

教材ソフト開発の為にイントラネットを用いた自動アンケートシステム

大 西 孝 臣*

Automatic questionnaire system with intranet for developing the instructional software

Takaomi OHNISHI

Abstract

An automatic questionnaire system with intranet is introduced in this article. This system is suitable for collecting the students' opinions about the instructional software for the senior grader's experimentation, because the number of the members of the senior grader's experimental groups by a week is much less than that of junior grader's.

1. はじめに

筆者は本校高学年の実習用の教材ソフトを開発¹⁾し、当教材ソフトに関する学生の意見を求めている。しかし高学年の実習の体制は毎週少人数の学生が入れ替わるものであり、さらに筆者は実習の実施時には他の学年の実験を担当していた為、筆者自身が直接アンケート調査を行う事ができず、学生の意見を取り入れるのに苦慮していた。そこで本稿では、開発した、学生各自の実習終了時に自動的にアンケート調査を行うシステムを紹介する。

2. システムの概要

本稿で紹介するシステム機器および必要なファイル群を図1に示す。

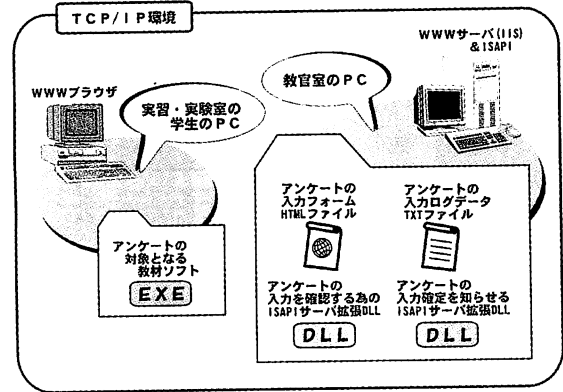


図1 システム機器・ファイル群の構成

TCP/IPに従うイントラネット上に、教官用と学生用のPCがあり、教官用のPCにはWindows2000 Ser.とISAPI²⁾と呼ばれるAPIを用いて拡張できるWWWサーバであるIIS²⁾が起動している。

3. コンテンツの実現と実行の様子

3. 1 Windowsアプリケーションからの起動

学生のPCにて教材ソフトが終了すると、自動的にWWWブラウザが起動し、次節に示すアンケートの入力フォーム(図2、図3)が表示される。

Visual C++6.0により作成した、教材ソフトのリストの一部を次に示す。

```
// Mainfrm.cpp : CMainFrame クラスの動作の定義を行います。
//
...
CMainFrame::CMainFrame() // デストラクタ
{
    STARTUPINFO si;
    PROCESS_INFORMATION pi;
    ZeroMemory(&si, sizeof(si));
    si.cb=sizeof(si);
    CreateProcess(
        "c:\\Program Files\\VPPlus\\VPMicrosoft Internet\\Iexplore.exe",
        "iexplore.exe http://ohnishi.jo.tomakomai-ct.ac.jp/questionnaire/questionnaire.html",
        NULL, NULL, FALSE, CREATE_NEW_CONSOLE|NORMAL_PRIORITY_CLASS, NULL,
        NULL, &si, &pi
    );
    ...
}
```

Win32APIのCreateProcess関数³⁾により、学生用のPCにあるInternet Explorerを起動させ、教官用のPCのWWWサーバからアンケートの入力フォームHTMLファイルをロードする。

3. 2 アンケートの入力フォーム

学生用のPCにてアンケートの入力フォームHTMLファイルがWWWブラウザ上に表示される

* 助 手 情報工学科

様子を図2、図3に示す。

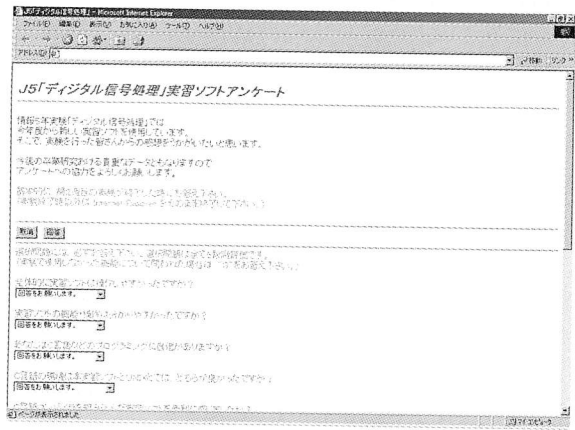


図2 アンケート入力フォームの初期画面

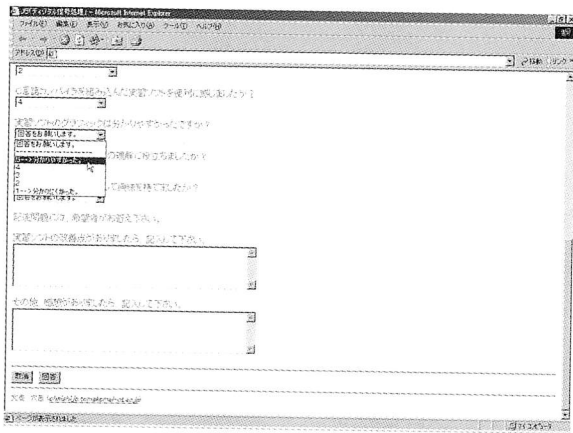


図3 5段階尺度による評価の様子

図2、図3に示すフォームでは、5段階尺度による設問8問と記述による設問2問を用意した。ソースとなるHTMLファイルのリストの一部を次に示す。

```
<html>
<title>J5「デジタル信号処理」</title>
...
<form action="http://ohnishi.jo.tokomai-ct.ac.jp/questionnaire/questionnaire.dll" method="post">
<input type="HIDDEN" name="MfcISAPICommand" value="reply">
<hr><input type="reset" value="取消"> <input type="submit" value="回答"><hr>
...
全体的に実習ソフトは操作しやすかったですか？<br>
<select name="alt_ans1" size="1">
<option value="0">回答をお願いします。
<option value="0">
<option value="5">5→操作しやすかった。
<option value="4">4
<option value="3">3
<option value="2">2
<option value="1">1→操作しにくかった。
</select><br><br>
... (ここで選択問題のフォームを7回繰り返し) ...
実習ソフトの改善点がありましたら、記入して下さい。<br>
<textarea cols=60 rows=5 name="txt_ans1" value=""></textarea><br><br>
... (ここで記述問題のフォームを1回繰り返し) ...
<hr><input type="reset" value="取消"> <input type="submit" value="回答"><hr></form>
...
</html>
```

フォームでの回答ボタンが押されると、学生による5段階尺度の回答はalt_ans1からalt_ans8までの変数に、記述による回答は変数 txt_ans1と

txt_ans2にそれぞれ格納され、アンケートの入力を確認する為のISAPIサーバ拡張DLL²⁾であるquestionnaire.dllにpostされる。

3. 3 アンケート入力の確認

入力フォームに回答すると、ISAPIサーバ拡張DLLであるquestionnaire.dllからの出力として、図4に示す様な、フォームでの入力が正しく反映されたかを確認するページが表示される。

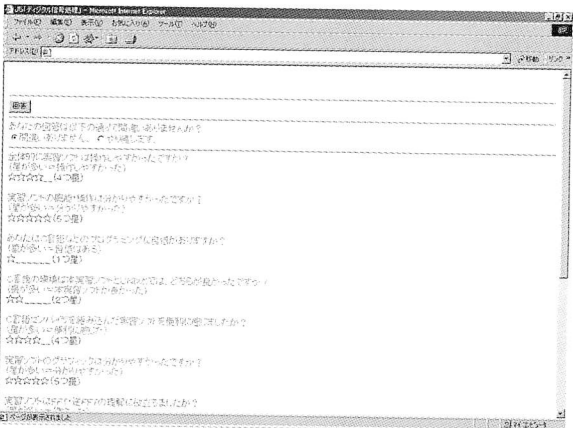


図4 アンケートの入力を確認するためのページ

Visual C++6.0 によるquestionnaire.dllのソースリストの一部を次に示す。

```
// QUESTIONNAIRE.CPP - インターネット サーバー用のインプリメンテーション ファイル
questionnaire Extension
...
void CQuestionnaireExtension::reply(CHttpServerContext* pCtxt, LPCSTR alt_ans1, LPCSTR alt_ans2, ...
{
    StartContent(pCtxt);
    *pCtxt << "<title> J5「デジタル信号処理」</title>\n";
    ...
    if((CString)alt_ans1 == "0" || ... || (CString)alt_ans8 == "0")
    {
        *pCtxt << "選択問題に未回答のがあります。<br>... ";
    }
    else
    {
        *pCtxt << "
        <form action='http://.../confirm.dll' method='post'>\n";
        *pCtxt << "<input type='HIDDEN' name='MfcISAPICommand' value='reply'>\n";
        *pCtxt << "<hr><input type='submit' value='回答'><br>\n";
        *pCtxt << "あなたの回答は以下の通りで間違いありませんか？<br>\n";
        *pCtxt << "<input type='radio' name='alt_ans1' value='1' checked='checked'>
        ありません。<br>\n";
        *pCtxt << "<input type='radio' name='alt_ans1' value='2'>やり直します。
        <br>\n";
        *pCtxt << "全体的に実習ソフトは操作しやすかったですか？<br>\n";
        *pCtxt << "（星が多い＝操作しやすかった）<br>\n";
        ...
        if((CString)alt_ans1 == "5")
        {
            *pCtxt << "☆☆☆☆☆ (5つ星)<br><br>\n";
        }
        if((CString)alt_ans1 == "4")
        {
            *pCtxt << "☆☆☆☆ (4つ星)<br><br>\n";
        }
        if((CString)alt_ans1 == "3")
        {
            *pCtxt << "☆☆☆ (3つ星)<br><br>\n";
        }
        if((CString)alt_ans1 == "2")
        {
            *pCtxt << "☆☆ (2つ星)<br><br>\n";
        }
        if((CString)alt_ans1 == "1")
        {
            *pCtxt << "☆ (1つ星)<br><br>\n";
        }
        ...
        CString cBuffer1;
        *pCtxt << "実習ソフトの改善点がありましたら、記入して下さい。<br>\n";
        ...
        cBuffer1 = (CString)txt_ans1;
        if(cBuffer1 == "")
        {
            *pCtxt << "（回答はありませんでした。）<br><br>\n";
        }
        else
        {
            cBuffer1.Replace("<br>\n", "<br>\n");
            *pCtxt << "回答：<br><br>\n" + cBuffer1 + "<br><br>\n";
        }
    }
}
```

(次ページへ続く)

(前ページからの続き)

```
...
    *pCtxt << "<hr><input type='Y' submit' value='Y'回答Y'><hr><Yn'";
...
HANDLE hMutex = CreateMutex(NULL, FALSE, "q_log.txt");
WaitForSingleObject(hMutex, INFINITE);
fp = fopen("c:\\Y\\...\\Yq_log.txt", "a");
fprintf(fp, (CString)buf +
    tNow.Format("%Y/%m/%d (%a) %H:%M:%S\\n\\n") +
    alt_ans1 + (CString)alt_ans1 + " " +
    ...
    "alt_ans8 = " + (CString)alt_ans8 + "\\n\\n" +
    "txt_ans1=\\n\\n" + (CString)txt_ans1 + "\\n\\n" +
    ... );
fclose(fp);
ReleaseMutex(hMutex);
CloseHandle(hMutex);
}
EndContent(pCtxt);
}
...
```

呼び出されたquestionnaire.dll が出力するページのコンテンツは、MFC²⁾でのISAPIに関連する基本クラスの1つであるCHttpServerContextクラス²⁾のオブジェクトへのポインタである *pCtxtに対して、HTMLに従う文字列をストリーム演算子 “<<” を使って出力する事で実現する。

コンテンツの実現の際には、post元から値を渡された変数 (alt__ans1からalt__ans8までとtxt__ans1, txt__ans2) に入っているフォームでのアンケート入力結果を反映させる。

また同時に、教官用のPCにあるアンケートの入力ログデータTXTファイル対しても、アンケートの入力結果を記録する。ログファイルを扱う際は、複数のプロセスによる競合が起きないように、WaitForSingleObject関数⁴⁾によるアクセスの同期⁵⁾を行う。

フォームでの回答ボタンが押されると、ラジオボタン「間違いありません。」を選択した場合、変数alt__ansに文字列 “1” が、ラジオボタン「やり直します。」を選択した場合、変数alt__ansに文字列 “2”が格納され、アンケートの入力確定を知らせるISAPIサーバ拡張DLLであるconfirm.dllにpostされる。

3. 4 アンケート入力確定の通知

入力確認のページにて、ラジオボタン「間違いありません。」を選択した上で回答すると、ISAPIサーバ拡張DLLであるconfirm.dllからの出力として、図5に示す様な、アンケートの入力が確定した旨のページが表示される。この入力確定のページにて「終了します」のボタンを押すと、JavaScriptにて記述した命令により、WWWブラウザが終了する。

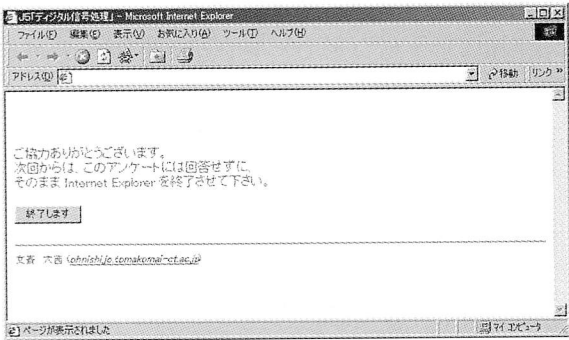


図5 アンケート入力の確定を知らせるページ

Visual C++6.0 によるconfirm.dllのソースリストの一部を次に示す。

```
// CONFIRM.CPP - インターネット サーバー用のインプリメンテーション ファイル
confirm Extension
...
void CConfirmExtension::reply(CHttpServerContext* pCtxt, LPCSTR alt_ans)
{
    StartContent(pCtxt);
    *pCtxt << "<title>J5「デジタル信号処理」</title><Yn'";
    ...
    CTime tiNow = CTime::GetCurrentTime();
    ...
    HANDLE hMutex = CreateMutex(NULL, FALSE, "q_log.txt");
    WaitForSingleObject(hMutex, INFINITE);
    fp = fopen("c:\\Y\\...\\Yq_log.txt", "a");
    if((CString)alt_ans == "1")
    {
        *pCtxt << "<script language='JavaScript'>";
        *pCtxt << "</script>";
        *pCtxt << "ご協力ありがとうございます。<br><Yn次回からは、...<br><Yn";
        *pCtxt << "そのまま Internet Explorer を終了して下さい。<br><br><Yn";
        *pCtxt << "<form ... ><input ... value='終了します' onClick='windowCloser()'>";
        *pCtxt << "</form><Yn";
        fprintf(fp, ... + "入力確認\\n\\n-----\\n\\n" );
    }
    else
    {
        *pCtxt << "アンケートに戻ります。<br><br><Yn";
        ...
    }
    fclose(fp);
    ReleaseMutex(hMutex);
    CloseHandle(hMutex);
    ...
    EndContent(pCtxt);
}
```

呼び出されたconfirm.dll が出力するページのコンテンツは、questionnaire.dll の場合と同様に、ポインタ *pCtxtに対してHTMLの文字列を出力する事で実現する。post元から値を渡された変数alt__ansの値が “1” の場合は、図5の様な、入力が確定した旨を知らせるページを表示する。“2”の場合は、アンケートの入力フォームHTMLファイルが再びロードされる様にする。

また、教官用のPCにあるアンケートの入力ログデータTXTファイル対して、変数alt__ansの値に従い、「入力確認」か「入力取消」のいずれかを記録する。

3. 5 アンケートの入力ログデータ

学生によるアンケートの入力や入力確認の結果は、図6に示す様な形で、アンケートの入力ログデータTXTファイルに記録される。

```
172.31.1.4 2000/05/22 (Mon) 16:51:27
alt_ans1 = 4 alt_ans2 = 4 alt_ans3 = 5 alt_ans4 = 3
alt_ans5 = 5 alt_ans6 = 3 alt_ans7 = 5 alt_ans8 = 4
txt_ans1 =
cpxファイルなどを関連付けしてすぐ起動できるようにする。
行番号を表示できるようにする。

txt_ans2 =
アイコンがBFCのままなので、作ってみてはいかがでしょうか。

-----
172.31.1.4 2000/05/22 (Mon) 16:51:42
入力確認

-----
172.31.1.3 2000/05/29 (Mon) 18:20:08
alt_ans1 = 4 alt_ans2 = 5 alt_ans3 = 1 alt_ans4 = 2
alt_ans5 = 4 alt_ans6 = 5 alt_ans7 = 5 alt_ans8 = 5
txt_ans1 =
プログラムのうち1つにgetch()をつけないと動かないものがあった。
txt_ans2 =
実験内容に沿ったとてもわかりやすいソフトであったが、DOS画面で
波形をプロットする処理はもう少しプロットの数が多いほうがいいです。

-----
172.31.1.3 2000/05/29 (Mon) 18:20:18
入力確認

-----
```

図6 アンケートの入力ログデータ

アンケートを提出する学生を匿名にする為に、1台の学生用のPCに対して同時に複数の学生がアクセスできないという性質を用い、アンケートの提出者の同定にはIPアドレスを用いる。

本稿の例では、アンケートを収集する学生数は数十のオーダーである為、アンケートの結果の記録についてはTXTファイルによるログデータの形で間に合った。将来のデータ数の増加に対応する為には、入力ログTXTファイルへの書き込みをAccessなどのデータベースへの入力という形に置き換える必要があると考える。

4. おわりに

本稿では、実習での教材ソフトにリンクしたアンケート収集システムを紹介した。本システムを使用する事により、高学年の実習・実験の少人数班による週ローテーション体制の様な事例での、ルーチンワークであるアンケート収集を効率よく行う事ができた。また、教材ソフトの評価という事例には限定せず、クラス単位で一斉に実施する授業アンケートや、クラスが幾つかの研究室に分

散している時のアンケート、遠隔教育でのアンケートの様な事例に対しても、本システムを適用できると考える。

参考文献

- 1) 大西 孝臣、杉岡 一郎共著、Windowsアプリケーションにコマンドラインコンパイラを編入させた実習教材の開発、教育工学関連学協会連合第6回全国大会講演論文集、pp.821-822、2000
- 2) Microsoft Developer Network ライブラリ 日本語版October 2000、Microsoft Corporation、2000
- 3) Jeffry Ritcher著、長尾 高弘訳、Advanced Windows 改訂第3版、アスキー、1997
- 4) James E. Beveridge、Robert Wiener 共著、インフォビジョン訳、Win32マルチスレッドプログラミング、アスキー、1997
- 5) 大西 孝臣、並行処理の同期に関する事例の実現、苫小牧高専紀要、No.35 pp.85-90、2000

(平成12年11月27日受理)