

講義における実用サイトの構築

Construction of a realistic Web site in education

小林 健一郎

Ken-ichiro KOBAYASHI

(平成20年10月8日受理)

要旨

本論文では、「講義内での実用可能なWebサイトの構築」を議論し、半期講義において、十分実用的なサイトが構築可能であることを示す。

1. はじめに

「教育におけるCGIプログラム」[1]では、「ネットワークを利用する実用的なプログラム」としてCGIプログラムを紹介し、その教員側での実用例を示した。また、そのようなCGIプログラムを学生に書かせ、実用することで、複合的な教育効果が期待できることを述べた。すなわち「教育のための道具（ツール）」としてのCGIプログラムを論じたのである。

その際に、「学生が作ったCGIプログラム」を実用する可能性を示唆したが、そのような実用的プログラムを学生に書かせる方法や実践については触れていない。本論文では、「実用できるCGIプログラム」を含む「実用可能なWebサイト」の構築をより詳細に議論し、実践例を紹介する。

2. CGIプログラムと自作ライブラリ

一般的な講義・演習においても、プログラミング経験の浅い学生に、実用できるCGIプログラムを書かせることができる。そのために必要な「準備」を示したい。

使用言語はC++とした。本学ではプログラミング初心者にはCを教えており、また、オブジェクト指向を利用したいという理由からである。CGIプログラムの基本に関しては[1]を参照していただきたい。

ただし、本論文では、新たなライブラリを書いて使うことにした。CGIプログラムをより初歩的な形にしたかったからである（興味のある方は、お知らせいただければ、全コードをお送りします）。

著者によるライブラリを使用すると、以下のようなフレームで、CGIプログラムを作成することができる。

```

#include "myutil.h"
using namespace std;

int main()
{
    InputData id;
    id.receiveData();
    cout << "ㄖㄥ" << endl;
    . . . 処理 . . .
}

```

ここで、myutil.hは、ライブラリのヘッダーであり、以下のようなInputDataを定義している。ただし、必要な<iostream>や<string>などもインクルードしている。また、見やすくするため、「std::」は省略している。

```

class InputData
{
    map<string, string> data_;
    . . . ヘルパ関数 . . .
public:
    void receiveData();
    string getData(const string& s);
};

```

これは、ブラウザから送られてくるデータを、receiveDataで受け取り、マップであるdata_に格納し、必要に応じてgetDataでそのデータを取り出すオブジェクトのクラスである。

一般に、ブラウザからは、POSTによって、nameタグ付きの文字列データが送られてくる。そこで、nameと「対応する文字列データ」をペアにしてdata_に格納するようにし、getDataは、引数にnameを渡すことで「対応する文字列データ」を戻すようにしたのである。

CGIプログラムにおいて複雑なのは、データの受信、解析、日本語化、保持であるが、それらのすべてをInputDataが行う。そのため、InputDataを使えば、他の部分は、普通のコンソールアプリケーションのように記述することができる。

実際、先に示したフレームのmainの1行目は、InputDataオブジェクトの生成で、2行目はブラウザからのデータの取得である。そして、「. . . 処理 . . .」部分では、ブラウザから送られてきたnameに対応するデータをgetData(name)として扱うことができるのである。（注：InputDataオブジェクトは一度しかデータの受信をしないので、コンストラクタでデータを受信するように書くこともできる。すると、receiveDataは不要になる。しかし、それでは、プログラミング初心者には「データを受信している」という意識

がなくなるかもしれないと考え、receiveDataを使う仕様にした。)

たとえば、

```
<html>
<body>
シス基礎マイテスト2<br>
<form method="POST" action="../cgi-bin/mytest2.exe">
<input type="text" name="data1" size="20"><br>
<input type="text" name="data2" size="20"><br>
<input type="text" name="data3" size="20"><br>
<textarea name="data4" rows="4" cols="40"></textarea><br>
<input type="submit" value="決定">
</form>
</body>
</html>
```

というHTMLは、次のような画面になるが、

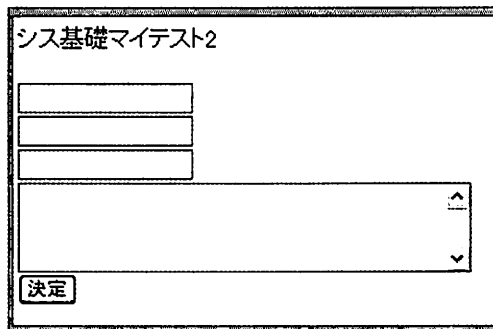
The image shows a web browser window displaying a form. The title bar of the window is not visible, but the page title is 'シス基礎マイテスト2'. The form consists of three single-line text input fields stacked vertically, followed by a larger multi-line text area with a vertical scrollbar on the right. At the bottom left of the form is a button labeled '決定' (Decide/Submit).

図 2 - 1 サイトの基本形

このHTMLに対して、「フレームの処理部分」を、

```
cout << id.getData("data1") << endl;
cout << id.getData("data2") << endl;
cout << id.getData("data3") << endl;
cout << id.getData("data4") << endl;
```

のように書くことができる。

このプログラムを実行する(図 2 - 1 の [決定] ボタンを押す) と、ユーザのブラウザに、自分が入力したデータが順番に表示されるわけである。

シス基礎マイテスト2

小林健一郎

テスト

送信

静岡産業大学情報学部

決定

小林健一郎
テスト
送信
静岡産業大学情報学部

図 2-2 プログラムの動作例

「フレームの処理部分」が、初歩的なプログラムである点に注意していただきたい。もちろん、テキスト入力だけでなく、ラジオボタン、チェックボックスも同様に扱うことができる。

3. 半期科目「システム設計基礎・演習」における実践

2008年度前期に、「システム設計基礎」、および「システム設計基礎演習」という科目（半期科目）の演習において、CGIプログラムの作成とWebサイトの構築を行わせた。「システム設計基礎」は、プログラミングの授業ではなく、システム設計を講義するものだが、その例としてCGIプログラムを取り上げたのである。

この科目は、当然、システム設計を学ぶことが中心であるが、ソフトウェア開発全般を視野に入れ、受注、要件定義、設計、プログラミング、運用・保守までを一通り見ていくものである。また、「プログラミング」も、プログラミングの技法を教えるのではなく、プログラミングの一般論や起こり得る問題点を中心に説明するものである。

「システム設計基礎」の演習である「システム設計基礎演習」のシラバスは次のようなものである。実際の講義も、概ね、シラバスに沿ったものであった。

- 第1週 ITの基礎知識
- 第2週 良いシステムとは何か
- 第3週 要求分析の実際
- 第4週 Apacheのインストールと稼働実験
- 第5週 簡単なWebサイトの構築
- 第6週 C/C++プログラミング
- 第7週 CGIを使ったWebサイトの概要設計
- 第8週 CGIを使ったWebサイトの詳細設計
- 第9週 CGIを使ったWebサイトの詳細設計
- 第10週 CGIを使ったWebサイトの開発
- 第11週 UMLによる記述
- 第12週 仕様変更への対処
- 第13週 CGIを使ったWebサイトのテスト
- 第14週 CGIを使ったWebサイトの運用と保守
- 第15週 最終課題

CGIプログラムの作成は、この演習の時間に行った。しかし、演習の目的は「講義で説明した一般論を具体的に確認すること」であるため、演習の時間をCGIプログラムの作成だけに使ったのではない。

実際、プログラミングの説明そのものに使ったのは、演習の第6週の全時間と、第7～10週までの一部の時間である。

CGIプログラムの作成に必要なネットワークに関する基礎知識は、第5週に説明した。HTMLの初歩も教えている。

受講者34名のほとんどが「プログラミング基礎」(極めて初等的なCの講義)を受けていたため、第6週では、コンパイラの使用方法などはほとんど説明せず、C++による入出力の方法とCGIの基本の説明に使うことができた。

サーバーとしてはApacheを使い、Webサイトは、それぞれの学生が持つノートパソコンに構築させた。ただし、学生はグループを組ませ、役割分担をさせた。その結果、プログラミングを担当する学生もあれば、担当しない学生もいた。実際にプログラミングを担当したのは、全体の3分の1程度の学生である。このような役割分担は、科目の性格上好ましいものと考ええる。

トータルで半年の時間をかけ、課題として実際に構築させたWebサイトは「講義のサポートサイト」である。サイトがサポートする「講義」は、システム設計基礎・演習ではなく、仮想の「一般的な情報教育」とした。教員である著者が「発注」し、学生が講義内で「受注」し、以後、要件定義から、設計へと進ませたのである。

この「講義のサポートサイト」の利用者も学生であるが、システム設計基礎・演習の学生は、「受注者」の役割を演じ、完成した「講義のサポートサイト」を利用するのは、別の講義を受講する学生(主に下級生を想定)とした。「自分たちが講義・演習で構築したサイトをその講義・演習で利用する」ということは、たいへん興味深い、その準備がで

きていなかったからである。ただし、グループごとに閲覧し、評価しあうようには指導した。実際にWebサイトとして稼働させるので、ネットワークを介して、お互いのサイトを見ることができたのである。

「講義のサポートサイト」の要件定義は、はじめから、明確な形で与えず、発注者である著者への聞き取りから実行させた。著者が「発注者」として要求したサイトの概要は、以下のようなものである。

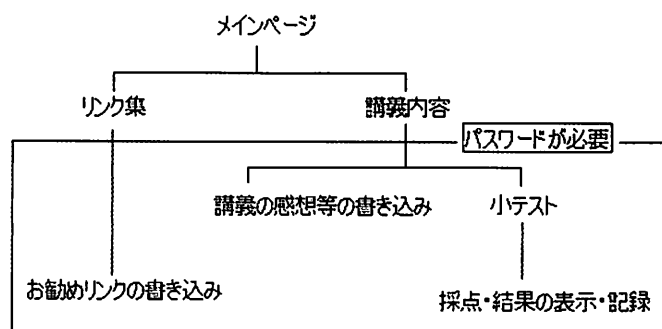


図3-1 期待される「講義のサポートサイト」の概要

上図の各項目は、構築したいサイトの主なページを表しているが、完全に対応するわけではない。また、ログインページなど、補助的なページは省略している。

「リンク集」では、「教員が勧めるサイトへのリンク」を表示し、「お勧めリンクの書き込み」では、学生が自分の「お勧めサイト」へのリンクを作れるように「要求」した。

「講義内容」は、仮想の講義の毎回の内容が表示されるものとした。その講義内容に対し、学生は感想等を書き込んだり（「講義の感想等の書き込み」）、また、小テストを受けられるようにした（「小テスト」）。小テストはすぐに採点され、その結果がブラウザ上に表示されるようにし、さらに、サーバー内にその結果が記録されるようにした（「採点・結果の表示・記録」）。

「パスワードが必要」と記した部分は、パスワードを入力しなければアクセスできないようにした。パスワードによるアクセス管理は、単純にクッキーをクライアントに保持させ、それを確認する方法を取った[3]。

「パスワードが必要」の部分にあるページは（ログインのページ含め）、動的なページであり、前節で説明したCGIプログラムを利用するように指導した。

以上の準備で、学生は、それぞれCGIプログラムを含むWebサイトを構築することができた。

4. 学生が構築したサイトの例

学生が構築したサイトの一部を紹介したい。本論文の表示用に、多少の手直しをしているが、本質的な部分は学生たちが作ったものである。

図4-1は、参考までに表示したが、動的なページではない。

5月28日 情報教育
<h2>講義内容</h2>
今回は、情報とはどういうものであるか 私たちににとってどのような役割を果たしているかを紹介します。
<h2>情報とは</h2>

<h2>情報の役割</h2>

図 4 - 1 講義概要のページ例

図 4 - 2 は、シンプルなログインページである。ここでログインすると、クッキーがクライアント側のパソコンに残されるようになっている。

<h2>ログイン</h2>
学籍番号 : <input type="text"/>
パスワード : <input type="password"/>
<input type="button" value="ログイン"/> <input type="button" value="戻る"/>

図 4 - 2 ログインのページ

図 4 - 3 は、各講義に対して、感想等を書き込むページの例である。

本日の授業であなたが感じたこと、あなたの考えを記述してください。	
学籍番号	<input type="text"/>
名前	<input type="text"/>
入力内容	<div><input type="text"/></div>
間違いがなければ送信ボタンを押してください。	
<input type="button" value="送信"/>	

図 4 - 3 感想等書き込みのページ例 1

感想等の書き込みのページとしては、図4-4のようなものもあった。6グループにサイトを作らせたので、それぞれ個性が出るのである。

感想

学籍番号:

名前:

決定

図4-4 感想等書き込みのページ例2

これらのページは、いずれの場合も、[送信] や [決定] のボタンを押すと、学籍番号、氏名、感想等が、サーバー側に保存されるようになっている。

図4-5は、テストのページの例である。このページでは、ラジオボタンを1つ選び、[決定] ボタンを押すと、テスト結果が表示され、また、サーバーにも保存されるようになっている。

第3回のテスト

次のうち、コンピュータの五大装置に入っていないものを選びなさい。

○制御装置 ○演算装置 ○出力装置 ○OS

決定

[TOP](#)

図4-5 テストのページ例

いずれもシンプルなものであるが、十分、実用的であると考える。

5. 結論と展望

「システム設計基礎・演習」で構築した「(仮想の) 講義のサポートサイト」は、十分実用的であると思われる。ライブラリを与えることで、プログラミング経験の浅い学生によっても、そのようなサイトが構築できてしまうということが、この論文の結論である。

また、ライブラリのコード自身も単純であり、決して、ブラックボックスではないことも述べておきたい。デコードなど、CGIにおける定型コードを含んでいるので、その部分を理解しようとするれば、ネットワークプログラムに関するある程度の深い知識が必要になる。しかし、それらのコードはヘルパ関数としてカプセル化しているので、興味に応じて読めばよい(それも、難しくはない)。したがって、興味を持ったプログラマ志望の学生が、ライブラリの内容を理解することは容易であるはずだ(ライブラリの利用などは[2]で論じている)。

「システム設計演習」は、登録者が34名であったが、6つのグループを作らせ、課題を行わせた。全グループがサイトの構築に成功したが、必ずしも、全グループのサイトが「良いもの」とは言えなかった。また、途中で講義を放棄した学生も数名いた。これらについては、今後の課題である。

授業アンケートで、「システム設計基礎・演習」の「総合して、この授業は自分にとって有意義である」の項目は、3.2(有効回答数27名、1点から4点で評価)であった。私が担当しているC言語の初歩を教える科目「プログラミング基礎・演習」のアンケート結果3.4(有効回答数29名)よりやや低いが、「情報学部 情報デザイン学科専門科目」の平均3.2と同じであり、学生の評価は決して悪くなかったと考える。

2008年度は、仮想の「講義のサポートページ」を対象にしただけである。将来的には、実際に「情報リテラシ教育」などを担当している教員と協力し、実用したいと考える。また、本学ではWebデザインに関する科目が多数開講されている。著者はその方面の専門家ではない。もし、Webデザイン系の講義と連動させることができれば、より実践的な教育になるのではないだろうか。

謝辞

著者は、CGIプログラムを文献[4]で学んだ。この本は、その後の改訂版も含めて絶版になっており、残念である。この本を私に紹介してくださったのは静岡産業大学情報学部の佐野典秀教授である。本研究は、半期科目「システム設計基礎・演習」を担当することで完成した。この科目を担当するにあたり、同学部の高橋恒介教授には、講義録を見せていただくなど、いろいろとご指導いただいた。佐野先生、高橋先生、また、いろいろとご教示いただいている静岡産業大学情報学部の先生方に感謝致します。また、本論分には「システム設計基礎・演習」の2008年度前期の受講生の作品を一部手直しして掲載している。これについては、あらかじめ、外部にも公開すると断っておいた。彼らにも感謝したい。

文献

- [1] 小林健一郎『教育におけるCGIプログラム』静岡産業大学情報学部紀要第8号
- [2] 小林健一郎『初心者を対象としたC++プログラミング教育について』静岡産業大学情報学部紀要第1号
- [3] W.W.Weinman『CGIブック』インプレス・プレントゥスホール
- [4] 笹木望、太田晶宏、藤崎 真美『ワンランク上のホームページのためのHTML&CGI入門』エーアイ出版

付録 ライブラリの概要

```
#ifndef _MYUTIL_H_
#define _MYUTIL_H_
//myutil.h
#include <string>
#include <map>
#include <sstream>

// データを受信するオブジェクトのクラス
class InputData
{
    std::map<std::string, std::string> data_; // データの保持
    char cxtoc(char c0, char c1);           // 受信データのデコード
    std::string jstring(std::string ustr);   // 文字列の日本語化
public:
    void receiveData();                      // データの受信
    std::string getData(const std::string& s); // 指定したデータを戻す
};

void setCookie(const std::string& s);        // クッキーのセット
std::string getCookie();                    // クッキーの取得

std::string dtos(int d);                    // 整数を文字列に変換
#endif
```