

授業中の学習行動把握のための授業支援システムの試作

2009SE053 平林 有理 2009SE139 熊井 啓人

指導教員：後藤 邦夫

1 はじめに

近年、情報通信ネットワーク及びPCの普及が著しいものとなっている。それにともない大学でも情報通信技術を利用した授業が増えている。そのような技術を利用することで、Webに講義資料をアップロードすることができ、さらにWebで小テストを行う事もできる。そうして情報通信技術を利用している大学は授業をより効率的にしている。しかしその一方、授業のために用意されたネットワーク環境を使って学生が授業とは関係のないウェブサイトを開覧するといった問題が増えている。

この問題を解決するために、本研究では授業中の学生に対しての参加強制、そして学生の行動把握を可能とするシステムを試作した。システムはプロキシとSQLite3データベース(以下データベースと記述する)とアクセス解析プログラムで構成される。TAPは学生と教員のWebアクセスのさい、データベースのリストを参照し、学生への振る舞いをきめ、同時にリストの内容を更新する。TAPで作成したログをアクセス解析プログラムで解析することにより学生の行動を把握することを可能にする。

本研究と目的がおなじ先行研究[4]が存在する。先行研究ではWeb-basedプロキシを作成し、授業中のアクセス制限を実現していた。本研究では、より高度な授業中のアクセス制限に加えて、アクセスログを使用したアクセス解析を行い学生の行動把握を支援する。

擬似授業を行うことでシステムが正常に動くかの確認と行動把握が行えるかどうかの評価を行う。授業の内容に取り組んでいる学生を想定した被験者とそうでない学生を想定した被験者に学生として利用してもらい、アクセス解析の結果そのちがいを判断することができれば行動把握ができていますと評価する。

第2節では先行研究との比較、第3節ではシステムの概要、第4節ではプログラムの構成、第5節ではシステムの実験、第6節では今後の課題、をそれぞれ記述する。共同でシステムの設計をおこなった。平林有理は主にプログラム作成を担当した。熊井啓人は主に文書作成を担当した。

2 先行研究との比較

この節では本研究と先行研究の違いを示す。

2.1 プロキシシステムの違い

先行研究ではCommon Gateway Interface(CGI)プログラムとWebサーバソフトウェアを組み合わせることでWeb-basedプロキシシステムを作成していた。そしてそれを学生とWebサーバの間に紹介させていた。

一方本研究では擬似ではないプロキシ(TAP)を作成して、それを学生とWebサーバの間に紹介させる。

2.2 本研究のプロキシシステムの利点

先行研究ではホスト名を使用して学生の判別を行っていたため、MACアドレスとホスト名の紐付けをしているDHCPサーバおよびDNSサーバを使用環境において用意しているという前提が必要であった。それに加えて、PCのホスト名が学生を区別できるものであるという前提も必要である。本研究ではTAPを使用する前に使用者にプロキシのユーザ認証のためのユーザ名とパスワードを設定させることで、その前提が必要ないものとしてできる。また、その2つの前提を満たしている場合、PCのホスト名で学生を判別することができ、出席には学生本人のPCが必要となる。それにより、学生が他の学生のPCを使用して代返するというような不正行為を防ぐことができる。

2.3 先行研究との機能比較

本研究と先行研究の機能比較を表1で行う。

表2で×とされている部分は先行研究の利用しているプロキシシステムの方式でも技術的には可能である。

表1 機能比較表

	本研究	先行研究
アクセス記録		
アクセス制限		
アクセス解析		×
Javascript 実行		×
FTP と HTTPS プロトコルへの対応		×

- アクセス記録
どの学生がいつWebページにアクセスしたかをデータベースに記録する。
- アクセス制限
学生のWebページアクセスの制限。教員がアクセスしたページにしかアクセスを出来なくする。先行研究ではCGIプログラムを使った擬似プロキシのため、CGIへのパスを含めたURLのみでアクセス制限が可能であった。本研究では、プロキシを使用している環境では許可されていないURLへのアクセスをすべて制限することができ、そこが先行研究よりも優れた点であると言える。
- アクセス解析
アクセス履歴を解析する。解析した結果をHTMLファイルに出力する。
- Javascript 実行
先行研究ではJavascriptが実行出来なかったが本研究では実行可能。先行研究ではWebサーバのJavascriptへのパスを書き換えていなかったために実行が不可能であった。

- FTP と HTTPS プロトコルへの対応
先行研究では FTP と HTTPS プロトコルに対応していなかったが、本研究では対応している。

3 システムの概要

TAP とアクセス解析の概要を示す。

3.1 システムの前提

1. ブラウザの設定

学生に今回作成した TAP を使用するようにブラウザの設定を変更させる。

2. ユーザ認証

プロキシのユーザ認証を行うためにユーザ名とパスワードを学生に設定させる。図 3 の CGI ページを用いて学生に設定させる。

3.2 システムの構成

システムの物理的な構成を図 3 に示す。

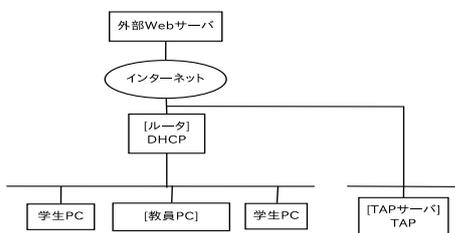


図 1 システムの構成図

教員と学生は Web サーバに HTTP 要求を行うさい、教員 PC の中にある TAP を経由して HTTP 要求を行う。TAP はそのたびに許可リストとアクセス記録を更新する。

3.3 システムの流れ

システムの流れを図 2 に示す。SQLite3 を利用し、学生のアクセスを許可する URL をデータベース化したものをアクセス許可リストとする。これは教員がアクセスしたページの URL を保存するデータベースである。

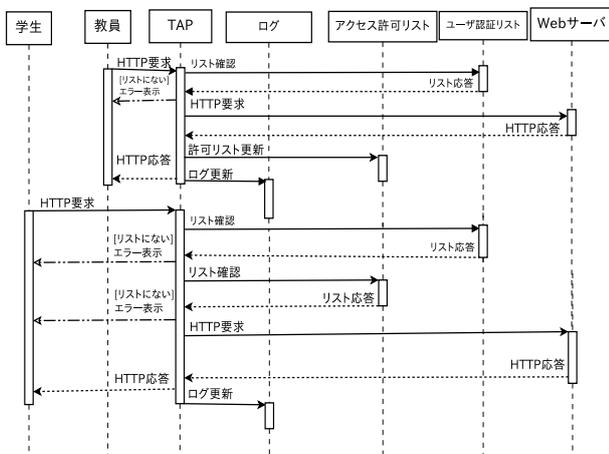


図 2 シーケンス図

教員は Web サイトを閲覧することで、その Web サイトの URL を許可リストに追加する。次にその流れを示す。

1. 教員が Web サイトを閲覧しようとする、そのユーザがユーザ認証リストに存在するか判定する。存在しない場合エラーページを教員に送信する。存在する場合 2 に進む。
2. TAP に HTTP リクエストが送られる。
3. TAP が Web サイトに HTTP リクエストを送る。
4. Web サイトから TAP に HTTP レスポンスが送られる。
5. TAP から教員の PC に HTTP レスポンスが送られる。
6. TAP は Web サイトの URL をアクセス許可リストに追加する。
7. TAP は教員からの HTTP 要求に対する結果をログに記録する。

学生が Web サイトを閲覧するさいの流れを示す。

1. 学生が Web サイトを閲覧しようとする、そのユーザがユーザ認証リストに存在するか判定する。存在しない場合エラーページを学生に送信する。存在する場合 2 に進む。
2. 学生が Web サイトに HTTP リクエストを送る。
3. TAP はその Web サイトの URL がアクセス許可リストに存在するか判定する。
4. 存在しない場合エラーメッセージを表示する。存在する場合その Web サイトに HTTP リクエストを送り 5 へ進む。
5. TAP は学生が Web サイトに HTTP 要求を送ったことをアクセス履歴リストに記録する。
6. TAP は学生からの HTTP 要求に対する結果をログに記録する。

3.4 アクセス解析プログラム

プロキシで作成したテキストログを解析して結果を HTML ファイルに出力し、学生の行動把握に使用する。

教員が新しくページにアクセスしたさい、学生がどれほどの時間でそのページへアクセスしたかを解析する。

教員が授業評価に使用するための HTML ファイル（以下教員用 HTML ファイル）と学生や教員が授業の状況が一目でわかるような HTML ファイル（以下開示用 HTML ファイル）を作成する。

• アクセス解析の方法

アクセス解析の視点は教員が新しく開いたページに一定時間内にすくなくとも一回アクセスしたかどうかの回数、そして教員が新しく開いたページへのアクセス時間の平均である。

• 教員用 HTML ファイル

学生ごとの IP アドレス、ホスト名、ユーザーエージェント、教員が新しく開いたページに一定時間内にすくなくとも一回アクセスしたかどうかの回数、

教員が新しく開いたページへのアクセス時間の平均を表示する。

- 開示用 HTML ファイル
学生全体の教員が新しく開いたページへのアクセス時間の平均を表示する。

4 TAP のプログラムとアクセス解析プログラムの概要

本研究では Python を用いて、TAP 及びアクセス解析プログラムを作成する。これは本研究におけるプログラムを作成するにあたり有用なモジュールが Python に多く用意されていたため [3][5] である。

TAP は Suzuki Hisao 氏が開発した Tiny HTTP Proxy in Python[2] を元に有志が改良を加えたもの [1] に手を加え Proxy 認証機能、アクセス制限機能を追加したものである。

5 TAP の負荷実験

システムの動作を確認するために実験を行う。

apache2-utils に含まれるベンチマークソフトウェア ab を使用し簡易的な負荷実験を行う。

ab コマンド

```
ab [ -X proxy[:port] ] [ -c concurrency ]  
[ -n requests ]  
[http[s]://]host-name[:port]/path
```

実験の結果、同時接続 100 で総数 1000 のリクエストを送ったとき、システムは負荷に耐えることができた。このことから、100 程度の規模の授業での使用ならば実用に耐えると思われる。しかし、同時接続 150 で総数 1500 のリクエストを送ったときは、システムは負荷に耐えることができなかった。原因は Python のスレッド数上限に達してしまったためではないかと思われる。

6 TAP とアクセス解析プログラムの実験と評価

3 人の学生を対象とした擬似授業を行うことで、システムの動作を評価する。

6.1 アクセス制限の動作検証

アクセス許可リスト作成と学生の Web サイト閲覧制限機能の動作が正確に動作するか確認する。その手順を次にしめす。

1. アクセス許可リストを確認する。
図 3 のようにアクセス許可リストの内容を確認する。
2. 学生側から TAP を通してアクセス許可リストにない Web サイトにアクセスする。

学生側から <http://h303s0.sd.nanzan-u.ac.jp/> にアクセスする。

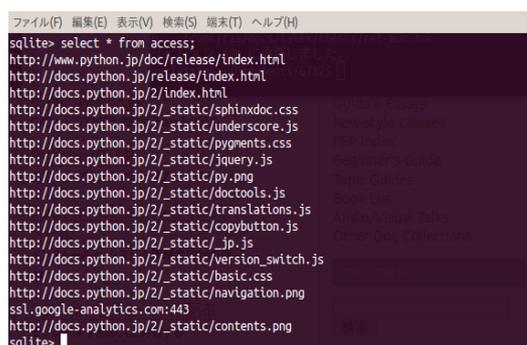
図 4 のようにエラーページが表示される。このことからアクセスしようとしたページの URL がアクセス許可リストに存在しないときに、学生側からその

URL のページへアクセスしてエラーページを返されたことが確認できる。

3. 教員側から 2 でアクセスしたページにアクセスし、許可リストを確認する。

教員側から <http://h303s0.sd.nanzan-u.ac.jp/> にアクセスする。アクセス許可リストを確認し、図 5 のようにそのページの URL がアクセス許可リストに追加されていることを確認する。

4. 学生側から 2 のアクセスページにアクセスする。
学生側から再度 <http://h303s0.sd.nanzan-u.ac.jp/> にアクセスする。アクセス先のページを正しく表示されることを確認できたため本システムが正確に動作していると判断できる。

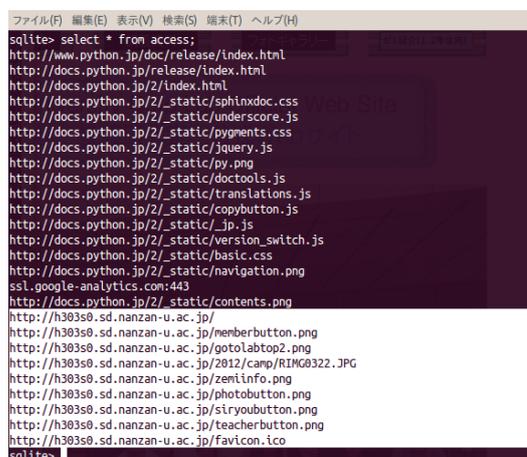


```
sqlite> select * from access;  
http://www.python.jp/doc/release/index.html  
http://docs.python.jp/release/index.html  
http://docs.python.jp/2/index.html  
http://docs.python.jp/2/_static/sphinxdoc.css  
http://docs.python.jp/2/_static/underscore.js  
http://docs.python.jp/2/_static/pygments.css  
http://docs.python.jp/2/_static/jquery.js  
http://docs.python.jp/2/_static/py.png  
http://docs.python.jp/2/_static/doctools.js  
http://docs.python.jp/2/_static/translations.js  
http://docs.python.jp/2/_static/copybutton.js  
http://docs.python.jp/2/_static/_jp.js  
http://docs.python.jp/2/_static/version_switch.js  
http://docs.python.jp/2/_static/basic.css  
http://docs.python.jp/2/_static/navigation.png  
ssl.google-analytics.com:443  
http://docs.python.jp/2/_static/contents.png  
sqlite>
```

図 3 実験前のアクセス許可リスト確認



図 4 エラーページ表示



```
sqlite> select * from access;  
http://www.python.jp/doc/release/index.html  
http://docs.python.jp/release/index.html  
http://docs.python.jp/2/index.html  
http://docs.python.jp/2/_static/sphinxdoc.css  
http://docs.python.jp/2/_static/underscore.js  
http://docs.python.jp/2/_static/pygments.css  
http://docs.python.jp/2/_static/jquery.js  
http://docs.python.jp/2/_static/py.png  
http://docs.python.jp/2/_static/doctools.js  
http://docs.python.jp/2/_static/translations.js  
http://docs.python.jp/2/_static/copybutton.js  
http://docs.python.jp/2/_static/_jp.js  
http://docs.python.jp/2/_static/version_switch.js  
http://docs.python.jp/2/_static/basic.css  
http://docs.python.jp/2/_static/navigation.png  
ssl.google-analytics.com:443  
http://docs.python.jp/2/_static/contents.png  
http://h303s0.sd.nanzan-u.ac.jp/  
http://h303s0.sd.nanzan-u.ac.jp/memberbutton.png  
http://h303s0.sd.nanzan-u.ac.jp/gotolabtop2.png  
http://h303s0.sd.nanzan-u.ac.jp/2012/camp/R1W0322.JPG  
http://h303s0.sd.nanzan-u.ac.jp/zeminfo.png  
http://h303s0.sd.nanzan-u.ac.jp/photobutton.png  
http://h303s0.sd.nanzan-u.ac.jp/sryoubutton.png  
http://h303s0.sd.nanzan-u.ac.jp/teacherbutton.png  
http://h303s0.sd.nanzan-u.ac.jp/favicon.ico  
sqlite>
```

図 5 アクセス許可リスト追加確認

6.2 アクセス解析プログラムの使用結果と評価

TAP とアクセス解析プログラムを動作させアクセス解析の結果を表示する。実際の授業と同じ形式で3人の学生に対して擬似授業を行いログを作成し、そのログを動作検証に使用する。新規ページアクセスに追従しない09seAAA、中程度新規ページアクセスに追従する09seBBB、ほとんどの新規ページアクセスに追従する09seCCCの3人である。追従アクセスの制限時間は5分である。前述(3.3節)のようにアクセス解析プログラムが作成するHTMLファイルは教員が授業評価に使用するものと授業の傾向を知るために使用するものとで2つ存在する。教員用HTMLファイルは図12のようにTEACHER, USER, IP, HOSTNAME, USER-AGENT, REQUEST-COUNT, TIME-AVGを表示する。開示用HTMLファイルは図13のようにAVGを表示する。

- TEACHER
教員が新規ページアクセスを行った回数。
- USER
アクセス者がのユーザ認証で登録したユーザ名。
- IP
アクセス者のIPアドレス。
- HOSTNAME
アクセス者のホスト名。
- USER-AGENT
アクセス者の利用者エージェント。
- REQUEST-COUNT
教員の新規ページアクセスに対し、アクセス者が指定時間内にそのページにアクセスした回数の合計。
- TIME-AVG
教員の新規ページアクセスから、アクセス者がそのページにアクセスするまで時間の平均。単位は秒である。指定時間内にページにアクセスしていない場合の時間は考慮されない。
- AVG
授業に参加しているすべての学生のTIMT-AVGの平均。単位は秒である。指定時間内にページにアクセスしていない場合の時間は考慮されない。

USER	IP	HOSTNAME	USER-AGENT	REQUEST_COUNT	TIME_AVG
XXX	10.64.6.XXX	09seXXX.G406.seto-private	Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:13.0) Gecko/20100101 Firefox/13.0.1	0	None
XXX	10.64.6.XXX	09seXXX.G406.seto-private	Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:15.0) Gecko/20100101 Firefox/15.0.1	30	9.56106669903
XXX	10.64.6.XXX	09seXXX.G406.seto-private	Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:17.0) Gecko/20100101 Firefox/17.0	53	3.98286792917

図6 教員用HTML

AVG
6.7719673141

図7 開示用HTML

図1を見るとわかるように教員の新規ページアクセス56回に対して、09seAAAは一回も追従アクセスを行わなかったことがわかる。09seBBBは30回の追従アクセスを行ったことがわかる。09seCCCは53回の追従アクセスを行ったことがわかる。節のはじめに記述した学生の特性と一致する結果となったため本システムはWebアクセスという視点からは学生の学習行動把握を可能にしているといえる。また、図4は正しく全体のアクセス時間の平均を表示しているため正しく動作していると思われる。

7 おわりに

本研究では、授業支援を目的として、学生のWebページアクセスの制限とWebページアクセスの解析を行うシステムを設計し、作成した。本研究のプログラムでは100程度の規模の授業には使用できると考えられる。実際の授業と同じ形式で3人の学生に対しての擬似授業のさいも、問題なく使用することができた。より良いシステムとするために課題点を記述する。

システムの耐久性がまだ低く、学生数が100人を超える授業で使用した場合リクエストが失敗する恐れがある点。

プロキシ切り替えツールやプロキシを設定していないブラウザを学生が使用した場合、アクセス制限ができない点。学生がMACアドレスを変えた場合、正しい出席確認ができない恐れがある点。環境変数がURLに含まれるようなサイトは参加度をうまく取得できない点。これらが今後の課題点である。

参考文献

- [1] Haralanov, M.: Simple Python HTTP Proxy (accessed Dec. 2012). (<http://www.voidtrance.net/2010/01/simple-python-http-proxy/>).
- [2] Hisao, S.: Tiny HTTP Proxy in Python (accessed Dec. 2012). (<http://www.oki-osk.jp/esc/python/proxy/>).
- [3] Lutz, M., (夏目大 訳): はじめてのPython, O'Reilly Japan, third edition (Feb. 2009).
- [4] 武藤 悠矢, 梅村 香輔: proxyによる授業資料のアクセス行動の把握と他サイトへのアクセス制限, 卒業論文 南山大学 数理情報学部情報通信学科 (Jan. 2012).
- [5] Python Software Foundation: Python 2.7ja1 documentation (Nov. 2012). (<http://docs.python.jp/2/index.html>).