

プラグイン可能なアンケートサーバの作成

田村 直之^{*1} 重村 哲至^{*2}

Development of questionnaire server with plug-in function

Naoyuki TAMURA ^{*1} and Tetsuji SHIGEMURA ^{*2}

Abstract

This paper suggests a server system to make and send out a questionnaire effectively by using the Internet. The system is a kind of Web application that has a function of making, answering and collecting a questionnaire. Also, as a plug-in it can add a special function of tallying up the result of the questionnaire. This function makes it possible for this system to be used for other purposes.

Key Words : plug-in, Java, Web application, XML

1. まえがき

本研究では、インターネットを利用して効率よくアンケートを行うサーバシステムを提案する。

本システムは、Web ブラウザを用い、アンケートの作成、回答、集計結果の表示をすることが可能なように Web アプリケーション¹⁾として実装する。このような Web ベースのシステムでは、Web ブラウザが動く環境さえあれば、インターネットを通じて使用できるメリットがある。また特別なソフトウェアをクライアントにインストールする必要もないため、誰でも簡単にシステムを利用することができる。

また、本システムは、アンケートの内容に依存する特殊な集計方法に対応するため、集計機能をプラグインとして追加することができる。この機能により、アンケート以外の目的にも、応用が可能である。

2. システム構成

2.1 全体像

本研究で作成するアンケートサーバのシステム構成を図1に示す。

本システムは Web ベースのクライアントサーバシステムである。サーバ PC には、Java[®]の開発環境 (JDK)をインストールする。JDK 上で、サーブレットコンテナ(Tomcat)と XML - DB[®](Xindice)を稼働させ、アンケートサーバを実現する。クライアントは、Web ブラウザを使用できる PC ならどのようなものでも良い。

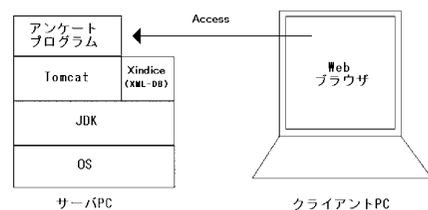


図1 システム構成

*1 情報電子工学専攻

*2 情報電子工学科

2.2 Tomcat

Tomcat は Java サブレットを動かすことのできる Web サーバの一種である。サブレットは、Java で書かれた CGI のようなものである。本研究で作成するシステムは Tomcat 上で実行されるサブレットとして実装される。

2.3 XML-DB

本研究では、扱うデータを全て XML 形式で保存する。Xindice は、XML²⁾形式のデータを格納するためのデータベース管理システム (XML - DB) の一種である。

XML - DB は、XML 文書をそのまま保存できる。リレーショナル DB に XML 文書を保存するには、XML 文書を細かく分解し、分解したデータをカラムにマッピングする必要があった。この方法では、複雑な XML 文書を保存することは困難な上、XML 文書内の複雑な条件の検索も難しい。しかし、XML - DB は、XML 文書を分解することなくそのまま保存するので、XML 文書内の複雑な条件の検索を高速実行することができる。また、複数の XML 文書を同時に処理することができる。

3. 本システムの特徴

3.1 プラグイン可能

本システム最大の特徴は、アンケート集計用プラグインの追加が可能なシステムである。この集計用プラグインについては第 5 章で詳しく説明する。

3.2 XML データ形式

本システムは、全てのデータを XML 形式で保存する。このメリットは以下の 2 つである。

1) 他システムとのデータの連携が容易

将来的に複数のアンケートサーバ間、または、アンケートサーバと MS - EXCEL 等のツール間でデータをやりとりする可能性も考えられる。この時、データを XML で記述しておけば、システム間で直接データのやりとりができるので便利である。

2) データの表現が柔軟

本システムでは、アンケートの内容をそのまま保存するため、柔軟性に富んだデータ保存形式が必要である。XML でデータを保存すれば、何重にも構造化されたデータを扱うことも容易である。

4. アンケート手順

本システムを用いてアンケートを行う手順は、アンケート作成、アンケート回答、アンケート集計に分けることができる。これらについて順に説明していく。

4.1 アンケート作成

アンケート作成者は、Web ブラウザを用い、本システムにアクセスする。図 2 に示すログイン画面が表示されるので、ユーザ名とパスワードを入力し、ログインする。

次に図 3 に示すアンケート作成画面でアンケート内容を入力し、アンケートを作成する。

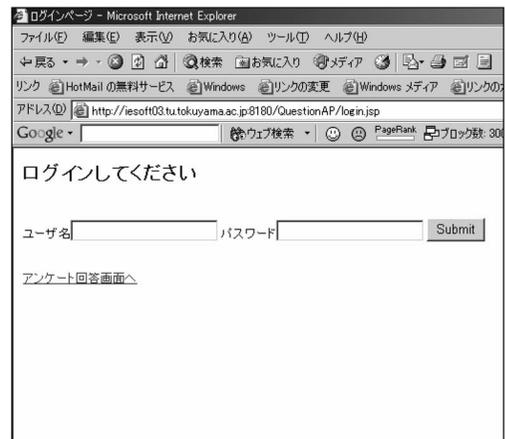


図 2 ログイン画面

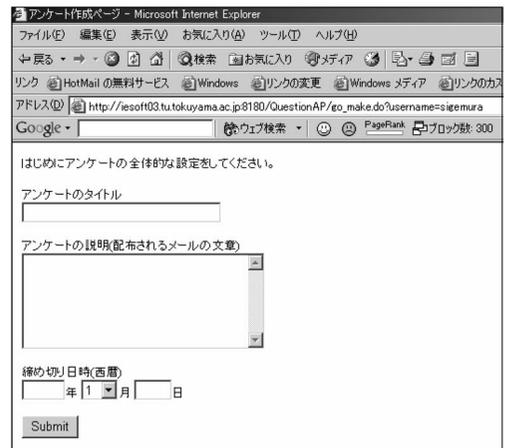


図 3 アンケート作成画面

アンケート作成手順は以下の通りである。

手順1：アンケートタイトルの設定

アンケート全体のタイトルを設定する。

手順2：締め切り日時の設定

アンケートの締め切り日時を設定する。この日時以降は、回答を受け付けない。

手順3：質問の内容の設定

質問ごとに質問文を入力する。

手順4：回答形式の設定

質問ごとにアンケート回答者がどのような形式でアンケートに回答するかを設定する。選べる形式は、以下の6通りがある。

- ・二者択一選択形式(「はい」、「いいえ」)
- ・多者単数選択形式(いくつかの選択肢の中から1つを回答する形式)
- ・多者複数選択形式(いくつかの選択肢の中から2つ以上を回答する形式)
- ・自由記述形式
- ・数字入力形式
- ・N段階評価形式

手順3, 4は質問ごとに設定するので、アンケート内の全ての質問について繰り返す。

手順5：集計方法の設定

集計方法は次のどちらかを選択する。

①標準の集計

②集計用プラグインによる集計

①を選んだ場合には、各質問の回答形式に沿った標準の集計方法で集計が行われる。

②を選んだ場合は、指定した集計用 Java クラスによりアンケート全体の集計が行われる。なお、作成した集計用 Java クラスファイルはシステム管理者に渡す必要がある。

手順6：アンケート内容の確定

最後に Web ブラウザ上の確定ボタンを押すと、作成したアンケート内容が XML - DB に保存される。

このように、アンケートを実施したい人は、質問とその回答形式を指定するだけで、簡単にアンケートを作成することができる

4.2 アンケート回答

まず、アンケート回答者は、アンケート回答ページにアクセスし、アンケート ID を入力する。

すると、図4のような回答用画面が自動的に表示されるので、質問に回答する。全ての質問に回答後、submit ボタンを押すと、アンケートの回答は完了する。回答はXML-DBに保存される。

4.3 アンケート集計

本システムでは、アンケート結果は自動的に集計される。アンケートの集計結果を見ることができるのは、アンケート作成者だけである。

実際のアンケート結果表示画面を図5に示す。この画面は、アンケート配布後はいつでも見ることができる。ただし、この画面を表示するまでに回収されたアンケート結果が集計され表示される。

また、プラグイン形式での集計が指定されている場合は、指定した Java クラスによる集計を行った結果を図5の代わりに表示される。

図4 アンケート回答画面

図5 アンケート結果表示画面

5. 集計用プラグインについて

5.1 概要

集計用プラグインは、アンケート作成者がアンケートの集計方法を Java で記述できるようにしたものである。集計用プラグインを利用すると、次の様なことが実現可能である。

(1) 複雑な集計方法の実現

例えば、「質問 1 で A を選択した人の中で、質問 2 で B を選択した人は何人いますか？」というような複雑な条件で集計する必要性が生じると予想される。これらの複雑な集計方法をシステム側で全て用意するのは不可能である。このため、アンケート作成者が自由に集計方法を記述できるようにすることが必要である。

(2) 集計結果の加工

集計用プラグインを利用すると、例えば、集計結果の分析やグラフ化をすることもできる。

分析やグラフ化の Java プログラムを集計クラス内に記述することで実現可能である。

5.2 実現方法

集計用プラグインの実現方法は、集計機能を実装した Java クラスをサーブレットコンテナにロードさせることで実現することができる。

作成する集計用クラスの一例を図6に示す。この例では、システムに用意された **Abstract** クラス **count** を継承して、集計用クラス **mycount** を作成している。

ここでは、**my_count** クラスの **do_count** メソッドにより、質問 **q1** で A と答えた人と、質問 **q2** で B と答えた人の合計を求め、Web ブラウザに表示している。

```

Class mycount extends count{
do_count(java.io.Write out){
    /* q1:質問 ID */
    int q1 = 20040906001;
    //質問 q1 で A と答えた人数は?
    int n = ans_count(q1," A" );
    //質問 q1 で B と答えた人数は?
    int m = ans_count(q1," B" );
    //合計を表示する。
    out.write(n + m);
}
}

```

リスト 1 作成した集計用 Java クラス

このように、あらかじめ用意されていない方法で集計したい場合は、システムで用意した集計クラス(**count**)を継承し、用意されたメソッド(**ans_count**等)を用いて集計方法を **do_count** メソッドに記述することができる。

用意されているメソッドを以下に示す。

- **String answer**(質問 ID, 回答者)
その回答者が答えた回答を返すメソッドである。ここで質問 ID は、アンケート番号とアンケート内の質問番号を連結した整数である。
- **int ans_count**(質問 ID)
その質問を回答した人数を返すメソッドである。
- **int ans_count**(質問 ID, 選択肢)
その選択肢を回答した人数を返すメソッドである。
- **int ans_count**(質問 ID, 選択肢, 人の集合)
人の集合の中から、その選択肢を回答した人数を返すメソッドである。
- **String[] ans_people**(質問 ID)
その質問を回答した人の集合を返すメソッドである。人の集合は **String** 配列で返す。
- **String[] ans_people**(質問 ID, 選択肢)
その選択肢を回答した人の集合を返すメソッドである。
- **String[] ans_people**(質問 ID, 選択肢, 人の集合)
人の集合の中から、その選択肢を回答した人の集合を返すメソッドである。
- **int num_count**(質問 ID, 数字, 判別条件)
数字入力方式、N 段階方式の質問に対してのみ使用できるメソッドである。判別条件を満たした回答をした人数を返す。使える判別条件は >, >=, =, <=, < などがある。
- **String[] num_people**(質問 ID, 数字, 判別条件)
数字入力方式、N 段階方式の質問に対してのみ使用できるメソッドである。判別条件を満たした回答をした人の集合を返す関数として機能する。

- double average(質問 ID)
数字入力方式, N段階方式の質問に対してのみ使えるメソッドである。回答の平均値を返す関数として機能する。
- int sum(質問 ID)
数字入力方式, N段階方式の質問に対してのみ使えるメソッドである。回答の合計を返す関数として機能する。

これらのメソッド内部で XML-DB にアクセスし、結果を返す仕組みになっている。アンケート作成者はメソッド内部の動きを知る必要はない。

アンケート作成者は、次の do_count メソッドをオーバーライドすることで集計方法を記述する。

- void do_count(出力ストリーム)
集計ページがアクセスされた時、実行されるメソッドである。do_count の中から ans_count 等のメソッドを呼び出すことで、集計方法を記述する。集計結果は、出力ストリームに書き出すことにより、Web ブラウザに表示される。

5.3 実例

集計用プラグインを用いて実際にどのようなことができるかを説明する。

1) 絞込み集計

「質問 q1 で A を選択した人の中で、質問 q2 で B を選択した人の人数」を調べたい時はリスト 2 の Java プログラムで実現できる。

```

1:void do_count(java.io.Write out){
2:    String[] people;
3:    int num;
4:    int q1 = 20040906001;
5:    int q2 = 20040906002;
6:    people = ans_people(1, "A");
7:    num = ans_count(2, "B", people);
8:    out.write("<H2>");
9:    out.write("該当者は" + num + "人");
10:   out.write("</H2>");
11: }
```

リスト 2 Java による集計手順の例

6 行目の ans_people メソッドで、質問 q1 で A と答えた人の集合を取り出す。7 行目の ans_count により、その中で質問 q2 で B と答えた人数を取り出す。8 行目から 10 行目で集計結果を HTML 形式で Web ブラウザに出力する。

このように、用意されたメソッドを組み合わせることで、複雑な集計方法を簡単に記述することができる。

2) アンケート以外への応用

学校で本システムを宿題提出システムとして利用することができる。いくつかの問題を書いた宿題をアンケートとみなし、また集計用プラグインを採点機能を持つものにより、集計時に自動採点可能な宿題提出システムとして応用する等が考えられる。

自動採点システムの例をリスト 3 に示す。

```

1:void do_count(java.io.Write out){
2:    String[] people;
3:    String ans;
4:    int q1 = 20040906001;
5:    int q2 = 20040906002;
6:    int q3 = 20040906003;
7:
8:    /*q1 の回答者の集合を取り出す*/
9:    people = ans_people(q1);
10:
11:   /*q1 の回答者全員を採点*/
12:   for(int i=0; i<people.length; i++){
13:       /*回答を取り出す*/
14:       ans = answer(q1, people[i]);
15:       out.write(people[i] + "さん");
16:       /*q1 の回答が A と同じかどうか*/
17:       if(ans.equals("A")){
18:           out.write("正解");
19:       }else{
21:           out.write("間違い");
22:       }
23:   }
24:   /*q2, q3 についても同様*/
25:   .
26:   .
27: }
```

リスト 3 自動採点システムの例

9行目の `ans_peole` メソッドで `q1` の回答者の集合を取り出し、12行目からの回答の採点をする。もし回答が `A` ならば“正解”，それ以外なら“間違い”と表示する。`q2`, `q3` についても同様に採点をする。

6. おわりに

インターネットで使用できるアンケートサーバを提案した。このアンケートサーバは、Web ブラウザを用い、アンケートの作成から集計までの全ての操作が可能なものである。また、プラグインの追加機能により複雑な集計が可能になり、さまざまな応用が考えられる。

文献

- 1) James Goodwill: “マスタリング Jakarta Struts”, 翔泳社(2003)
- 2) 坂田 健二: “Windows で学ぶ XML”, 技術評論社(2002)
- 3) @IT: “Xindice : 無料で使える XML - DB”, <http://www.atmarkit.co.jp/fxml/tanpatsu/18xindice/xindice01.html>
- 4) 越田一朗: Java によるオブジェクト指向プログラミング入門, 培風館(1998)

(2004. 9. 6 受理)