Vol.2022-IOT-58 No.14
2022/7/13

学習管理システムを利用する人手による業務自動化の懸念とその対策について

久保田 真一郎[1,a)]

概要：学習管理システムに記録される学習者の情報を使い，学習支援や学習状況の可視化などを行う人手の業務が発生している．そこで RPA(Robotics Process Automation) による業務の自動化を目指すことにした．利用者が GUI を用いる動作を再現するプログラムを利用して，作業を自動化することができれば，通常のアカウントを利用してアクセスし，フィードバックに必要な情報を取得して自動加工することができる．これら業務の自動化を検討した際の運用面やセキュリティ面で懸念される問題点について整理し，対策などについて共有する．

キーワード：Robotics Process Automation(RPA)，学習管理システム (LMS)

Countermeasure and Concerns About the Automation of Human Work Using Learning Management Systems

Abstract: Manual tasks to support learning and visualize learning status using learner information recorded in the learning management system is occurring. Therefore, we decided to automate these tasks using RPA (Robotics Process Automation). If we can automate these tasks using a program that reproduces the actions of a user using a GUI, the user can access the system with an usual teacher role, obtain information necessary for feedback, and automatically process the information. This paper will summarize the issues of concern in terms of operation and security when considering the automation of these operations, and share countermeasures and other information.

Keywords: Robotics Process Automation(RPA), Learning Management System(LMS)

1. はじめに

学習管理システム (LMS;Learning Management System) は，学習者の閲覧やクイズに回答した結果など，ある程度の学習者の行動を示す情報 (以降，学習ログ) を保持している．教授者は，学習ログを閲覧して，学習者へのフィードバックに役立てることができる．学習ログを，データベースや WebAPI などを使い，システム間で連携することで学習者の状況を可視化する取り組みがある [1], [2]．しかし，すべての教員が自身の授業のために学習状況を可視化するシステムを構築することは容易ではない．システム間の連携を行った先行研究の取り組みは，システム管理者が関与する開発になり，システム管理者本人やシステム管理者と同じプロジェクトメンバーなどの限られた人が関与する．また，システム間を連携する場合には，データベースや WebAPI へのアクセス許可や特定 IP アドレスからのアクセス許可などが必要となる．アクセスを許可するだけセキュリティ面で脆弱になるため，他の利用者がアクセスできないネットワークの整備など対策が必要となる．

授業によって学習ログの質も異なると考えられ，教授者ごとにフィードバックのための判定も異なると考えられる

1 熊本大学
Kumamoto University, Kumamoto, Kumamoto 860–8555, Japan
a) kubota@cc.kumamoto-u.ac.jp

ⓒ 2022 Information Processing Society of Japan

ことから，教授者は独自に LMS に表示される学習ログを確認してフィードバックしていると考えられる．例えば，LMS の 1 つである Moodle であれば，教授者は「評定」機能や「レポート」機能を使い学習者の状況を確認して，対象者にフィードバックできる．実際に，私が担当する授業では，学習状況を可視化するために，Moodle の「評定」機能からダウンロードしたファイルを処理して，再度「評定」に反映させる人手での業務が発生している．これを自動化するには「評定」機能や「レポート」機能で学習者を表示し，その表示を確認して，必要な学習者にフィードバックを提供するというプログラムが必要になる．Web ブラウザでアクセスする操作やログイン操作，学習ログを表示する操作を制御するプログラムが必要で，コマンドの実行と GUI 操作のプログラムと組み合わせた RPA(Robotics Process Automation) システムの構築が必要になる．本研究では，人が LMS に対して行う GUI の操作を再現するプログラムとして，PyAutoGUI[3] を使用することにした．通常のアカウントを使用してアクセスし，必要な情報を取得可能であることから通常のセキュリティ対策をとりながら，その業務を自動化できる可能性があると考えた．

2. 開発環境

PyAutoGUI は，Python で動作するモジュールで，PyAutoGUI 自体も様々なモジュールを応用して構成されている．開発環境の CPU は i710610U，メモリは 16G で，OS は Ubuntu 20.04.4 LTS を利用した．Python のバージョンは 3.8.10，PyAutoGUI のバージョンは 0.9.53 で開発に取り組んだ．

3. 自動化する作業

Moodle の学習コースで，学習活動の完了を記録する「レポート」機能により，特定の学習活動 (例えば，「課題」機能) に対して学習者の完了と未完了を示す一覧を表示し，その表示情報を CSV ファイルでダウンロードする．ダウンロードした CSV ファイルをもとに完了した学習者と未完了の学習者の一覧をファイル出力する作業を仮定する．Moodle で必要となる操作手順は，次のようになる．

(1) Moodle の学習コースを Web ブラウザに表示する
(2) Moodle の「レポート」機能で，「活動完了」を表示する
(3) 必要な学習活動の種類を選び，表示された一覧を CSV でダウンロードする

CSV ファイルをダウンロードした後は，GUI なしのプログラミング処理で，完了と未完了の判定処理，対応する学生の一覧をファイル出力する処理で所望の作業処理は完了する．本研究では，上記 Moodle で必要な操作手順を PyAutoGUI で自動化する際に，検討した内容を報告する．

```
1  webbrowser.open("学習コースのURL")
2  time.sleep(10)
3  pyautogui.screenshot('保存ファイルパス')
4
5  target_box = pyautogui.locateOnScreen(
       img_loginusername)
6  x, y = pyautogui.center(target_box)
7  pyautogui.moveTo(x, y, 1)
8  pyautogui.click()
9  pyautogui.typewrite('ユーザ名', interval=0.1)
10 pyautogui.press('\t')
11 pyautogui.typewrite('パスワード', interval=0.1)
12 pyautogui.press('\t')
13 time.sleep(2)
14 pyautogui.screenshot('保存ファイルパス')
15
16 pyautogui.press('enter')
17 time.sleep(10)
18 pyautogui.screenshot('保存ファイルパス')
```

図 1 LMS へのログイン処理
Fig. 1 Program Code to Login in LMS

3.1 LMS へのログイン操作

処理「1. Moodle の学習コースを Web ブラウザに表示する」のためには，学習コースの URL をもとに Web ブラウザに表示する処理と LMS へのログインする処理についてプログラムを作る必要がある．Web ブラウザで特定の URL を開くモジュールとして，webbrowser モジュールがあるので利用した．webbrowser モジュールを使って学習コースの URL を開くと，LMS の認証画面が表示される．そのため，PyAutoGUI を使って，ユーザ名欄とパスワード欄に適切なユーザ ID とパスワードを入力するようプログラムコードを記述した (図 1).

1 行目が webbrowser モジュールにより，学習コースの URL を Web ブラウザで開く処理である．学習コースにアクセスして表示が完了するのを待つため，time モジュールの sleep 関数を使用して，10 秒の間，処理を停止している．本来は表示をチェックするような処理とあわせ，表示されたら次の処理に進むように記述したいが，今回は実装していない．処理結果を記録して確認するために，処理ごとに Web ブラウザの表示を，PyAutoGUI モジュールの screenshot 関数を使用して，スクリーンショットを記録している（図 1 の 3,14,18 行目).

次に，ユーザ名とパスワードの入力処理は，図 1 の 5 行

```
webbrowser.open_new_tab("ダウンロードのためのURL")
time.sleep(10)
```

図 2 CSV ファイルのダウンロード処理
Fig. 2 Program Code to Download CSV File

目から 13 行目に記述されている．ユーザ名の入力欄をアクティブにするために，ユーザ名の入力欄の場所をクリックする必要があったので，ユーザ名の入力欄の画像をもとに一致する画像の座標を返す PyAutoGUI の locateOnScreen 関数を用いた (5 行目)．6 行目でその重心座標を変数 x,y に代入し，7 行目でマウスカーソルを重心座標へ移動して，8 行目でその重心座標でクリックするという処理を行っている．PyAutoGUI の typwrite 関数は引数の文字を interval で指定した間隔で入力する関数で，9 行目，11 行目でユーザ名とパスワードを入力している．入力欄の移動は Tab キーを押して移動している (10 行目，12 行目)．

図 1 のプログラムコードを実行したところ，デスクトップ上に Web ブラウザが表示され，ユーザ名とパスワードが入力され，学習コースにログインすることができた．

3.2 学習者の学習活動の一覧をダウンロード

次に，学習者の学習活動の一覧を CSV でダウンロードする処理である．まず，学習者の学習活動の一覧を表示し，次にその一覧をダウンロードするという処理になる．Moodle の場合，URL に「/report/progress/」で活動完了の一覧へのアクセスで，「course=」で所望の学習コースを指定でき，「activityinclude=」で活動の種類を指定できる．また，ダウンロードするファイルの URL も指定可能である．そのため，GUI による操作の必要はなく，Web ブラウザで指定の URL を開くことでダウンロード処理が行われる．そのため，webbrowser モジュールの open_new_tab 関数にダウンロードのための URL を指定する (図 2) ことで，所望の CSV ファイルをダウンロードできた．

4. RPA 処理における問題点と解決策の検討

前節では人の GUI 操作をコード化した．前節のコードを実行すると，デスクトップ上に Web ブラウザが表示され，ユーザ名とパスワードが入力され，学習コースにログインすることができた．そして，その状態で CSV ファイルをダウンロードするための URL を開くだけで CSV ファイルをダウンロードできた．しかし，動作中のデスクトップで故意に操作すると，所望の結果とはならない．例えば，ユーザ名の入力欄をアクティブにする処理の途中でマウスカーソルを別の場所に動かすとその場をクリックしてしまい正しく動作しない．また，アクティブウィンドウを変える介入をすると Web ブラウザは操作できず正しく動作しない．悪意のあるユーザであれば，ユーザ名とパスワードを入力のタイミングで，テキストエディタなどをアクティブウィンドウにし，パスワードを盗むことができる．人が行う GUI の操作を再現するプログラムなので，当然であるが，cron などを使った定期的なバッチ処理を行う場合でも，今回のプログラムが動作している間に別の処理を実行することはできない．複数の専用 PC を準備することで並列処理は実現可能ではあるが，RPA による自動処理を増やすたびにパソコンを増やすことになる．

RPA による自動処理が動作する間もデスクトップに影響しない方法として，X Virtual Frame Buffer(Xvfb)[4] の使用を検討した．Xvfb は X サーバの 1 つで，ディスプレイや入力デバイスなしで動作する X サーバである．RPA による自動処理を行うためのディスプレイとして仮想メモリ上で動作させることで，デスクトップの操作の影響を受けず，仮想メモリ上で正しくコードが動作すると考えた．

5. Xvfb による RPA 処理の実行

Xvfb はディスプレイ変数を指定して起動できる X サーバなので，デスクトップ (ディスプレイ変数「:0」) 以外のディスプレイ変数を指定して Xvfb を起動した．その上で，指定したディスプレイ変数上（以降，「仮想デスクトップ」と呼ぶ）で今回作成したコードを実行した．デスクトップで Firefox を利用中の場合に，同じプロファイルを使用して新しい Firefox を仮想デスクトップ上で起動することができなかったので，Firefox に新しいプロファイルを作成し，新しいプロファイルを指定して Firefox を起動し URL を開くことで対応した．ディスプレイ変数を変え，Firefox のプロファイルを準備することで，並行して RPA 処理が可能と考えられる．しかし，並行して Firefox で処理する場合には，その数だけ Firefox のプロファイルが必要となる．

vmstat コマンドで空きメモリ容量，CPU 使用率を記録しながら，今回のプログラムコードをデスクトップと異なるディスプレイ環境変数で仮想デスクトップ上で実行した．同様に，デスクトップで今回のプログラムコードを実行し記録した．測定は同時に行わず，それぞれ別の時間に行った．デスクトップと仮想デスクトップとで CPU 使用率を比較したグラフを図 3 に示す．また，空きメモリ容量を比較したグラフを図 4 に示す．実行時間は，デスクトップを用いる場合に 71 秒で，仮想デスクトップを用いる場合に 80 秒となった．Xvfb を起動することもあり，仮想デ

 Vol.2022-IOT-58 No.14
2022/7/13

スクトップの方が実行時間を要している．CPU の使用率を見ると Xvfb の起動で立ち上がりがずれているが，ほとんど同じ程度およびタイミングで CPU 使用率に変化が見られる．使用率の平均は，デスクトップで 3.5 % (標準偏差 4.32)，仮想デスクトップで 3.1 % (標準偏差 4.56) とほとんど変わらない．空きメモリ容量では，仮想デスクトップ利用時の方がデスクトップ利用時に比べて余裕がある傾向が見られる．デスクトップの画面描画に使用している可能性がある．仮想デスクトップで今回のプログラムコードを実行した場合，空きメモリ容量の平均は 9569MB(標準偏差 259.8) であった．デスクトップで実行した場合の空きメモリ容量の平均は 9036MB(標準偏差 414.5) であった．平均で 500MB 近く差が見られる．

6. おわりに

学習者のために学習者の情報を LMS から取得し，その情報を使って必要な学生にフィードバックを行う活動を考えたときに，毎回，同じ処理を人手で行うのではなく，人が LMS に対して行う GUI の操作を再現するプログラムを使って自動化することを検討した．PyAutoGUI は，1 つの解決を与えてくれるツールとなったが，デスクトップがプログラムによって占有され，他の処理ができないことが問題となった．そこで，Xvfb を利用した仮想デスクトップを利用し，デスクトップを利用しない手法での解決を試みた．ディスプレイ変数を制御することでデスクトップでの作業の影響を受けることなく，仮想デスクトップでの動作が確認できた．デスクトップと仮想デスクトップでプログラムコードを実行した場合，CPU 使用率や空きメモリ容量など，ハードウェアリソースに差はなく，仮想デスクトップの方がメモリを節約できる可能性があることがわかった．今回の使用環境である Web ブラウザの Firefox を使用する場合，同一プロファイルでデスクトップと仮想デスクトップの両方で Firefox を起動することができないという問題があり，処理ごとにプロファイルを作成する必要がある点は運用上の問題と言える．セキュリティ面では，パスワードを入力するプログラムが実行される際に，テキストファイルなどに出力させることができ，パスワード盗まれる危険があった．

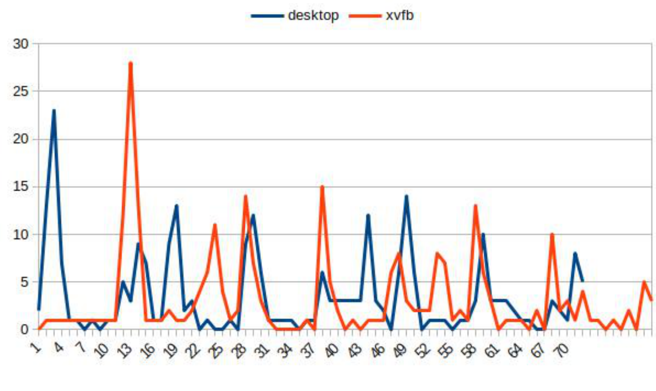

図 3　デスクトップと仮想デスクトップでの CPU 使用率の比較
Fig. 3　Comparison of CPU Rate between Desktop and Virtual Desktop

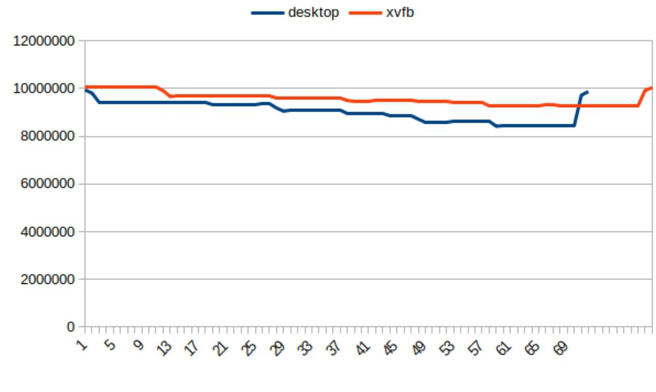

図 4　デスクトップと仮想デスクトップでの空きメモリ容量の比較
Fig. 4　Comparison of Free Memory Capacity between Desktop and Virtual Desktop

謝辞　この研究の学習者へのフィードバックを返すという文脈は，JSPS 科研費 20H04297 の助成によるもので，RPA システムは JSPS 科研費 JP19K12273 の助成によるものです．

参考文献

[1] 古川雅子, 上田浩, 浜元信州, 中村素典, 山地一禎: 学認 LMS における標準規格に基づく教材配信及び学習履歴取得システム, 情報処理学会研究報告, Vol.2019-IOT-47, No.13, pp.1–4, 2019.
[2] 中野裕司, 喜多敏博, 杉谷賢一, 松葉龍一, 久保田真一郎, 宇佐川毅: 学務システム、LMS 等と連動した学習成果可視化システムの開発, Vol.2018-CLE-24, No.4, pp. 1–5, 2018.
[3] `https://github.com/asweigart/pyautogui` (2022 年 6 月 20 日確認)
[4] `https://www.x.org/releases/X11R7.6/doc/man/man1/Xvfb.1.xhtml` (2022 年 6 月 20 日確認)

ⓒ 2022 Information Processing Society of Japan