

LMS による学習進捗情報記録における問題点 ～ LMS の「学習足跡」への対応の一例～

長谷川 紀幸

横浜国立大学 工学研究院等技術部

1. はじめに

本学では 2016 年度から全学システムとして LMS を導入しており、コロナ禍以前から先駆的教員等によって授業での資料配布、簡単な小テストやレポートの提出・回収として利用されてきた。

2020 年度のコロナ禍では否応なく全国的に遠隔授業を実施することとなり、それまで LMS を利用してこなかった教員も学生も利用せざるを得なくなり、LMS を利用する学習が一気に広まることになった。2022 年度から対面授業が限定的に再開され、2023 年では新型コロナウイルス感染症の 5 類移行に伴い「コロナ禍以前」と同様の授業形態に戻ったが、コロナ禍で広まった LMS の利便性が対面授業でも取り入れられるようになった。

一方、発表者が授業サポートとして関わっている「プログラミング入門（理工学部数物・電子情報系学科数理科学教育プログラム及び情報工学教育プログラム）」「プログラミング基礎（理工学部数物・電子情報系学科電子情報システム教育プログラム）」では、コロナ禍以前より LMS を利用してブレンディッド・ラーニングを実施してきたが、学習者にとって e-Learning コンテンツで学習しながら別途エディタでコーディングを行うことは、作業環境として慣れないこともあり教授者側が意図していないような使い方もするため、学習効率が上がらないことも生じている。

また、教授者側が意図していないような使い方により、LMS における本質的な問題にも改めて気づかされることとなり、本発表ではその対応について一例を報告する。

2. 受講者の学習状況と学習進捗情報記録における問題

上記の授業科目はどちらも 1 年生秋学期に実施しているプログラミングの入門科目であり、座学による講義の回と PC を使って講義で学んだ知識を課題の実施で身につける演習の回を交互に行う形態で授業を進めている。

授業における教材は学習テキストと講義で使用するスライドを pdf ファイルで配布し、演習で実施する課題の説明は LMS 上の e-Learning コンテンツを閲覧する。開発環境としては初学者および非情報系の学生も受講しているため、GDB Online^[1] のオンライン・コンパイラを使用してプログラミングを行うことを基本としている。（他の IDE やシェル+エディタを使うことを禁止はしていない）

1 年生秋学期に実施しているので、デジタル・ネイティブ世代とは言え大学で PC も LMS も利用してきた経験が長くないこともあり、自分の PC 上で、pdf で配布されるテキストブックを読み、e-Learning コンテンツで説明される課題説明を閲覧しながら、IDE やエディタでコーディングし、わからないことは Web ブラウザで検索し、作成したプログラムを実行するような同時に複数のウインドウを切り替えながら思考と実装をスムーズに進めていくことは容易ではないようである。

コロナ禍以前のブレンディッド・ラーニングでは LMS を使用しているとはいえ、学生は目の前におり、学習に取り組む姿勢や進捗などを机間巡視で確認できることもあって、受講生の学習状況などのデータをそれほど重視して取得・分析していなかったが、コロナ禍の遠隔授業では受講生がどのように学習しているのかは授業改善の観点からも重要となっており、詳細な学習分析までは行っていないが受講生の学習状況を注視するようになった。

本学の LMS でも、他の LMS 同様に LMS 上の教材における、閲覧記録、学習進捗度などの情報を取得することができるため、受講生の学習状況を注視し、これらの情報を確認していくと、あることに気が付き始めた。

それは、pdf で配布している資料をダウンロードしていない受講生もちらほらいるが、演習の課題を説明している教材を閲覧している情報が無い受講生が多いのである。

Pdf で配布している教材は自分にあった学習教材で学べば良いので問題ではないが（今後はその分析も必要であるかもしれないが）、演習課題の説明も見ずに演習課題を提出しているのは一体、どういうことであるのだろうか、疑念がわく。

これは、一つには本学で導入している LMS に詳細な記録を教授者に提示する機能が貧弱であるため、実際には演習課題の説明の教材を閲覧しているにもかかわらず、教材を閲覧した記録を確認することができないことである。これは、個別 LMS の問題であるので、LMS の製作側になんとかしてもらえない（が、ほとんどは期待できないであろう）

もう一つは、LMS の利用の仕組みそのものの問題であり、こちらは学習者による「LMS に想定されていない操作」によって、「教材を閲覧した足跡」などが正しく LMS に記録されないものである。

3. LMS における問題点

個別 LMS の問題点は個別の問題であるので、LMS 利用の仕組みにおける問題点について事例を取り上げる。

先に述べたように、オンライン・コンパイラ環境でプログラミングを行うため、LMS 上のコンテンツとプログラミング環境は別々のウインドウあるいはタブで作業することになる。演習授業におけるコーディングのフェーズでは LMS 上のコンテンツは「邪魔」であるが、教授者としては複数のウインドウやタブを切り替えて課題説明教材を時折確認しながらプログラミングを行うことを想定していた。この想定内であれば、説明教材を LMS の操作で読み進める、閲覧終了することで正しく学習進捗情報が記録される。

一方、LMS で教材を閲覧しているウインドウやタブを「閉じて」しまい、別のウインドウやタブのオンライン・コンパイラでコーディングを始めると、LMS はコンテンツへのアクセスのログは残るが「LMS の操作」としてコンテンツの閲覧を終了していないので、LMS にはその教材を「読み終えた」情報が記録されず、従って、教材の学習進捗状況を確認すると「閲覧していない」ままになってしまう。

この問題は「ステートレス」な LMS 利用の仕組みからくるものであり、LMS の機能ではいかんともしがたいものである。

4. 対応の一例

上記の問題に対しては、LMS の機能ではなく、コンテンツ側で対応することが可能である。一例として発表者は LMS が SCORM^[2] に対応していたため SCORM コンテンツを作成し、教材を閲覧した時点（ブラウザで HTML を読み込んだ時点）で、LMS に教材閲覧の情報を記録するように対応したことで、少なくとも「提出する課題の内容を見ていないため、課題の内容がわからないはずであるのに、課題を提出している」ことに対する疑念を解消することができた。

参考文献

[1] GDB Online, <https://www.onlinegdb.com/>（2024年1月10日閲覧）

[2] SCORM Website, <https://scorm.com/>（2024年1月10日閲覧）