>’ から’)’ までにある語を属性名として判断する．
- ’*’ を見つけると，その後は装飾子であると判断する．装飾子ではフォームの横幅や縦幅を指定しているので，それぞれ’width=’,’height=’ から’,’ または’}’ までにある語を横幅や縦幅の値として判断する．
- ’WITH’ を見つけると，同じ行に’MULTIPLE’

か'DUPLICATE' がないか探す．'MULTIPLE' を見つけると，その行には多値属性になってしまう属性が一つだけ書かれているので，すぐ右隣りの'(' から')' までにある語を属性名として判断する．一方，'DUPLICATE' を見つけると，その行にはユーザが冗長的に作りたいテーブルの属性のリストが書かれているので，',' または')' までにある語を属性名として判断し，属性名のリストが得られる．それぞれ同じ行に'TO' を見つけると，すぐ右隣りの空白から次の空白までにある語をサブテーブル名として判断する．

- 'TO' を見つけると，すぐ右隣りの空白から 2 番目の空白 (最後) までにある語をメインテーブル名として判断する．

3.3 データベースとの連携

本システムにおいて重要なポイントとなるのは，データベースとの連携部分である．

本研究では，ユーザは自分のデータベースの内容を知った上でシステムを利用するという前提でシステムの構築を行った．つまり，ユーザはその時点での自分のデータベース中のテーブルの内容を把握した上で，どのような属性からなるテーブルを作るかというスキーマ設計を行い，フォームから得られるデータの格納方法を決定し，データ格納言語によるファイルを作成する．

システムは，ファイルを読み込んでいく過程でテーブルと属性に関する情報が得られると，データベースに接続し，データベース中にその名前のテーブルがすでに存在しないか調べる．そして，データベース中に同じ名前のテーブルが存在しないときのみテーブルを作成する．そのときに必要となるのが，先程も述べたテーブル名と属性名のリストの情報である．テーブルが存在しないとき，*"create table* テーブル名 *(id integer,* 属性名 *text,* 属性名 *text, ...);"* という sql 文を実行し，テーブルを作成する．

また，*ID* 文 を読み込んで得られるシーケンスの情報に関しても同様のことを行う．同じ名前のシーケンスが存在しないとき，*"create sequence* シーケンス名 *start* 初期値*;"* という sql 文を実行し，シーケンスを作成する．

ここまでが，データ入力プログラムを自動生成するためにシステムが行う前処理である．

3.4 データ入力プログラムの生成

データ格納言語で書かれたファイルを読み込む過程で得られたフォームに関する情報を用いて HTML ファイルを作成する．その後，同じファイル中に PHP プログラムも埋め込むように記述する．そのときに注意すべきことを以下に示す．

データ格納言語では，WITH MULTIPLE 文，WITH DUPLICATE 文，TO 文を用いることにより，結果的に複数のテーブルを扱うことになる．それらすべてのテーブルに挿入するデータに一貫性を持たせるために，それぞれのテーブルに同じ id を与えなければならない．そのための PHP の記述を以下に示す．

- メインテーブルにデータを挿入するための sql 文

$sql = " insert into テーブル名 *values (nextval('* シーケンス名 *'),' ");*
*$sql .= $*属性名*. " ' ");*
*$sql .= " ,' ". $*属性名 *. " ' ");*
:
:
$sql .= ") ");

- サブテーブルにデータを挿入するための sql 文

$sql = " insert into テーブル名 *values (currval('* シーケンス名 *'),' ");*
*$sql .= $*属性名*. " ' ");*
*$sql .= " ,' ". $*属性名 *. " ' ");*
:
:
$sql .= ") ");

このように記述することで，同じデータについては同じ id が与えられることになる．

4. システムの使用例

1. 章で述べたように，最近よく目にするようになった Web での企業の採用/就職におけるエントリーを題材として使用方法の一例を挙げる．

企業が，氏名・ふりがな・性別・住所・電話番号・メールアドレス・希望職種・希望勤務先・コメントの 9 項目からなるエントリーフォームを作成し，得られた情報をデータベースに格納したいとする．それぞれの項目から得られたデータは，以下の属性に格納することにする．

項目	属性
氏名	name
ふりがな	kana
性別	sex
住所	address
電話番号	tel
メールアドレス	e_mail
希望職種	job
希望勤務地	area
コメント	memo

データを格納するテーブルのスキーマ設計は以下の通りである．

ENTRYFORM(id,name,kana,sex,address,tel,e_mail,area,memo)

EMPLOYEE_INFO(id,name,kana,address,tel)

JOB(id,job)

フォーム中にチェックボックスを用いるとき，チェックボックスは複数チェックされる可能性があり，一つの属性に対して

値が複数できる場合があるので，テーブルを別に作らなければならない．このとき，チェックボックス以外のすべての属性を持つテーブルにデータを格納するときに id を与えておき，チェックボックスの属性を持つテーブルにも同じ id を与えるようにすれば，これを外部キーとして SQL 文で JOIN を行うことができる．チェックボックスのテーブル名の指定は **WITH MULTIPLE** 文で行う．

企業がこのデータを後に従業員情報として利用したい場合，希望職種や希望勤務先，コメントなどのデータはいらなくなることが予想される．そのようなことを見据えた上で，必要とするデータだけからなるテーブルを別に作っておけば，データが必要でなくなったときに必要でなくなった属性を含むテーブル自体を削除するだけでよく，データの修正も簡単に行える．好みの属性からなるテーブルを作るときの属性名とテーブル名の指定は **WITH DUPLICATE** 文で行う．

以上のことを行うためのソースを以下に示す．このファイルを読み込むことによって PHP データ入力プログラムが生成され，それをブラウザを通して見ると図 1 のようになる．データがテーブルに格納された様子を図 2 に示す．

```
ID SERIAL 100
TITLE エントリーフォーム
{" 氏名：",form(text>name)},
{" ふりがな：",form(text>kana)},
{" 性別：",form(radio(男, 女)>sex)},
{" 住所：",form(text>address)*{width=40}},
{" 電話番号：",form(text>tel)},
{" メールアドレス：",form(text>e_mail)},
{" 希望職種：",form(check(SE, 営業, 経理)>job)},
{"希望勤務先:",form(listbox(東京, 大阪, 東北, 九州)>area)},
{" コメント:",form(textarea>memo)*{height=5,width=30}}
WITH MULTIPLE(job) TO JOB ON id
WITH DUPLICATE(name,kana,address,tel) TO EMPLOYEE_INFO ON id
TO ENTRYFORM
```

図 2 より，同じデータに対しては同じ id がふられていることがわかる．次の人がこのフォームに入力すると 101 番の id がふられることになる．

5. 評　　価

5.1 PHP との比較

本研究の評価をするにあたり，定義したデータ格納言語と PHP との比較を行う．比較のためのひとつの指標として，今回定義したデータ格納言語で書いた読み込みファイルとシステムを通して得られたデータ入力プログラム (PHP プログラム) を用いる．

ソースの長さの異なる 4 種類のファイルを実行し，それらの読み込みファイルの行数と得られた出力ファイルの行数を表にして以下に示す．(実行例 1 は 4. 章で用いた例である)

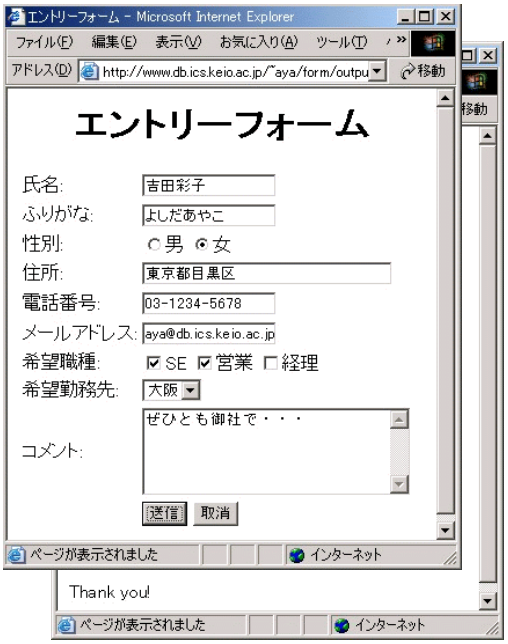

図 1　エントリーフォーム (入力・送信済み)

テーブル 1 : ENTRYFORM

id	name	kana	sex	address	tel
100	吉田彩子	よしだあやこ	女	東京都目黒区	03-1234-5678

e_mail	area	memo
aya@db.ics.keio.ac.jp	大阪	ぜひとも御社で …

テーブル 2 : EMPLOYEE_INFO

id	name	kana	address	tel
100	吉田彩子	よしだあやこ	東京都目黒区	03-1234-5678

テーブル 3 : JOB

id	job
100	SE
100	営業

図 2　テーブルにデータが格納された様子

表 1　ソースの行数比較

	データ格納言語 (入力)	データ入力プログラム (出力)
実行例 1	14	161
実行例 2	12	138
実行例 3	18	225
実行例 4	52	621

これより，データ格納言語のソースの行数がデータ入力プログラムのソースの行数の 10 分の 1 以下になっていることが読み取れる．このデータ入力プログラムは本システムから得られるものなので，同じことをもっと簡潔に少ない行数で書ける可能性もある．しかし，たかだか十数行でフォームを出力し，データベースに接続してデータを格納するということは不可能であろう．よって，本システムによってデータ収集アプリケーショ

ンの開発の際に必要なソースの行数が大幅に減少したと言うことができる．

現段階では，データベースの既存の値を用いてフォームの作成を行なうことはできない．しかし，チェックボックスやラジオボタンの値はデータベース中に存在するものを自動で表示させたい場合もあると考えられるので，今後データベース中の既存のデータとの連携も行なえるようにする必要がある．

5.2 関連研究との比較

5.2.1 ColdFusionとの比較

「Macromedia ColdFusion(コールドフユージョン)」とは，米国 Allaire 社が開発した Web アプリケーション開発ツールであり，イントラネット構築，サイト構築などデータベースと連携した Web アプリケーションを開発する場合の RAD（迅速な開発）ツールとして現在広く用いられている．

ColdFusion での Web アプリケーション開発には，CFML(ColdFusion Markup Language) と呼ばれる HTML を拡張したタグ言語を使用してプログラムする．CFML は HTML タグに非常によく似た言語を使用し，表示デザイン部は HTML のタグを使用する．

ColdFusion では様々なことが行えるが，本研究で取り組んだ “データ入力” ということのみに着目して以下の 3 点について比較を行う．

- データ格納言語と CFML の記述面での難易度

データ格納言語では form 文を一行書くだけでフォームが一つできるが，それに対し，CFML では表示部分は HTML でデザインするので，表示内容を一つ一つタグで囲みながら記述しなければならない．行数が長くなればなるほど，タグの対応関係を考えなければならなくなり，記入ミスということも起こり得るので，その点ではデータ格納言語の方が優れていると言えるだろう．

また，根本的な問題であるが，CFML は HTML を書けることが前提となっている言語である．それなので，HTML がわからない人にとっては，言語定義が少なく簡素であり，書かなければならないソースの行数も少なくて済むデータ格納言語のほうが難易度が低いと感じるかもしれない．

- レイアウトの柔軟性

データ格納言語では，現時点では表示時のレイアウトは限られており，ユーザ指定などはあまり行えない状態である．それに比べ，CFML は表示部分は HTML でデザインするので，HTML が書ける人ならば好きなようにレイアウトを指定して出力させることができる．よって，レイアウトの柔軟性は CFML の方が高いと言える．

- データの取扱い

CFML では，HTML で表示部分を記述する際に，データを渡す先のページを <FORM ACTION=”xxxxx.CFM” METHOD=”POST”> で指定し，HTML 入力ページを作成する．すると，入力フォームから入力されたデータは xxxxx.CFM に渡される．CFM という拡張子は ColdFusion のページ (テンプレート) であることを示しており，CFM の拡張子を持ったページは ColdFusion のアプリケーションサーバーで解釈され，DB との通信を行う．xxxxx.CFM には，<CFINSERT DATASOURCE=”データソース名” TABLENAME=”テーブル名”> と書き，HTML も同じページに記述することができる．<CFINSERT> は CFML タグの一つで，登録したデータソースのテーブルに，入力フォームから入力されたデータを挿入する．

データ格納言語では，<FORM ACTION=”xxxxx” METHOD=”xxxxx”> という一文はデータ入力プログラム生成時に自動的に記述されるので，わざわざ記述する必要がない．また，入力フォームから得られたデータを一括してテーブルに挿入するというところは CFML と同じであるが，データ格納言語ではデータ挿入に関する特別な指定も行える．その指定は WITH MULTIPLE 文や WITH DUPLICATE 文で行う．本システムでは，WITH MULTIPLE 文で多値属性に対処したり，ひとつの入力フォームから得られるデータをまとめて扱うのではなく個々に扱うことによって，必要なデータのみを好きなテーブルに冗長的に挿入することを WITH DUPLICATE 文で行えるようにした．これは，本システムの大きな特徴である．

5.2.2 Microsoft Accessとの比較

Microsoft Access とは，データの管理や操作が簡単にできるように設計されたリレーショナルデータベースソフトであり，様々なツールが存在する．

Microsoft Access では様々なことが行えるが，本研究で取り組んだ “データ入力” ということのみに着目して比較を行う．

Microsoft Access では，データベースのテーブルの作成や複数のテーブルの関連付けなどが簡単に行え，またデータ入力を行うためのフォームを簡単に作成することができる．本研究の目的の一つであった Web でのアンケートによるデータ収集というものも行えるが，Microsoft Access のフォームには複数選択できるチェックボックスが存在していないなどアンケートを作成するには不向きである．複数選択可能なチェックボックスが存在しないということは，ひとつのフィールドに複数の値が存在する多値属性に対処していないということになる．アンケートに限らなくとも，データを登録する際にひとつのフィールドに複数の値を格納する必要性は存分にあると考えられるので，そのような機能は必要である．本システムでは，多値属性の問題を WITH MULTIPLE 文を用いて解決したので，チェックボックスをたくさん含む大規模なアンケートなどにも簡単に対処できる．多値属性のデータを扱えるという点で Microsoft Access よりも優れていると言える．

6. ま と め

本研究では，Web ベースのデータベースアプリケーションを専門的な知識がなくても簡単に構築できるようにすることを目

的として，独自のデータ格納言語を定義した．そして，これからデータ入力を行う PHP 入力プログラムを自動生成するシステムを開発した．

定義したデータ格納言語の記述の容易さは，評価の際に行った PHP での記述との比較で実証された．このシステムを用いることで，PHP などのプログラミング言語を書かなくても簡単にデータをデータベースに格納することができることが示された．また，アンケートによるデータ収集の際に必要となる多値属性のデータへの対処を行った点で，他のツールよりも優れているということを示すことができた．

現時点では，一度データベースに格納してしまったデータの修正・更新は行なえない．また，データベースの既存のデータを，フォーム中のリストボックスやチェックボックスの値として用いるといったデータベースとの連携は行なえない．しかし，それらの機能は最低限必要なものであると考えられるので対処したい．また，現時点では単独画面には対応できるが，複数ページからなる大規模アプリケーションの構築には対応できない．今後はそれらも視野に入れてシステムを拡張していきたい．

文　　献

[1] PHP: http://www.php.net/

[2] クレイグ・ヒルトン, ジェフ・ウィルス “PHP による Web データベース構築” ピアソン・エデュケーション

[3] 堀田倫英, 石井達夫, 廣川類 “PHP4 徹底攻略 ー Web とデータベースの連携プログラミング” ソフトバンク パブリッシング株式会社

[4] 西沢 直木 “PHP による Web アプリケーション スーパーサンプル” ソフトバンク パブリッシング株式会社

[5] 石井達夫 “PostgreSQL 完全攻略ガイド” 技術評論社