

研究ノート

# Moodle用自動・半自動出席ブロックの作成

井 関 文 一\* マ ッ キ ン ケ ネ ス ジ ェ ー ム ス\*

Moodleにおいて自動的及び半自動的に授業の出欠管理を行うブロック（モジュール）の作成をおこなった。この機能を利用して、複数の授業で出欠の管理を行ったので、これらの授業での使用を踏まえて、その利点と欠点についての報告を行う。

**キーワード**：Moodle、自動出欠、出欠管理

## New Auto and Semiauto Attendance Block for Moodle.

Fumikazu ISEKI and Kenneth James Mackin

We present a improved new attendance block (module) function for Moodle. This block manage the attendance of two or more classes by automatic and semiautomatic.

In this paper, we report on the advantage and the fault of new attendance block by these classes.

**Keyword**：Moodle, LMS, auto attendance, e-Learning

## 1. はじめに

今回我々は、Moodle [1] において自動的及び半自動的に授業の出欠管理を行うブロック(モジュール)の作成を行い、この機能を利用して、複数の授業で出欠の管理を行った。今回はこれらの授業での使用を踏まえて、その利点と欠点についての報告を行う。

## 2. Moodleの出欠ブロック(モジュール)

Moodleは様々な授業で使用できる非常に優れたLMS (Learning Management System) である。そのMoodleの汎用性を支えているのが、数々の優れたモジュールまたはブロック群であり、そのようなモジュール・ブロック群の中で学生の出欠管理を行う代表的なものとしては、Dmitry Pupinin氏のattendanceブロック(モジュール) [2] がある。

しかしながら、これは手動で出欠を取りそれを管理するためのものであり、従来の出欠管理をMoodle上に移動させたにすぎない。授業中にコンピュータ(ネットワーク)を使用しない授業であれば、十分機能を利用できるかも知れないが、Moodleを時間中に学生に使用させる授業では機能的に不十分であると思わざるを得ない。

例えば、学生にMoodleを参照させるような授業において、授業時間内にMoodleの該当コースを教室内から参照した場合に自動的に出席とする機能や、学生がMoodle上の出席ボタンをクリックことによって半自動的に出欠を取る機能などがあれば非常に有益であると考えられる。

そこで我々は、Dmitry Pupinin氏のattendanceブロックとモジュール(v1.0.8)を改造し、上記の機能を持たせたautoattendブロックおよびモジュールの開発を行った [3]。図1に新しく開発したautoattendブロックの一括授業登録画面の例を示す。

なお、Dmitry Pupinin氏のattendanceはメイ

ンとなるブロックの他に、対になる補助的なモジュールattforblockが存在する。このモジュールをインストールしなくとも、ブロックだけでも動作可能であるが、モジュールをインストールすることにより出席点を「評定」に反映させることが可能となる。

我々のautoattendもこの機能を引き継いでおり、対となるautoattendmodモジュールをインストールすることにより、出席点を「評定」に反映させることは勿論、さらなる追加機能も使用可能となる。

開発したブロックとモジュールは平成19年4月にβ版として公開し [4]、19年度前期の2つの科目(各科目2クラス)で実際に使用し評価を行った。

## 3. 自動出欠ブロック・モジュールの機能

Dmitry Pupinin氏のattendanceブロックを改造したautoattendブロックでは、本来の点呼方法である手動モードの他に、前出の自動モードと半自動モードを搭載している(図1)。

図1. 授業の登録画面例

なお、記録される出欠の状態としては、「欠席」、「出席」、「遅刻」、「早退」がある。

### \* 自動モード

ユーザが指定時間内にコースにアクセスした場合に自動で出席を取るモードである。対になるモジュールをインストールすることにより、Moodleのcron（クロックデーモン）機能を利用して定期的にアクセスログを解析し、出席の確認を取る。

そのため、どうしても最新の情報との間にタイムラグが生じてしまう。ただし、手動でリフレッシュボタン（図2）をクリックすればその時点での最新の情報に更新することも可能である。

機能の特徴としては以下の点が挙げられる。

- 1) IPアドレスによる制限を追加可能。（教室以外からの出席登録を禁止するため。ただし、ユーザがIPを偽装することは不可能ではない）
- 2) コースのログさえ残っていれば、いつでも出席を取り直せる。
- 3) 機能上、早退者を記録することはできない。

ID	日時	開始時刻	終了時刻	説明	ユーザー	出席方法	出席状態	操作
1	09/27 (月)	10:55	12:25	なし	gabo	半自動	完了	更新
2	09/28 (月)	10:55	12:25	なし	ogawa	半自動	完了	更新
3	09/28 (月)	10:55	12:25	なし	geraf	半自動	完了	更新
4	09/27 (月)	10:55	12:25	なし	ryuga	半自動	完了	更新
5	09/24 (月)	10:55	12:25	なし	tsuj	半自動	完了	更新
6	10/01 (月)	10:55	12:25	なし	-	自動	無出席	更新
7	10/01 (月)	10:55	12:25	なし	-	自動	無出席	更新
8	10/29 (月)	10:55	12:25	なし	-	自動	無出席	更新
9	10/30 (月)	10:55	12:25	なし	-	自動	無出席	更新

図2. コース全体の出欠表の例

9/24まで半自動での出欠が完了している。ただし10/1からは自動で出欠をとるように設定してある。

### \* 半自動モード

ユーザ（学生）が出欠ボタンをクリックすることにより出欠を取るモードである。

このモードを使用する場合は、ブロックと対になるモジュールをインストールしておくことと便利である。モジュールが出席登録の最初エントリボタンとして働き、学生がそのモジュールをクリックすると、その時間帯に有効な授業の出欠確認画面が自動で表示される。その画面の出席ボタンをクリックすることにより出席の記録を行う。

特徴としては以下の点が挙げられる。

- 1) リアルタイムに出席を確認することが可能。
- 2) IPアドレスによる制限を追加可能。（教室以外からの出席登録を禁止するため。ただし、ユーザがIPを偽装することは不可能ではない）
- 3) キーワード（教員が学生に通知する）による制限を追加することが可能。
- 4) 同一のマシンからの出席の追加を禁止するが可能。（代返防止）
- 5) 学生は一々出席を送信する作業をしないといけない。
- 6) 機能上、早退者を記録することはできない。

### \* 手動モード

改造前から存在する本来のモードであり、教師が手動で点呼を取って記録する。融通は利くが、登録には手間がかかる。コンピュータ（ネットワーク）を使わない授業で使用するか、自動・半自動モードで出欠を取った後の修正（早退、公欠など）で使用すると効果的である。

特徴としては以下の点が挙げられる。

- 1) デフォルトのモードが自動・半自動であっても、強制的に手動モードでデータの追加・修正を行うことが可能。
- 2) Dmitry Pupinin氏のattendanceブロック本来の動作モードであり、このモードしか使用しない場合は、出席帳に付けていたのをコンピュータに代えただけとなる。

自動モードと半自動モードはコース内で混在可能であり、半自動でとったものを全て破棄して自動で出欠をとり直すことも可能である（逆は不可）。図2にコース全体の各授業時間での出欠表の例を、図3に点呼中の出欠表の例を示す。

それぞれの授業における出欠の結果は図4のように表示され、この画面で出欠状態を変更することも可能である（手動モード）。

またコース全体の出欠結果は図5のような表として表示可能であり、テキストやExcelのデータとしてダウンロードすることも可能である（ただし、漢字コードは強制的にSJISになる）。

また先に述べたように、対になるモジュールをインストールすることにより、出席点を「評定」に反映させる機能も改造前のモジュールから継承している。

学生からは、当然のことながら自分の出席状況を確認することも可能である。

#### 4. 運用例とβバージョンのバグ

このautoattendブロック（モジュール）を使用して、UNIX論a（半期2クラス。各70名程

#	科目	開始時間	終了時間	説明	モード	出席方法	出席状態	操作
1	UNIX論a(大)	10:00	12:30	なし	-	自動	未	編集
2	UNIX論a(中)	13:00	15:30	なし	-	自動	未	編集
3	UNIX論a(小)	16:00	18:30	なし	-	自動	未	編集

図3. 点呼中の出欠表の例

4 / 5 の授業で点呼中の状態を示す。テスト用の授業なので全学生数は1名になっている。

度)、ネットワークプログラミング演習（通年2クラス。各30名程度）の2科目に於いて、19年4月から一年間の出欠をとった。

UNIX論aでは最初は半自動モードを使用し、ネットワークプログラミング演習では自動モードで出席をとった。

βバージョンということで、実用性の検証を兼ねた実験であったため、幾つかのバグも確認した。

半自動モードでは、サーバの負荷が極端に重くなる現象が現れたが、自動モードでは問題なく動作し、全体として出欠の管理が非常に楽になることを確認した。特に出席点の「評定」への反映では、レポート、テストと合わせて、学生の成績を判定する際に非常に役に立った。

バグについて代表的なものは、

- 1) 自動モードで「点呼終了」のタイミングによっては、「未了」が「欠席」にならない場合がある。（図5）
- 2) 半自動モードでシステム（サーバ）が異常に遅くなる。

の2点であった。

特に先に述べたように 2) の不具合では、サーバが一時的に非常に重くなる現象が現れ、途中で自動モードに切り替えて授業を行った。この障害の原因は、代返の禁止を行うための同一PCからの複数の出席登録を禁止する機能に於いて、出席の登録があった場合にその都度コースのアクセスログを検査し、過去のIPアドレ

#	ユーザID / 氏名	出席状況	出席方法	IPアドレス	備考
1	██████████	○ ○ ○ ○ ○	未点呼	-	
2	██████████	○ ○ ○ ○ ○	未点呼	-	
3	██████████	○ ○ ○ ○ ○	半自動	172.28.2.13	

図4. 個人個人の特定授業の出欠表の例

「未」状態は学生がその時点でコースに登録していなかったことを示す。この画面から出欠の状態を変更することも可能。

ユーザID / 氏名	04/24	05/08	05/15	05/22	07/03	07/10	07/17	07/24	出席	早退	欠席	未	点/20	%
● [REDACTED]	未	出	未	出	出	欠	出	出	9	0	0	2	10	69.2%
● [REDACTED]	未	出	出	出	欠	出	出	出	10	0	0	1	20	75.0%
● [REDACTED]	出	出	出	出	出	出	出	出	11	0	0	0	22	84.4%

図5. 個人個人のコース全体お出欠表の例（一部省略）

バグにより一番目の学生の5/15分が「未」状態になっている。なお、4/24の「未」状態は学生がその時点にコースに登録していなかったため、正常な表示である。

スの履歴を検査していたためであった。

出席の登録の度にアクセスログをチェックするため、サーバに重い負荷がかかった。

現在開発中のRCバージョンでは上記の2点については既に修正済みである。

## 5. 問題点とまとめ

半期間プログラムを使用することにより、一部問題はあったが、この種の機能が非常に有用であると感じた。ただし、運用に当たって注意すべき点もあった。

半自動モードでは学生は能動的に出席ボタンをクリックするため、出席をとったという自覚がある。一方自動モードでは、出欠が自動でとられるため、学生は出欠をとっているということに自覚せず、ゲストのままコースの資料を参照する場合がある。最後に「小テスト」など、ログインの必要のあるサービスを利用する状況になって、初めて自分がログインしていないことに気が付き、慌てて教員に申請に来る学生が時々存在した（この場合は手動モードで「遅刻」を「出席」に変更した）。

この場合の対策としては、コースをゲスト不可にして、最初から強制的にログインさせるという処置が効果的である。ゲストを許可するコースでは、半自動モードで出席をとる方が問題（点呼ミス）の発生率は少なくなると思われる。

また、実際のプログラムの使用を通して、以下の機能の必要性も感じた。即ち、

1) 欠席者の一覧表示機能。

2) 1科目に複数のクラスがある場合のグループ機能。

の2点である。

特に2)については、現時点では集計後に手動で分ける必要があり、2クラス程度ならそれ程の負担でもないが、3、4クラス以上となると大変な手間となることが予想されるので、正式なバージョンでは是非搭載したい機能であると考えている。

この研究は東京情報大学平成19年度共同研究の援助を受けて行われました。

## 参考文献・参照

- [1] <http://moodle.org/>
- [2] <http://moodle.org/mod/forum/view.php?f=125>  
Moodle Attendance module Forum
- [3] <http://www.netp.tuis.ac.jp/moodle/course/view.php?id=16>
- [4] <http://www.infosys.tuis.ac.jp/xoops/modules/wmpdownloads/viewcat.php?cid=25>