

## ネットワークを利用したインタラクティブなソフトウェア

榊原 正明・高見 和邦・堀内 敏史・増原 良子  
物質工学科

### An Interactive Computer Software Program using Network

Masaaki SAKAKIBARA, Kazukuni TAKAMI, Toshifumi HORIUCHI, Ryoko MASUHARA

Department of Materials Science, Faculty of Engineering  
Tottori University, Tottori, 680-8552 Japan  
E-mail: sakaki@lemon.bio.tottori-u.ac.jp

**Abstract:** The use of HTTP COOKIES makes it possible to distinguish different computer users. Using this system, an interactive computer program for qualitative analysis experiments involving chemical reagents was converted from stand alone use by individual computers to Network computer use. This computer program provides clients with guidance and a record of the questions, photographs and videos of the experiments.

**Keywords:** Educational computer program, Interactive, Cookie, Qualitative analysis experiments, Web

### 1. はじめに

全世界にインターネット網が張り巡らされ、あらゆる分野での利用が推進されている。インターネットにはスーパーコンピュータ、ワークステーションからパソコンに至るまで接続されており、共通の通信規則（プロトコル）である TCP/IP で情報をやり取りしている。現在、インターネット上で通信可能な主なものに「FTP」、「電子メール」、「Web」、などがある[1]。

Web のシステムは、1989 年に、研究者間での効率的な情報共有を目指して開発されたものである。その後、1993 年に Graphical な User Interface を持つブラウザ Mosaic が開発されると、一気に普及していった[1]。

Web には、次のような利点がある。①ハイパーテキストによってユーザ制御を可能にする。②視

聴覚、アニメーション等のマルチメディアによる提示が行なえる。③CGI スクリプト等による双方向性が可能である。④クロスプラットフォームである。⑤ファイルの更新と拡張が容易である。

これらの利点を生かして Web は、化学計算や様々な実験のデータベースとして、化学分野での利用の試みが進んでいる[2]。例えば、Web 上で分子の振動モードや反応の様子を動画で表示したり、ゲノムやタンパク質の巨大なデータベースを作成するといったものである。

また、教育分野においても、いろいろな活用が試みられている。矢野らは学内 LAN を利用した講義システムを試作した[3]。これは、Web システムを中核とし、電子掲示板および電子メールシステムを組み合わせたクライアント・サーバシステムである。この講義システムは自然科学学習のための補習教育用に開発され、教育内容をテキストや画像などのマルチメディア教材として有機的に組み

合わせて構成されている。しかし、答えた内容に対応してヒントを表示する、といったインタラクティブな機能は持っていない。

また、吉村らは、物理化学実験の事前学習のためにビデオ画像を導入し、CAL 教材を開発して個々のコンピュータで活用している[4]。特に、開発した化学ビデオ教材を HTML ファイルに変換して Web 上の化学 CAL 教材にしている。教材の内容は「中和熱の測定」、「電池の起電力」、「電位差滴定」、「分配率」の4つである。そして、それぞれの実験の内容(理論と目的、実験装置、実験方法、実験結果、考察)が書かれている。HTML ファイルの化学ビデオ教材をクライアントとして利用した場合は、サーバコンピュータからのビデオデータの取り込みに時間がかかっているため、ビデオの第一画面のカット・ペーストによる画像イメージで学習を進めている。

教育へのマルチメディアの利用は、まだ静的なホームページが多く、インタラクティブ(双方向性)なソフトは十分に開発されていない。今回は、Web 上でインタラクティブな操作を実現するために、スタンドアロンのインタラクティブな定性実験のソフト[5]を用いて、様々な方法を検討した。ここでインタラクティブとは、解答時の、ユーザの問題に対する進み具合を判断して、ユーザを識別してヒントを表示することを意味している。間違えた場合に、すぐに正解を表示するのではなく、ヒントを示すことにより、考える機会を与えることが出来る。

ユーザを識別するために、インターネットのオンラインショッピングに利用されている HTTP COOKIES の使用を試みた。また、ソフトの操作性を考えて、同一画面にヒントを表示するための方法の検討を行なった。さらに動画を含む画像を取り入れてマルチメディア利用の試みを行なった。この際、動画の表示などによって学習を中断するこ

とがないように、画像再生の時間短縮についても検討も行った。

## 2. ネットワークシステムの構築

CGI 等を使用するとサーバに対する負荷が大きくなり、また、共通に使用するサーバでは障害があった場合にそのサーバを使用している他のユーザに迷惑を及ぼすことがあるので、研究室内でサーバを立ち上げ、ネットワークシステムの構築を行なった。

今回構築した、システムのサーバは次のような仕様となっている。

サーバマシンには NEC PC9821V20 を使用し、OS には UNIX 系 OS で、パソコン用である FreeBSD(98) を使用し、Web のサーバプログラムに Apache httpd を使用した。

## 3. ユーザの識別

図 3-1 のように IP アドレスが固定されている場合、サーバ 0 はクライアント A とクライアント B を IP アドレスで識別することができる。しかし、同じパソコンを使っている、ユーザ A1 とユーザ A2 は、IP アドレスでは識別できない。

そこで、HTTP COOKIES (CGI を利用して cookie と呼ばれるデータを Web サーバからブラウザに送り保存するシステム)を利用した。これによって、ユーザの情報を複数回の接続に対して保持することが可能となり、ユーザの識別が可能となった。

なお、異なるヒントを出すだけなら HTML ファイルを使用しても可能である。しかし、これではユーザが選択した答えが分からないので成績を表示することが不可能である。

cookie は環境変数 HTTP\_COOKIE に格納されて HTTP ヘッダーに含めて送信される。cookie は「(

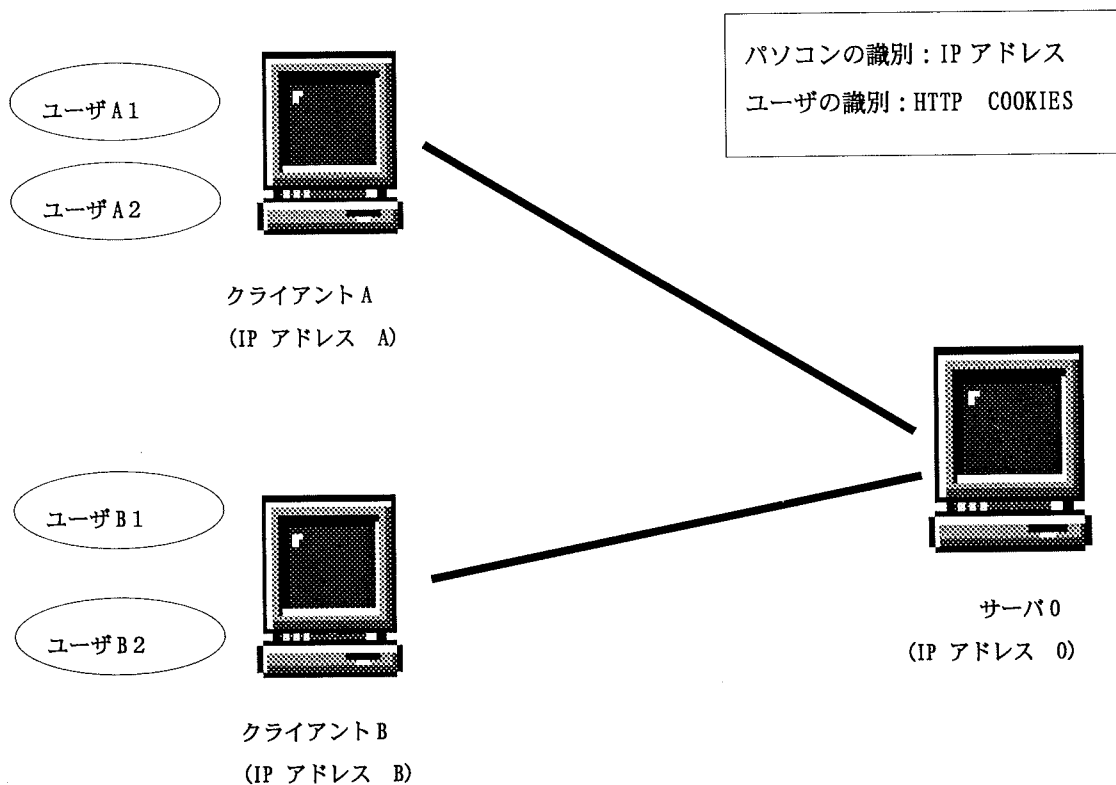


図 3-1 パソコンの識別とユーザの識別

キーワード)=(値)」の形式で受け渡しされる。

cookie の一つの大きさは 4Kbyte 以下に制限されており、その大きさを越えた分は切り捨てられる。他の制限としては、1つのブラウザに 300 個までしか登録出来ず、1つのドメインあたり 20 個となっている。この制限を越えると古い cookie から順に削除される。HTTP COOKIES は Netscape 2.0 以降、Internet Explorer 2.0 以降のブラウザに対応している。CGI プログラムとしては、今回 Perl 言語を使用した。

#### 4. 定性実験ソフトへの適用

このシステムの適用例として、インタラクティブな定性実験のソフト[5]をスタンドアロンのもの

から Web 上のものに作り替えた。

##### 4. 1 定性実験のソフトの説明

定性実験では 1, 2, 3 族のみを取り扱っている。1 族では鉛, 銀, 2 族では鉛, ビスマス, 銅, 3 族では鉄, アルミニウム, クロムの各陽イオンを扱っている。問題は化学実験書を基に作成し、使用する試薬を選択するものである。問題数は 1 族が 10 問, 2 族は 14 問, 3 族は 19 問である。

選択する試薬は、同じ族の問題では試薬の種類 (1 3 種) および表示の順序は同じであるが、試薬の種類は各族ごとに異なっている。

試薬を選択し、正解すると得点の一つ増えて次の問題へ移り、誤答の場合はヒントが表示され、

減点される。3回間違えると正解を表示して次の問題に移る。

同じ設問でも、間違えた回数によって異なるヒントが表示される。結果は、個人別のファイルを作成して、保存する。設問ファイルは HTML ファイルである。

## 4. 2 実行画面

最初の画面(図4-1)で学生番号を入力し、送信ボタンを押す。

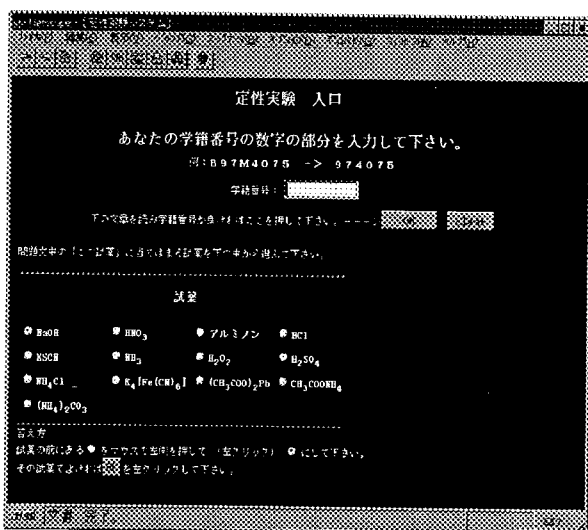


図4-1 学生番号の入力画面

設問の画面では、試薬の表示には Table タグを使用している。画面上の選択肢から答えを選び、ラジオボックスで試薬を選択する。答えは Submit ボタンで送信する。画像は IMG タグで表示し、画像をクリックすることによって、ビデオの画面になる。ビデオの画面では EMBED タグを使用している。

ヒントの表示された実行画面を図4-2に示す。全ての設問が終了すると、画面に成績を表示する(図4-3)。

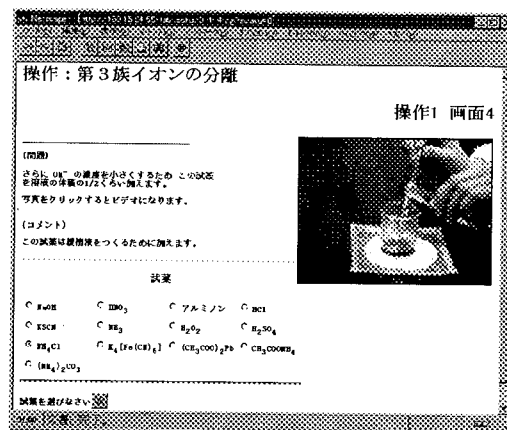


図4-2 ヒントの表示画面

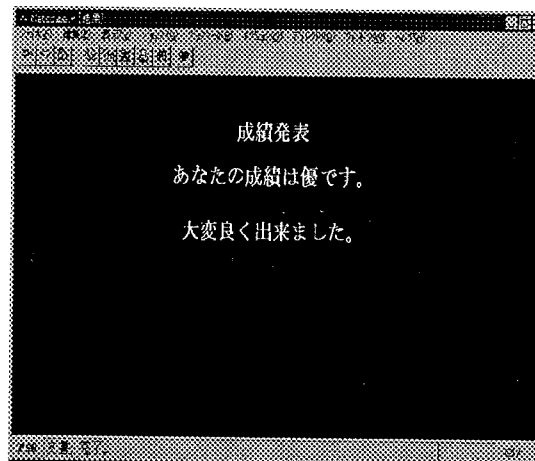


図4-3 成績表示の画面

## 5. CGI プログラムの説明

今回作成した CGI プログラムは、3つの部分「初期化の部分」「設問に答える部分」「成績を表示する部分」からなっている。

### 5. 1 初期化 CGI プログラム

初めの CGI は、入力された番号を cookie に保存

し、その番号をファイル名として使用したユーザファイルをサーバに作成する。これは cookie に入っている他の変数を初期化するためのプログラムである。

クライアントのブラウザとサーバ内の Web サーバと3つのファイル(CGI プログラム, HTML ファイル, データファイル)とのデータの流れは図 5-1 のようになる。①ユーザがブラウザからユーザ番号を入力する。②ブラウザは Web サーバに URL を指

定して初期化 CGI プログラムを要求し、データを送る。③Web サーバは URL で指定された CGI プログラムにデータを渡す。④初期化 CGI プログラムが実行されると、データをファイル名に利用したユーザファイルを作成する。⑤CGI プログラムは cookie を初期化し、初めの設問ファイルを読み出す。⑥CGI プログラムは cookie と実行結果を Web サーバに渡す。⑦Web サーバは cookie を含む受け取った結果をブラウザに送信する。⑧ブラウザは

① ユーザがユーザ番号を入力する

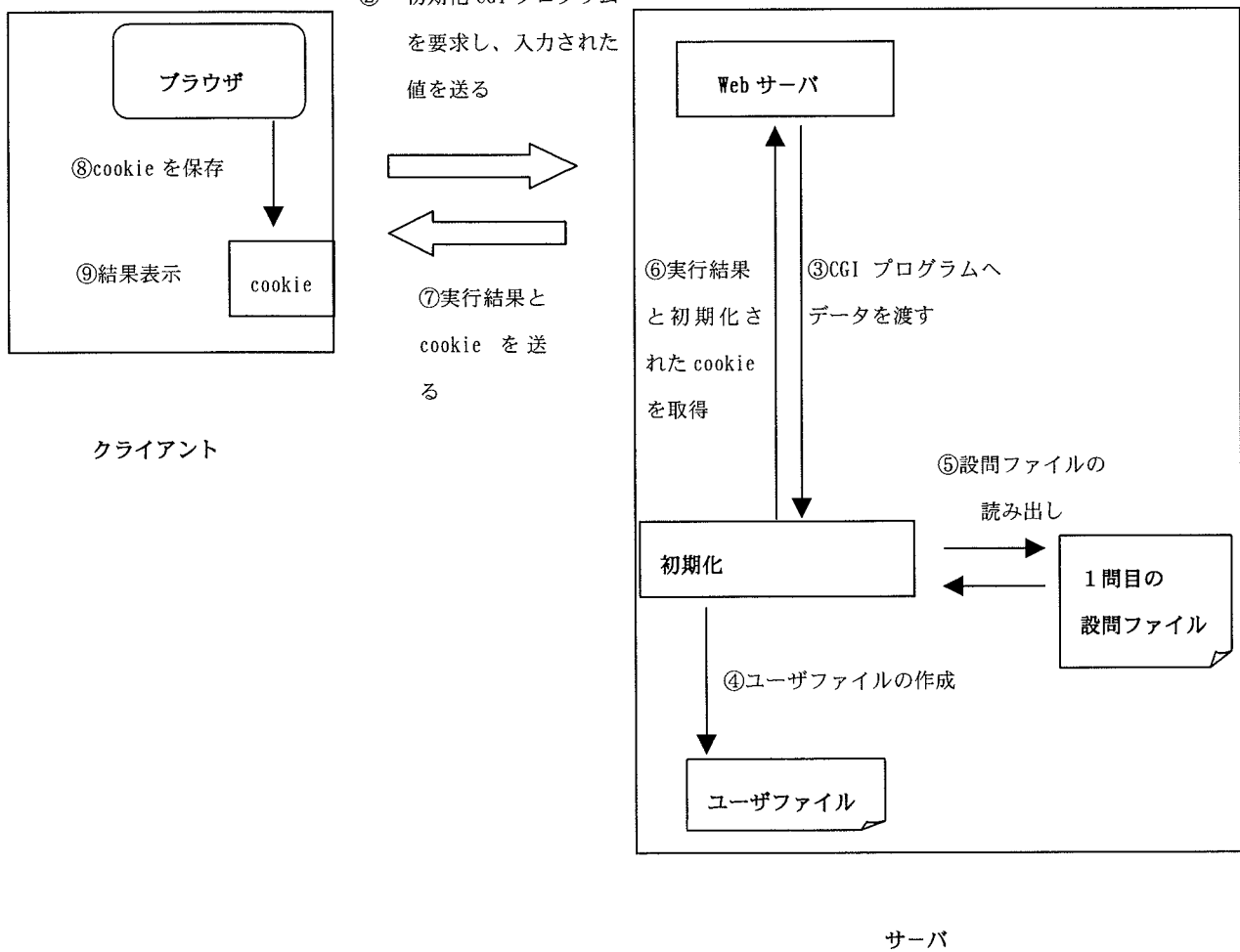


図 5-1 初期化 CGI プログラムにおけるデータの流れ

受け取った cookie を保存する。⑨ブラウザは実行結果を表示する。

初期化 CGI プログラムの流れ図は図 5-2 に示す。

初期化 CGI プログラムは以下の通りである。

```
#!/usr/bin/perl
require '/usr/local/www/cgi-bin/cgi-lib.pl';
&ReadParse;
# クッキーに付ける名前 (他と重ならないようにする)
$cookie_name = 'num';
# 記録するクッキーを設定
$cook = $in{"iid"};
# 時刻取得を国際標準時とする
$ENV{'TZ'} = "GMT";
# クッキーを消化する時刻を設定 (30*24*60*60 : 30日後)
($secg, $ming, $hourg, $mdayg, $mong, $year g, $wdayg, $ydayg, $i
dstg) = localtime(time + 1*60*60);
if ($yearg < 10) { $yearg = "0$yearg"; }
```

```
if ($secg < 10) { $secg = "0$secg"; }
if ($ming < 10) { $ming = "0$ming"; }
if ($hourg < 10) { $hourg = "0$hourg"; }
if ($mdayg < 10) { $mdayg = "0$mdayg"; }
$y0="Sunday"; $y1="Monday"; $y2="Tuesday";
$y3="Wednesday"; $y4="Thursday"; $y5="Friday";
$y6="Saturday";
$youbi = ($y0, $y1, $y2, $y3, $y4, $y5, $y6) [$wdayg];
$m0="Jan"; $m1="Feb"; $m2="Mar"; $m3="Apr"; $m4="May";
$m5="Jun";
$m6="Jul"; $m7="Aug"; $m8="Sep"; $m9="Oct"; $m10="Nov";
$m11="Dec";
$month = ($m0, $m1, $m2, $m3, $m4, $m5, $m6, $m7, $m8, $m9, $m10, $m11)
[$mong];
$date_gmt = "$youbi, $mdayg-$monthg-$yearg
$hourg:$ming:$secg GMT";
```

```
#ファイルの初期化
$f_result="./datafile/res$cook";
open(IN,">$f_result");
print IN " ";
close(IN);
```

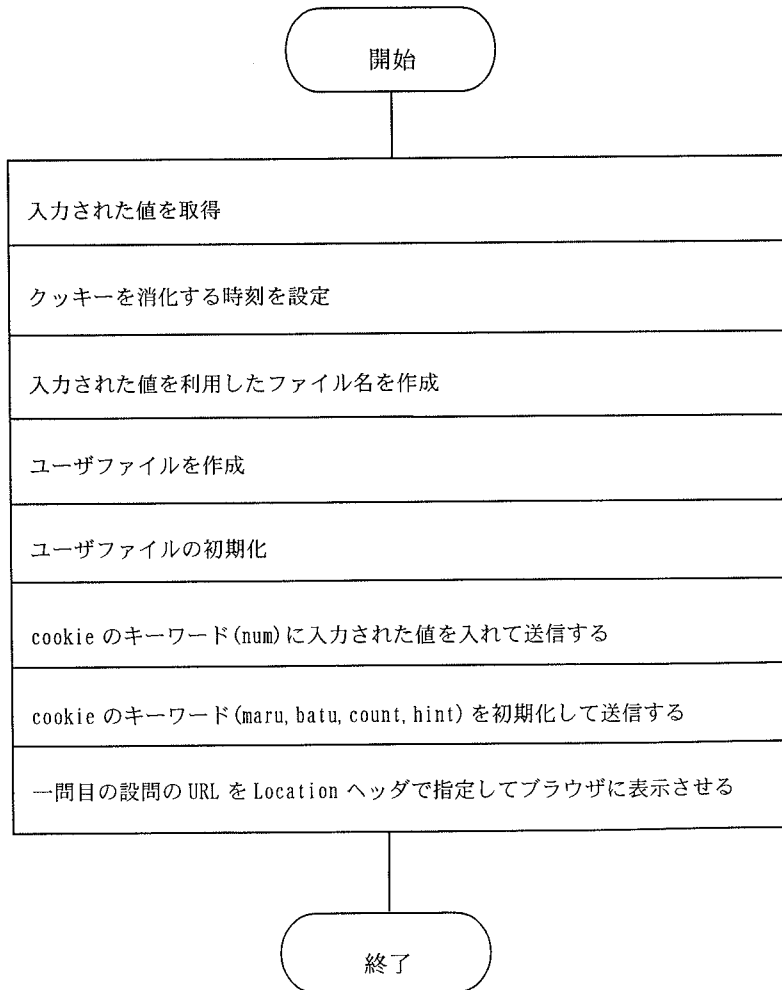


図 5-2 初期化 CGI プログラムの NS チャート

```
# ヘッダとして送信
print "Set-Cookie:          $cookie_name=$cook;
expires=$date_gmt¥n";
print "Set-Cookie: maru=0; expires=$date_gmt¥n";
print "Set-Cookie: batu=0; expires=$date_gmt¥n";
print "Set-Cookie: count=-1; expires=$date_gmt¥n";
print "Set-Cookie: hint=: expires=$date_gmt¥n";
print "Location: 1 問目の設問 CGI プログラムの URL ¥n¥n";
```

### 5. 2 設問の CGI プログラム

このプログラムは、ユーザが選択した答えをユーザファイルに保存し、その正誤を判断して設問を進めていくプログラムである。

図 5-3 にブラウザとサーバとのデータの流れを示す。データの流れは次のようになる。①ブラウ

ザで答える。②ブラウザは、URL を指定して CGI プログラムを要求し、選択した答えを送る。③Web サーバは URL で指定された CGI プログラムに受け取ったデータを渡す。④設問 CGI プログラムはユーザファイルに選択した答えを書き込む。⑤設問 CGI プログラムは設問ファイルを読み出す。⑥設問 CGI プログラムは実行結果と cookie を Web サーバへ渡す。⑦Web サーバはブラウザに実行結果と cookie を送信する。⑧ブラウザは受け取った cookie を保存する。⑨ブラウザは実行結果を表示する。

ここでの cookie にはヒント、ユーザ番号、同じ設問での誤答数(count)、そして、全体の正答数

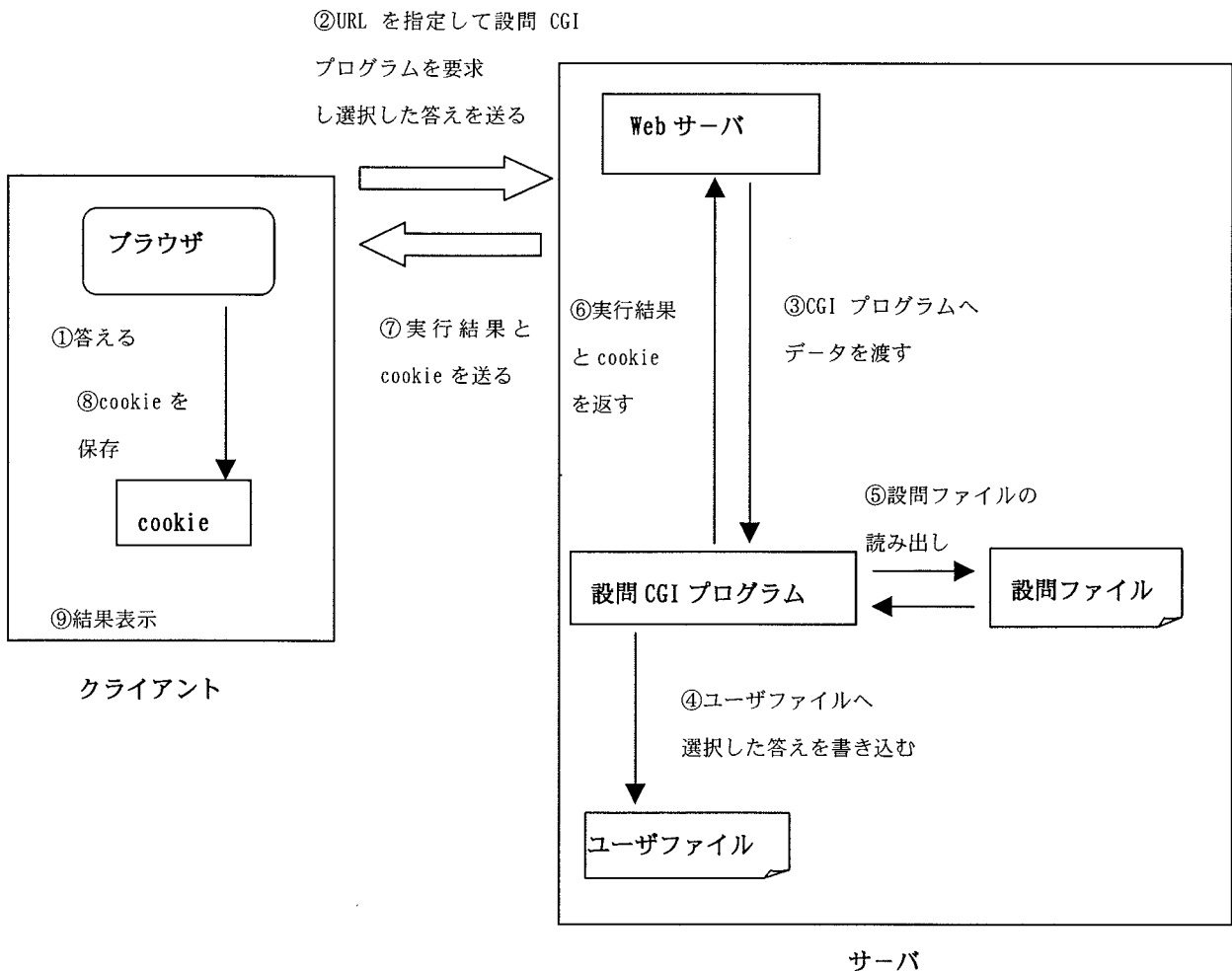


図 5-3 設問 CGI プログラムにおけるデータの流れ

(maru)と誤答数(batu)が含まれている。

CGI プログラムの流れ図を図 5-4 に示す。プログラムの流れは以下のものである。初めにデータが送られる。次にユーザファイルへの書き込みを行なう。その後、正誤の判断をする。間違えた場合は「ヒント」を、正解の場合は「正解です」を、3回間違えた場合は「答え」をそれぞれ cookie に入れる。次に、読み込むファイルを指定する。間違えた場合は同じ設問のファイルを、正解あるいは3回間違えた場合には次の設問のファイルを指定する。そして cookie を送信し、指定したファイルを読み込んで送信する。ヒントを表示する場所にはダミーの文字(\$hint)が書いてあるので、その文字を置換して送信することによってヒントを表示する。count が-1 の場合、ヒントの表示される部分を空白にして表示する。

CGI プログラムのソースを以下に示す。なお下線部の部分に具体的な番号、あるいは url が入る。

```
#!/usr/bin/perl
require "/usr/local/www/cgi-bin/cgi-lib.pl";
&ReadParse;
#cookie を環境変数から取り出す
foreach(split(/;/, $ENV{'HTTP_COOKIE'})) {
($key, $number)=split(/=/);
$cookieJar{$key}=$number;
}
if($cookieJar{count} == -1)
{
    $url = '同じ設問ファイル';
    $cookieJar{count}++;
    $cookieJar{hint} = "";
    &disphead;
}
else{
    #結果のファイル
    $f_result="/datafile/res$cookieJar{num}";
    open(IN, ">>$f_result");
    flock(IN, 2);
    print IN          "設問番号",
"$¥:", $cookieJar{count}, "¥:", $in{"siyaku"};
    print IN "\n";
    flock(IN, 8);
    close(IN);
    if($in{"siyaku"} eq "正解の試薬番号") {
        $cookieJar{maru}++;
        $cookieJar{count} = 0;
        $cookieJar{hint} = "正解です.";
        $url = '次の設問ファイル';
    }
    else{
        $cookieJar{batu}++;
```

```
        $cookieJar{count}++;
        if ($cookieJar{count} == 1) {
            $cookieJar{hint} = "1 回目
のヒント";
            $url = '同じ設問ファイル';
        }
        elsif ($cookieJar{count} == 2) {
            $cookieJar{hint} = "2 回
目のヒント";
            $url = '同じ設問ファイル';
        }
        else {
            $cookieJar{hint} = "正解
の試薬を表示";
            $url = '次の設問ファイル';
            $count = 0;
        }
    }
    &disphead;
}
sub disphead
{
    print "Set-Cookie: maru=$cookieJar{maru};
expires=$date_gmt\n";
    print "Set-Cookie: batu=$cookieJar{batu};
expires=$date_gmt\n";
    print "Set-Cookie: hint=$cookieJar{hint};
expires=$date_gmt\n";
    print "Set-Cookie: count=$cookieJar{count};
expires=$date_gmt\n";
    print "Content-type: text/html\n\n";
    open(fp, "$url");
    while (<<fp) {
        chop($_);
        if ($_ eq '$hint') {
            print "$cookieJar{hint}";
        }
        else{
            print ;
        }
    }
    close(fp);
}
```

「設問の解答」では設問毎に別の CGI プログラムを使用している。

設問ファイルと CGI プログラムの関係を図 5-5 に示す。

ブラウザから設問 1 を答えると CGI プログラムが実行される(破線矢印)。CGI プログラム 1 は、間違えた場合には設問 1 のファイルを読み込んで(点線矢印)表示し、正解あるいは3回間違えた場合には設問 2 のファイルを読み込んで(実線矢印)表示する。次に設問 2 に進む。ブラウザから答えると CGI プログラム 2 が実行される(破線矢印)。1 と同様に、間違えた場合は同じ設問(点線矢印)、それ以外は次の設問ファイルを読み込んで(実線矢印)表





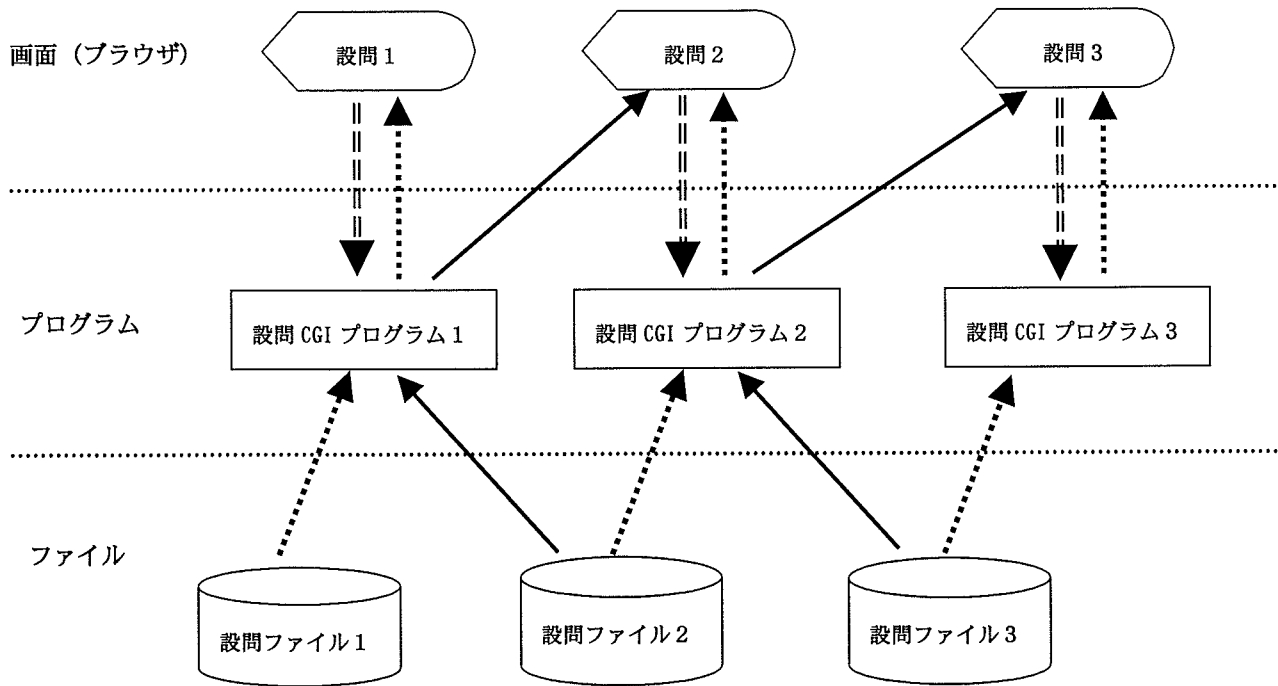


図 5-5 設問ファイルと設問 CGI プログラムの関係

示する。以後の設問も同様に進んでいく。

### 5.3 成績を表示する CGI プログラム

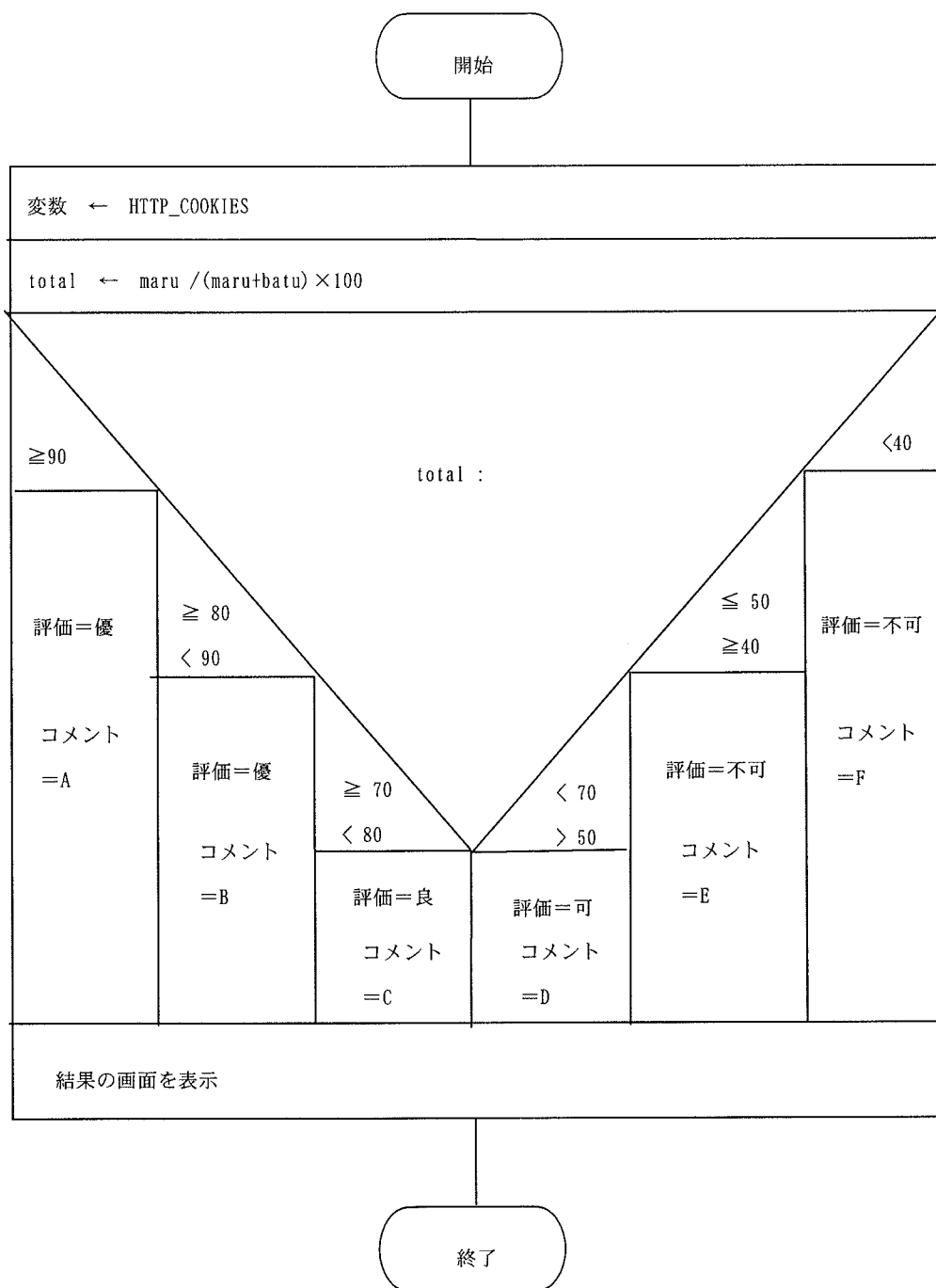
最後に「成績の表示」では、CGI プログラムは cookie に保存してある全体の正答数 (maru) と誤答数 (batu) により成績を画面に表示し、ユーザファイルに保存する。

成績を表示する CGI プログラムの流れ図は図 5-6 のようになる。

CGI プログラムのソースは以下に示す。

```
#!/usr/bin/perl
require '/usr/local/www/cgi-bin/cgi-lib.pl';
foreach(split(/;/, $ENV{'HTTP_COOKIE'})) {
    ($key, $value)=split(/=/);
    $cookieJar {$key}=$value;
}
$f_total = "./datafile/tot$cookieJar{num}";
$total =
$cookieJar{maru}/($cookieJar{maru}+$cookieJar{batu})*100;
open(IN, ">$f_total");
print IN $total;
```

```
close(IN);
#結果のファイル
$f_result="./datafile/res$cookieJar{num}";
open(IN, ">>$f_result");
flock(IN, 2);
print IN "maru", "¥:", $cookieJar{maru};
print IN "\n";
print IN "batu", "¥:", $cookieJar{batu};
print IN "\n";
print IN "total", "¥:", "$total";
print IN "\n";
flock(IN, 8);
close(IN);
if ($total >= 90) {
    $hyouji="優";
    $coment="大変良く出来ました。";
} elsif ($total >= 80) {
    $hyouji="優";
    $coment="良く出来ました。";
} elsif ($total >= 70) {
    $hyouji="良";
    $coment="もう少し注意深く実験しましょう。";
} elsif ($total >= 50) {
    $hyouji="可";
    $coment="もう一息がんばりましょう。";
} elsif ($total >= 40) {
    $hyouji="不可";
    $coment="かなりの努力が必要です。";
} else {
    $hyouji="不可";
    $coment="初めから勉強し直す必要があります。";
}
print "Content-type: text/html\n\n";
#文字化けする場合は、該当文字の直後に¥を書く (例 表¥示
```



コメント

A:大変よくできました。 B:良くできました。

C:もう少し注意深く実験をしましょう。D:もう一息がんばりましょう。

E:かなりの努力が必要です。F:初めから勉強をしないす必要がありま

図 5-6 成績 CGI プログラムの NS チャート

```
機能)
print "<HTML><HEAD><TITLE>結果</TITLE></HEAD>";
print "<BODY bgcolor=#000000 text=#fff fff>";
print "<BR><BR><BR><h1><P ALIGN=CENTER>成績発表
</P></h1>";
print "<H1><P ALIGN=CENTER>あなたの成績は<b>$hyouji</b>
です。 </P></H1>";
print "<H1><P ALIGN=CENTER><BR>$coment </P>";
print "<BR><A HREF=$初期化の画面のURL>><BR>";
print "最後にここを左クリックして下さい。 </A>";

print "</BODY></HTML>";
```

## 6. 画像について

画像には静止画と動画の両方を利用している。

静止画はデジタルカメラ (FUJIFILM DS-7) を使用して、JPEG 形式でパソコンに取り込んだ。これを画像ファイルのサイズを小さくするために、イメージファイルのコンバータによって GIF 形式に変換して使用した。

実験中の溶液などの変化の様子を示すために、動画の使用も試みた。動画を作成するために 8 ミリビデオ (SONY HandycomPRO CCD-V90) で映像を撮り、I-O DATA のキャプチャーツールを用いて AVI 形式でパソコンに取り込んだ。

ビデオの形式は 24bitRGB を使用、ビデオのサイズは 320×240 を使用した。最大キャプチャ時間は 10 秒に設定した。

しかし、AVI 形式は、サイズが大きく、ファイル全てを受信し終わるまで再生を行わないダウンロードタイプのため、ブラウザでの再生までの時間がかかりすぎる。そこで、圧縮率が高く、ビデオなどのファイルを、ユーザが受け取りながら再生するストリーミングビデオを用いた。今回は、ソフトの操作性を考えて、ブラウザのみで使用するため、外部ビューアが立ち上がらず、専用サーバが必要としない、Vivo Software 社が開発した VivoActive を使用した。これに伴い、AVI 形式から VivoActive の形式である VIV 形式に変換して使用した。この結果、動画再生までの時間を短縮す

ることができた。

しかし、静止画に比べてサイズが大きいため設定の画面には表示せず、別のウィンドウを開いて、そこで表示するようにした。

## 7. 実施結果

このソフトを農学部の化学実験受講生 27 名に実施した。結果として、実験のイメージがしやすかった等おおむね良好であった。

## 8. 終わりに

Web サーバを中心としたネットワークシステムを構築することにより、インタラクティブな機能を持ち、ユーザを識別したソフトの開発が可能となった。化学領域への試みとしては、実験操作の画面を簡単に示すことができ、動画を用いることによって実験の操作をイメージしやすくなった。なお、動画にストリーミングビデオを使用することによって再生までの時間を短縮することができた。

今後のプログラムの改良については、選択する試薬を操作ごとに変更する、試薬だけでなく沈殿の種類 (化学式)、色、反応等も問うようにする、

学生が使用しやすく、わかりやすい画面構成にするといった点が挙げられる。また、実際の画面をさらにきれいな映像として取り入れるようにすることも考えられる。

## 参考文献

- [1] Kris Jamsa Ph.D, Suleiman Lalani, Steve Weakley: Web プログラミング, 翔泳社.
- [2] 「化学」編集部編: 研究者のための Mac 読本, 化学同人.
- [3] 鈴木久雄, 伊澤俊二, 尾崎成子, 矢野敬幸: ネ

ットワーク新時代に対応した新しい講義システムの試み, *J. Chem. Software*, 3, 165, (1997).

[4] 吉村忠与志, 岩崎利史, 青山義弘: インターネット・ブラウザによる化学ビデオ教材の開発と試用, *化学教育ジャーナル (CEJ)*, 第1巻, 1号.

[5] 榭原正明, 高見和邦, 立花良一: 定性分析実験における学習プログラムの利用, *コンピュータ&エデュケーション*, 2巻, pp. 85-89, 1997.

(受理 平成 10 年 8 月 21 日)

