VirtualCampus ラーニング・システムにおける管理機構

Management Features of VirtualCampus Learning System

関 根 弘 之

要 約 ユニシス VirtualCampus は，インターネット/イントラネットならびに WWW ブラウザを利用した遠隔教育システムである．WWW サーバを教材ならびに学習者管理サーバとして利用する．学習者は WWW ブラウザを利用してオンデマンドな学習が可能となっている．学習を行うためにはあらかじめ学習可能な学習者を登録しておく必要がある．学習履歴はこの学習者単位に取得される．学習者の特定ならびに学習履歴の収集には CGI を使用したアプリケーション・プログラムを使用する．

同時に多数の学習者から利用要求があった場合には，サーバ上では対応する CGI アプリケーションが起動されることになり，サーバの負荷が大きい．

VirtualCampus の実装にあたってはサーバでの CGI アプリケーション数を削減することを目的に Cookie を利用した．近年，WWW サーバならびに WWW ブラウザ間でのデータ送受信を伴うアプリケーションでは Cookie を利用するケースが増えている．本稿では VirtualCampus の開発における Cookie の利用方法について紹介する．

Abstract UNISYS VirtualCampus is a remote learning system using the Internet and/or intranet including the WWW browser, and enables a trainee to learn by using a WWW browser on demand. In VirtualCampus system, the WWW server is used to distribute educational materials and control trainees. A person who want to learn must be registered as a trainee in VirtualCampus prior to the use of the learning system. The learning records is obtained for individual trainees.

Identification of a trainee and collection of learning records is performed by using the CGI（Common Gateway Interface）programs.

If the learning requests arise simultaneously from many trainees, the loads to the WWW server is increased by execution of the related handling programs.

Cookies have been employed in the implementation of VirtualCampus in order to decrease the number of CGI programs running on the WWW server. Many applications, which use Cookies to transfer information between WWW server and WWW browser, have been employed recently by degree. This paper introduses the application of Cookies employed in VirtualCampus.

1. はじめに

インターネット/イントラネット等のコンピュータネットワーク関連技術の進展に伴い，コンピュータ・ネットワークを利用した遠隔教育（学習）システムが注目を浴びている[1]．弊社でも，1997 年 12 月よりコンピュータ・ネットワークを利用した遠隔教育システムのユニシス VirtualCampus（以降，UVC）を開発，社内研修の手段として利用を開始した．また 1998 年 8 月にはこのシステムを商品化し，お客様にもご利用いただいている．経済環境の悪化等を背景とした企業内教育費用削減のための手段として，関心も高い[2]．

UVC はコンピュータ・ネットワークならびに WWW ブラウザを利用した学習環境を提供する．教材，学習者ならびに学習履歴は WWW サーバで管理し，学習者は WWW ブラウザを利用して学習を行う．WWW ブラウザは指定された URL に対応したサーバ上の HTML ファイルをオープンし，その内容をクライアントのブラウザに転送，内容を表示する．

通常この時点で，サーバとクライアント間のセッションが終了し，コネクションは切り離されることになる．学習者の認証や学習に関わる履歴情報をサーバにて蓄積・管理といった何らかの処理が必要となる場合には，CGI[*1] を利用したサーバ上のアプリケーション・プログラムを利用する．学習者へこのようなサービスを提供するためには，対応する CGI アプリケーションを起動しなければならない．学習者からの要求に対して，CGI はそれぞれ個別のプロセスとして実行されるため，サーバへの負荷が大きくなる．本稿ではサーバの負荷を軽減するための対応方法と，その実装について述べる．

2. 学 習 手 順

2.1 学 習 手 順

UVC を利用して学習を開始し教材の選択メニューが表示されるまでの流れを図 2 に示す．学習者は初期画面（図 1）からログインボタンをクリックし，ログイン画面であらかじめ登録されている学習者コードならびにパスワードを入力する．サーバでは学習者の認証を行うための CGI アプリケーションが起動され，学習者が登録済みかどうかの判定を行い，認証結果をブラウザに表示する．学習者の学習開始ボタンのクリックにより，学習様式ならびに教材選択画面が表示される．

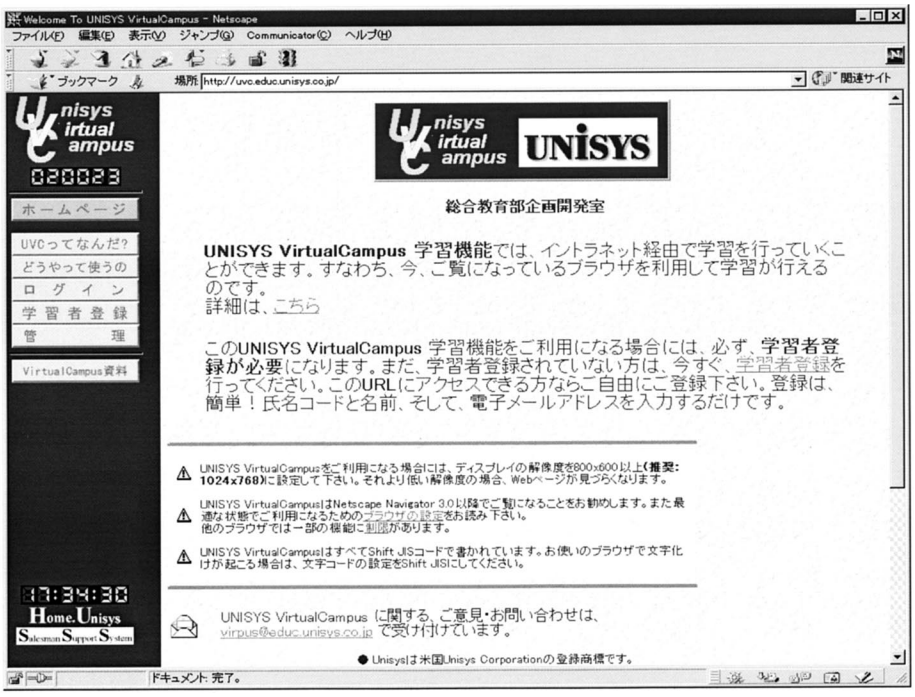

図 1　VirtualCampus の初期画面

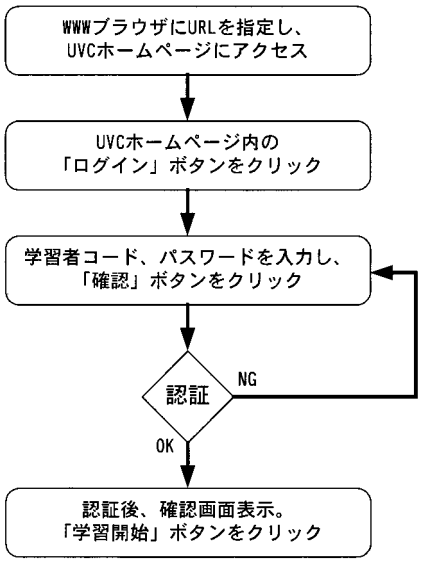

図 2　学習開始までの流れ

2.2 学習者ログ

学習を開始すると，サーバでは学習者単位に各種ログ情報を収集するために CGI アプリケーションが起動される．収集されるログ情報としては以下のものがある．

- ログイン日時
- 学習開始時間
- 選択された学習様式
- 選択された教材
- FAQ 参照日時
- その他

UVC の管理者向け機能によって，ログ情報を参照することができる．図 3 は，学習ログの表示画面である．このログ情報は，学習者の要求に応じてサーバ上の CGI アプリケーションによって収集される．

2.3 進捗管理

特定, あるいはすべての学習者の学習進捗状況を確認することも可能となっている．図 4 は，学習者の進捗状況の表示画面である．学習に利用されている教材で，学習済みの学習項目数と全学習項目数の比率で，進捗状況としている．

この情報は，学習者のログ情報を編集することによって表示している．

3. UVC におけるセッション管理

3.1 HTTP での CGI アプリケーションによるセッション管理

ブラウザを利用して何らかのデータをサーバへ送信，サーバによる処理ならびに処理結果をブラウザに返すまでのプロセスを，セッションと呼ぶことにする．

最も単純なセッションは，ブラウザから指定された URL に対応するサーバ上の HTML ファイルをオープンし，ブラウザに転送，その内容をブラウザが表示するも

教材ダウンロード	1998/06/30 16:45:26	172.17.16.130	[HMP-PDF]第1章 ネットワーク上でのビジネス展開
HTMLBASE学習開始	1998/06/30 16:46:28	172.17.16.130	Oracle7 技術留意点
教材ダウンロード	1998/06/30 16:47:10	172.17.16.130	Java入門(PDF文章)
教材ダウンロード	1998/06/30 16:55:10	172.17.16.130	Java入門(PDF文章)
教材ダウンロード	1998/06/30 16:55:38	172.17.16.130	Javaアプレット作成(PDF文章)
ログイン	1998/06/30 17:00:07	172.17.16.130	再ログイン
ログアウト	1998/06/30 17:00:39	172.17.16.130	学習終了
ログイン	1998/06/30 17:05:46	172.17.16.130	
教材ダウンロード	1998/06/30 17:07:23	172.17.16.130	[HMP-PDF]第1章 ネットワーク上でのビジネス展開
ログイン	1998/06/30 17:15:16	172.17.16.130	
HTMLBASE学習開始	1998/06/30 17:16:13	172.17.16.130	Oracle7 技術留意点
質問ボタンクリック質問ボタンクリック	1998/06/30 17:17:51	172.17.16.130	チャットによる質問を受付ない(Mail)
質問メール送信	1998/06/30 17:20:00	172.17.16.130	Subject: 今、名古屋です。
ログアウト	1998/06/30 17:20:25	172.17.16.130	学習終了
ログイン	1998/07/01 12:53:40	192.168.99.186	
教材ダウンロード	1998/07/01 12:54:27	192.168.99.186	[HMP-PDF]付録3 競合情報
ログイン	1998/07/01 12:57:13	192.168.99.186	
教材ダウンロード	1998/07/01 12:57:31	192.168.99.186	[HMP-PDF]第1章 ネットワーク上でのビジネス展開
ログイン	1998/07/08 10:31:43	192.168.99.186	
教材ダウンロード	1998/07/08 10:32:03	192.168.99.186	[HMP-PDF]第1章 ネットワーク上でのビジネス展開
ログアウト	1998/07/08 10:33:42	192.168.99.186	学習終了
ログイン	1998/07/08 10:33:54	192.168.99.186	
ログイン	1998/07/08 10:34:27	192.168.99.186	再ログイン

図 3　学習ログ表示画面（管理者向け機能）

345012	工藤 隆司	部署 不明 ryuji.kudo@unisys.co.jp	Excel97入門 Windows95入門	2 % 2 %
351018	栗城 章郎	部署 不明 Akio.Kuriki@unisys.co.jp	教育工学概論 Windows95入門 Excel95入門	50 % 18 % 5 %
357239	前田 篤	部署 不明 Atsushi.maeda@unisys.co.jp	viエディタ使用法	21 %
382036	松本昇三	部署 不明 shozo.matsumoto@unisys.co.jp	Access95入門 Windows95入門	4 % 4 %
382208	松本嘉臣	部署 不明 Yoshiomi.Matsumoto@unisys.co.jp	Word97入門	5 %
392402	松本 健作	部署 不明 電子メール 不明	UNIX入門 シェル・プログラミング viエディタ使用法	学習済 学習済 16 %
392945	松山美樹	部署 不明 Miki.Matsuyama@unisys.co.jp	Excel97応用 Excel97入門 Access97入門	学習済 学習済 学習済
401118	三宅 晴夫	部署 不明 HARUO.MIYAKE@UNISYS.CO.JP	Windows95入門 Excel97入門	学習済 39 %
417131	宮本 重克	部署 不明 Shigekatsu.Miyamoto@unisys.co.jp	Excel95入門 Excel97入門	52 % 2 %
421233	宮脇亨	部署 不明 電子メール 不明	Access97入門 UNIX入門	47 % 学習済
			Excel97応用	29 %

図 4　学習進捗状況表示画面

ので，サーバの負荷は少ない．

しかし，学習者のブラウザから入力された学習者コードならびにパスワードを利用し，該当する学習者が登録されているか否かの判定を行おうとすると，サーバ上で学習者を管理しているデータベースを検索し，検索結果を HTML 形式のデータに変換し，学習者のブラウザに送信するという処理が必要となる．一般には，サーバ上で CGI を利用したアプリケーションプログラムを作成することになる．学習開始に関わるセッションの概要を図 5 に示す．ここで注意すべきことは，各セッションごとに，必ず学習者コードならびにパスワードを指定してセッションを開始しなければならないということである．

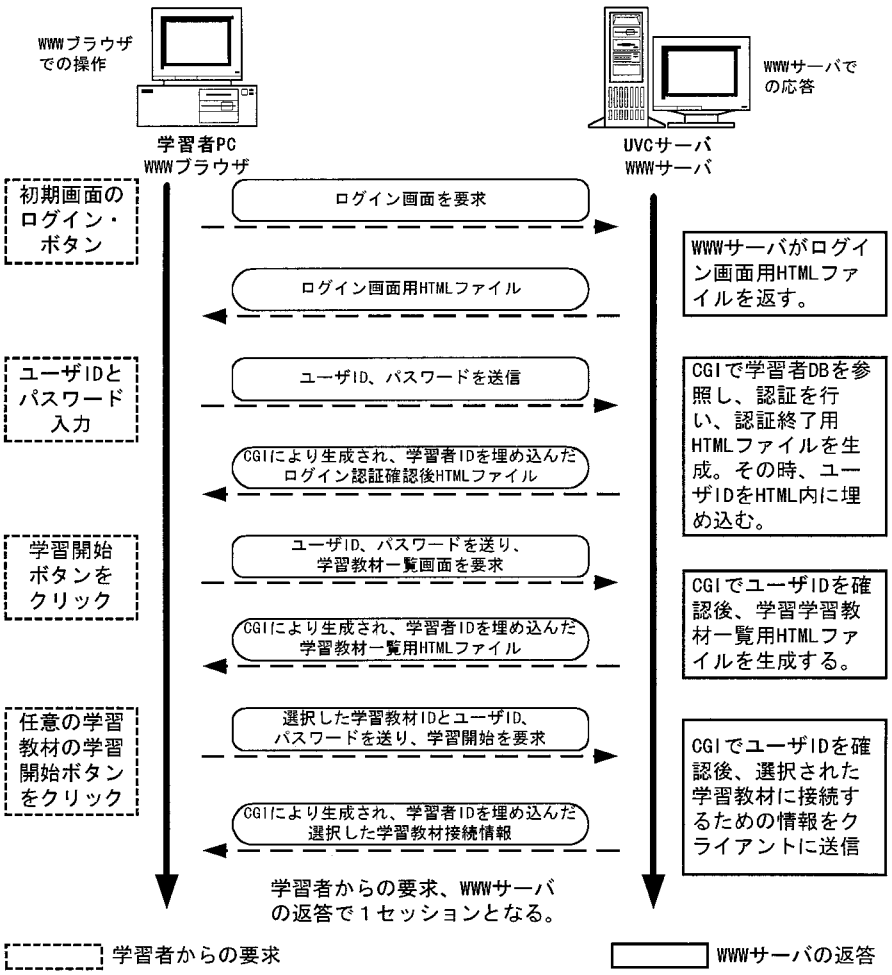

図 5　学習者のログイン後の教材選択画面表示までの CGI アプリケーションの働き

学習者の要求に応答するためには，学習者コードならびにパスワードを常時やり取りして，特定の学習者の要求に対応した処理を行なうのが一つの方式である．この方式を使用する限り，学習者の認証を必要としない場合でも，CGI アプリケーションを利用して，ブラウザの表示画面に対応する HTML データを作成しなければならない．

もしも，このような CGI アプリケーションを利用しない場合は，その時点で学習者コードとパスワードの情報を失うことになる．この方式では，学習者が任意のページを参照しようとするたびに，サーバ上で CGI アプリケーションが起動されるため，サーバに対する負荷も増大し，学習者へのレスポンスが遅くなる．また HTML の変更や，表示画面のレイアウト変更等が発生した場合には，CGI アプリケーションで設定されている HTML 出力部分の更新が必要となり，その都度 HTML の再作成（結果的に CGI アプリケーションの再コンパイル）が必要となる．結果として，システムの保守性も低下する．

3.2　Cookie を利用したセッション管理の仕組み

3.2.1　Cookie とは

CGI アプリケーションを利用してセッション情報を管理する方式では，学習者の要求の都度，学習者の認証をサーバの CGI アプリケーションで行わなければならず，サーバに対する負荷が大きい．UVC の実装にあたっては，WWW ブラウザと WWW サーバ間で擬似的にセッションを継続し，サーバで学習者の認証を行うための CGI アプリケーションの利用を省略するための方式として，Cookie と呼ばれる機構を採用した．Cookie は，Netscape Communications 社が，WWW ブラウザならびに WWW サーバ間での通信を簡略化するために開発した機構であるが，今では業界標準となっており，他のブラウザでも利用することができる．

Cookie は最初のセッションにおいてサーバの CGI 等で生成，WWW ブラウザに送られ，その後のセッションで WWW ブラウザから WWW サーバに送信される情報である．サーバで生成された Cookie は，クライアントのハードディスクに自動的に保存される．Cookie をクライアントがサーバに送ることにより，セッションごとにコネクションが切断するプロトコルである HTTP を利用しても，擬似的なコネクションを継続することが可能となり，サーバの負荷を軽減することができる．Cookie の書式を表 1 に示す．

3.2.2　Cookie の機能

セッションにおいて Cookie を利用する場合，まずクライアントのハードディスクに作成されている Cookie ファイルの内容を検査する．ブラウザから指定された URL をアクセスする場合，指定された URL とファイル内の Cookie が持つドメイン名，およびパスの情報を比較する．ドメイン名は後方一致でチェックするため，Cookie に「domain＝xxx.yyy.zz.jp」とあった場合，アクセス先のサーバ名が「www.xxx.yyy.zz.jp」でも一致していると見なす．パスは前方一致でチェックする．パスを「/aaa」と指定してあれば「/aaa 1」や「/aaa/bbb」とする URL も条件を満足しているとみなす．また，サーバからクライアントに Cookie が送られた時，すでに Cookie を管理するファイル内に，同一名称ならびに同一パス名の Cookie が存在すれば Cookie は上書きされる．

Cookie の生成はサーバ側で行い，クライアントのブラウザに転送されるが，この時 Cookie の生成ならびにブラウザへ送信するためのプログラムが必要なる．通常 CGI や JavaScript や Java 等を利用する．CGI を利用する場合には，HTTP ヘッダー部分に「Set－Cookie：」に続けて Cookie 情報を付加するようにプログラムを作成すれば

表 1 Cookie の仕様

レスポンス・ヘッダー（WWW サーバ から WWW ブラウザへ送信される情報）	
Cookie の属性	Set-Cookie: *NAME=VALUE;* expire=*DATE*; path=*PATH*; domain=*DOMAIN_NAME*; secure
NAME=VALUE	Cookie の名称（NAME）とその値（VALUE）を指定する．必ず指定しなければならない．
expires=*DATE*	Cookie の有効期限．「Sunday,22-MAR-1988 09:30:45 GMT」のように指定する．何も指定していない場合、そのセッションが終了すると消去される．
domain=*DOMAIN_NAME*	Cookie を転送してきたドメイン名．サーバにアクセスするときこのドメイン名と指定された URL のドメインが一致すればそのサーバに Cookie を送信する．
path=*PATH*	Cookie を転送してきたサーバでのパス名．ドメイン名やパスを指定しない場合には Cookie の送信元が設定される．
Secure	これが指定してある時は、SSL のようなセキュリティが確保されているチャネル上でのみ転送される．
リクエスト・ヘッダー（WWW ブラウザから WWW サーバへ送信される情報）	
Cookie の属性	Cookie: NAME1=VALUE1; NAME2=VALUE2;… ここで NAMEn はサーバから送られた Cookie の名称を示す．VALUEn は任意の文字列で Cookie の名称に対応する値となる．

よい．以下は，実際に CGI で作成した Cookie 情報をクライアントに送るサンプルである．

HTTP のヘッダー情報として WWW ブラウザに送る例（C 言語）

```
printf("Content-type: text/html¥n");
printf("Set-Cookie: SESSION=UVC12345; Expires= Tue, 10-Jun-1969 00:00:00 GMT; path=/; ¥n¥n" );
printf("<HTML>¥n");
printf("<HEAD>¥n");
        :
```

HTML の HTTP-EQUIV を利用して WWW ブラウザに送る例（C 言語）

```
printf("Content-type: text/html¥n¥n");
printf("<HTML>¥n");
printf("<HEAD>¥n");
printf("   <META HTTP-EQUIV=¥"Set-Cookie¥" CONTENT=¥" SESSION=UVC12345; Expires= Tue, 10-Jun-
1969 00:00:00 GMT; path=/¥"¥n");
printf("   <TITLE>Welcome to VirtualCampus</TITLE>¥n");
printf("<HEAD>¥n");
        :
```

WWW サーバは，WWW ブラウザに返事を返すときに，応答の HTTP のヘッダーに Cookie の名前と値等の文字列を埋め込んで返送される．逆にブラウザからサーバへ送られる Cookie は，CGI 等から環境変数「HTTP_COOKIE」で参照できる．付録 1 に，C 言語を利用した Cookie の取得する関数のサンプルを紹介している．この関数は，引数に取得する Cookie 情報の名前を指定することにより，その値を戻り値として返す．

JavaScript や Java を利用した場合は，サーバ側で用意したプログラムを転送し，そのプログラムをブラウザ上で動作させて Cookie を操作することもできる．

JavaScript などを使うメリットは，クライアント側だけで Cookie 情報を利用した

操作ができることである．ダウンロードしたHTMLファイル内に，JavaScriptのプログラムを埋め込むことによって，クライントでJavaScriptを処理し，Cookieを操作することができる．これを利用するとWWWブラウザのフォームに入力したデータ等を，WWWサーバを経由せずにCookie情報として設定することも可能となる．付録2に，JavaScriptを利用したCookie情報の取得方法と設定方法の例を紹介している．UVCではCGI及びJavaScriptを利用してCookieを操作している．

3.3 UVCにおけるセッション管理

UVCではCookieに学習者情報を登録することによって，要求が出されたWWWブラウザ（学習者）を特定する．すでに認証が行われた学習者であれば，再度，認証を行わずに処理を進めることができる．図6にCookieによるセッション管理の概要を示す．

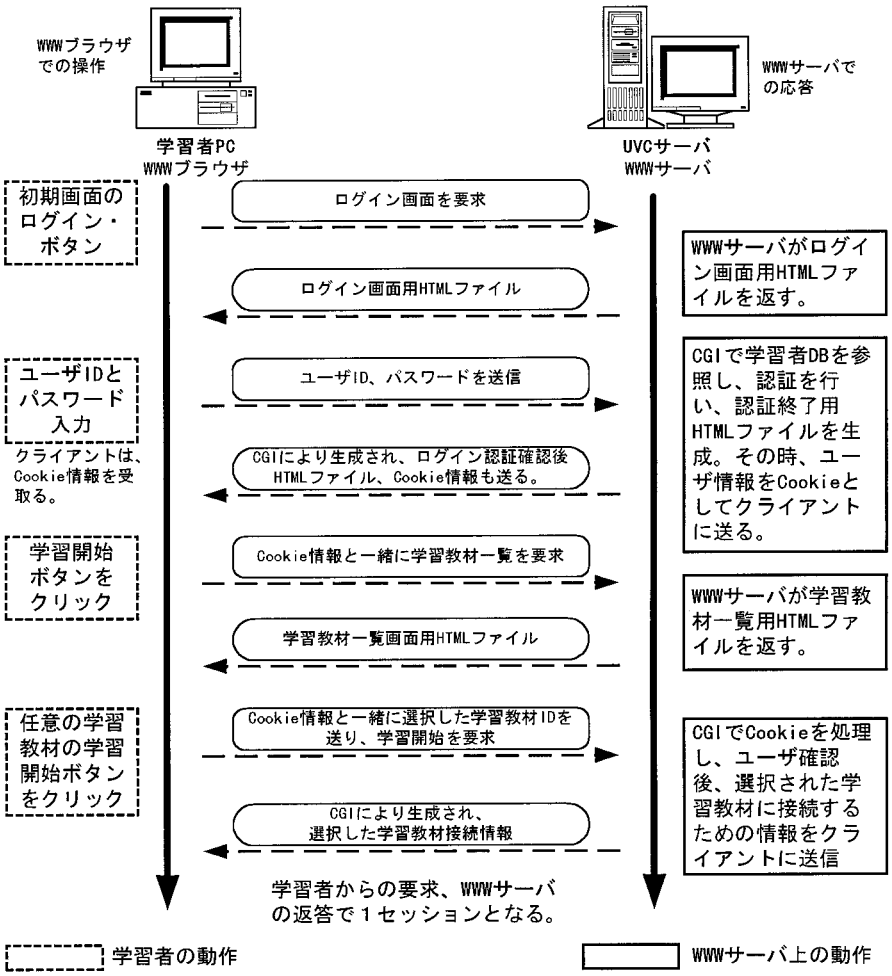

図 6　Cookieを利用したWWWサーバとWWWブラウザ間のセッションの仕組み

学習者がログインを要求すると，UVCサーバ上でCGIアプリケーションにより学習者固有のCookieを生成し，ブラウザに送信する．Cookieの生成以外にもセッショ

ン管理のために以下のような各種処理を行う．

① UVC がインストールされているディレクトリ内（wwwroot\UVC）に，セッションを管理するためのディレクトリを学習者毎に作成する．このディレクトリ名は，5 桁の数値を乱数により取得し，UVC という文字列に連結した 8 文字の文字列（例：UVC 12345）となる．

② 作成されたディレクトリに学習に必要な HTML ファイルをコピーする．

③ 作成されたディレクトリ名，完全パス名，学習者コードといったセッション情報ををログとして記録する．

④ CGI アプリケーションが生成する HTML 内に，作成されたセッション情報を Cookie に組込み，学習者の WWW ブラウザに送信する．

この処理で作成されるディレクトリは，ログインを実行した学習者専用となり，学習に必要なファイルなどが置かれる．また，IIS[*2] の仮想ディレクトリ機能により，UVC サーバの URL に関連付けられる．例えば，UVC サーバ上で作成されたセッション情報管理のためのディレクトリが「C:\UVC\wwwroot\UVC\UVC 12345」となった場合，学習者は WWW ブラウザから「HTTP://uvc.educ.unisys.co.jp/UVC/UVC 12345/」という URL を指定して接続することになる．

表 2　Cookie のデータ

Name	SESSION
Value	8 文字のセッション名
Domain	UVC サーバのドメイン名
Expire	CGI 起動時刻＋24 時間をグリニッジ標準時形式で指定
Path	/ (/でない時はその Path を指定)

④の処理で生成される Cookie のデータを表 2 に示す．Cookie の有効時間（expires の値）は 24 時間としている．これにより，ログイン後，24 時間はセッション毎にクライントからサーバへ Cookie を送信するので，これを利用して学習者の特定を行うことができる．また，学習終了時に行うログアウトにより，WWW ブラウザに送り込んだ Cookie を削除する．

3.4　Cookie 利用における運用上の工夫

セッション情報は，ログインする学習者毎にセッション情報を管理するログに保存される．このログに追加された情報は，ログアウトにより削除される．しかしながら WWW ブラウザの場合，ハイパーリンクをクリックすることで，簡単に他のサイトに移動することができるため，学習終了時に必ずログアウト作業が行われるという保証はない．ログアウトせずに終了した場合には，セッション情報はログにそのままに残る．このため，学習許可が与えられていない学習者が Cookie を不正操作し，ログインせずに学習を開始される可能性が生じる．UVC では，この問題を WindowsNT のスケジュール機能の利用で解決している．

Windows NT のスケジュール機能と AT コマンド[*3] を組み合わせて，一定時間毎（1 日 4 回 6 時間毎）にセッション情報をログから削除するプログラムを起動し，作

成されてから 24 時間（Cookie の有効時間）アクセスされていないセッション情報をログから削除する．ログアウトせずに学習を終了した場合には，セッション情報がログに 24 時間保存されることになる．24 時間以内に再度ログイン作業を実行するとセッション管理のログを参照し，ログへの追加時間を再ログインの時間に更新する．

学習者がログインを行う時には，UVC のホームページにあるログインボタンをクリックするという操作を行う．通常，この操作を行うと学習者コードとパスワードを入力するフィールドが表示される．しかし，Cookie によって格納されているセッションの有効時間内で，かつ，UVC サーバ上のセッションと対応したセッション情報がログに残っている場合には，ログイン画面は表示せずに，直接学習開始画面が表示される（図 7）．

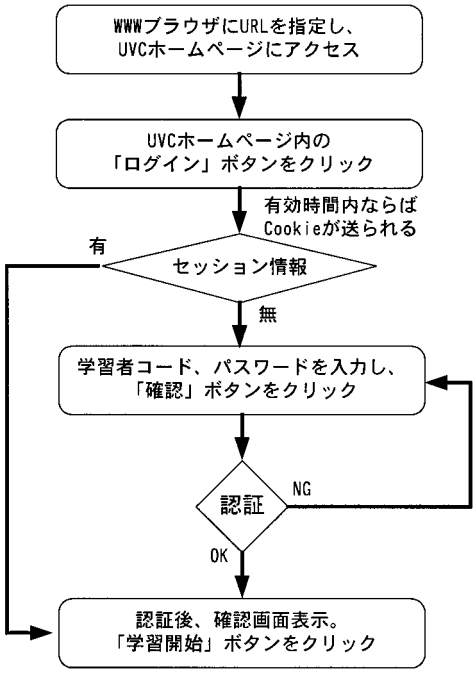

図 7　学習開始までの流れ

この機能では，ログインボタンがクリックされた時点で UVC サーバ上の CGI アプリケーションが起動され，Cookie がクライアントから送られてきているか，セッション情報がログに存在しているかを確認し，ログイン画面の表示もしくは再接続かを判断している．これにより，Cookie が有効時間内にある等の条件があえば，学習者は再ログインのために学習者コードとパスワードを入力する作業を省略することができる．

4. おわりに

UVC システムでは，Cookie を利用してセッションを制御することにより，必要となる CGI アプリケーション数を減らし，WWW サーバの負荷を軽減することを実現

した．また学習で利用する各種表示画面に対応するHTMLファイルを固定的に作成することができ，保守性も向上させることができた．Cookieを利用することによるデメリットもある．

- Cookie情報を必要としない場合でも，サーバに対してCookieが送信される．

  指定された有効時間の間，条件が一致すれば必ずクライアントはサーバに対してCookie情報を送る．サーバ側のCGI等でCookie情報が必要でない場合にも，ネットワークのトラフィックをあげてしまうことになる．しかし，Cookieは一つのドメインに対して20個しか持つことができないので，深刻なネットワークの輻輳を作り出すことは考えにくい．

- セキュリティの問題が有るといわれている．

  ブラウザを特定できるため，アクセス属性をトレースするなどすると，プライベート上の問題につがる可能性がある．Cookieを利用する開発者が故意に行わない限り，このようなことは起こり得ないことではある．

- Cookieデータは，クライアントのHDDに保存される為，簡単に参照できる．

  Cookieデータは，ブラウザにより決められた場所に，テキストファイルとして保存される．もしも，パスワードなどがCookieに保存されていた場合，そのファイルを参照することでパスワードを盗むことが可能になる．また，データを書き換えてしまえば不正にアクセスすることも可能となる．

現在，IETF（Internet Engineering Task Force）で標準化が進んでいる，Cookieの次期版「HTTP State Management Mechanism」は，利用方法やフォーマットは現行のCookieとほとんど同じである．しかし，ポート番号を指定したり，Cookieの有効時間を秒単位で指定できるなど，セッション管理に有効な機能が追加されている．

上記にあげたCookieのデメリットや，CGIの利用によるサーバ負荷の問題を解決するために，CGIアプリケーションと同等の機能を，Javaのservletを用いて実現することを検討している．servletは，WWWサーバで利用しているプロセスのスレッドとして動作する．通常のCGIアプリケーションは，リクエストそれぞれに対してひとつのプロセスとして実行されるが，servletの場合，一つのプロセス中で複数のスレッドを発生させて実行するので，サーバに与える負荷は少ない．結果として，WWWサーバ上での使用メモリを押さえ，かつ，高速に動作させることが期待できる．現在UVCはWindows NTのInternet Information Serverで利用することができる．Java技術を使用することによって，複数のプラットフォームでの利用という副次的効果も期待できる．

本稿で紹介した管理機能は，UVCシステムの一部の機能であるが，利用者のニーズに対応した機能を追加し，学習者にとってはより学習しやすく，管理者にとってはより管理しやすいシステムにしていきたいと考えている．

＊1　Common Gateway Interface．サーバ上のプログラム．
＊2　Microsoft Internet Information Serverの略．WWWサーバ機能などを提供する．
＊3　指定した日時にコマンドとプログラムがコンピュータで実行されるようにスケジュールするプログラム

付録 1　C 言語を利用した Cookie データの取得関数

```
char *getCookie( char *CookieName ) {
    char *cookie, *cp;   /* 環境変数 HTTP_COOKIE の値を格納 */
    char *name;          /* COOKIE の名前    */
    char *pos;           /* 文字列操作用      */
    int   len;           /* COOKIE 名の長さ */
    int   i = 0;         /* 文字列操作用添字 */

    /* COOKIE データのフォーマット */
    /* name1=val1; name2=val2; name3=val3 */

    if ( !CookieName ) return NULL; /* 指定された CookieName が NULL ならば、NULL を返す. */

    len = strlen( CookieName ) + 1;   /* 取得する Cookie の名前の長さ +1 を得る */

    if ( !(name=(char *)malloc(len + 1)) ) return NULL; /* メモリの確保に失敗したら、NULL を返す. */
    strcpy( name, CookieName );   /* Cookie 名検査用に"="を追加 */
    strcat( name, "=" );

    /* 環境変数"HTTP_COOKIE"から Cookie 文字列を取得 */
    if ( !(cp = cookie = getenv("HTTP_COOKIE")) ) {
        free( name );
        return NULL;
    }

    /* 指定された Cookie 名に対応する値を取得する */
    while ( *cookie != '¥0' ) { /* COOKIE のデータを一文字ずつ比較 */
        if ( !_strnicmp(cookie, name, len) ) { /* Cookie 文字列に検索文字列 name があるかをチェック */
            /* 見つかった文字列のデータ部分にポインタを移動 */
            for ( i = 0; i < len; i++ ) cookie++;
            if ( !(pos = strchr(cookie, ';')) ) { /* データの区切りの';'を検索 */
                if ( !(pos = strchr(cookie, '¥0')) ) { /* データの区切りの'¥0'を検索 */
                    free( cp ); free( name );
                    return NULL;   /* 見つからなければ、NULL を返す.  */
                }
            }
            len = pos - cookie;   /* データの文字数を取得 */
            if ( !(pos = (char *)malloc(len + 1)) ) {
                free( cp ); free( name );
                return NULL;
            }
            strncpy( pos, cookie, len );
            free( cp ); free( name );
            return pos;
        }
        cookie++;
    }
    free( cp ); free( name );

    return NULL;
}
```

付録 2　JavaScript を利用した Cookie 用関数

```
// 引数で与えられた Cookie の情報を返す
// 値がなければ、null を返す
function GetCookie( name ) {
    var cName       = name+'='; // チェック用 Cookie 名
    var cNameLen    = cName.length; // チェック用 Cookie 名の文字数
    var myCookie    = document.cookie; // 全 Cookie データ
    var startPos    = myCookie.indexOf( cName ); // データのスタート位置
    var endPos;

    if ( document.cookie.length == 0 ) return null; // 全 Cookie 文字列数が 0 なら null を返す
    if ( startPos != -1 ) { // 指定された Cookie データが見つかったら、
        startPos = startPos + cNameLen; // データのスタート位置を決める
        endPos = myCookie.indexOf( ';', startPos ); // データの終了位置を決める
        if ( endPos == -1 ) endPos = document.cookie.length; // データの終了位置を決める
        return unescape( myCookie.substring(startPos, endPos) ); // 値を返す
    }
    return null;
}

// 引数で与えられた Cookie の情報を設定する
function setCookie( name, value ) {
    var expire = new Date();
    // 有効期限を 1 ヶ月（31 日）とする
    expire.setTime(expire.getTime() + (31*24*60*60*1000));
    // 決められた書式でデータを設定する
    document.cookie = name + '=' + escape(value) + '; expires=' + expire.toGMTString() + '; path=/';
}
```

参考文献

［1］ 玉置亮太, 教育支援システムの新動向, 日経コンピュータ, no. 450, pp. 104〜106, 1998.

［2］ 日本ユニシス株式会社総合教育部, 教育いんふぉ, VC 導入事例, 1999 年 3 月号〜6 月号.

［3］ Simon St. Laurent, Cookie 入門, アスキー出版, 1998.

［4］ 小松原健, 末安泰三, Cookie を活用する, 日経インターネットテクノロジー, No. 9, 4 月号, 1998, pp. 50〜73.

［5］ Shishir Gundavaram, CGI プログラミング, 株式会社オライリー・ジャパン, 1996, pp. 222〜224.

［6］ 大津真, JavaScript パーフェクトリファレンス, 株式会社 BNN, 1997, pp. 198〜208.

執筆者紹介　関　根　弘　之（Hiroshi Sekine）

1969 年生．1993 年神奈川大学工学部卒業．同年日本ユニシス(株)入社．客先システムエンジニアの教育・カリキュラム作成に従事．1996 年以降，CAI システム，Virtual-Campus 等の開発・保守を行い，今日に至る．現在，総合教育部企画開発室に所属．