演習状況モニタリングシステムの開発とその評価

森田直樹*1
Email: morita@tokai.ac.jp

*1：東海大学情報通信学部通信ネットワーク工学科

◎Key Words プログラミング，モニタリング，履歴

1. はじめに

本研究では，プログラミング演習の授業において，受講者が PC に対して行ったすべての操作を，OS のシステムメッセージよりグローバルフックを用いて取得するシステムを開発したので報告する。

プログラミング演習において，受講者や教師を支援するシステムは，数多く存在する[(1)(2)]。特にプログラミング演習では，ソースコードを編集する過程が重要である。ソースコードを作成する過程には，受講者がどのように演習に取り組んだのかを講師が判断する情報が豊富に含まれる。従来の方法は，ソースコードを編集する過程を取得するために専用のエディタが必要であった。

本研究で開発したシステムは，専用のエディタを用いることなくソースコードを編集する過程を取得することができる。さらに，本システムは，ソースコードを編集する過程だけでなく受講者が行うプログラミングに関するすべての過程を蓄積することができる。

2. 本研究の目的

本研究の目的は，専用のエディタを用いることなくソースコードを編集する過程を取得し，かつ，コンパイルや実行ファイルの実行などのプログラミングに関するすべての動作を記録することである。

本研究で想定するプログラミング演習の環境は，OS は Windows とし，実行ファイルを作成する方法は，Visual Studio IDE を用いてビルドする，または，Visual Studio コマンドプロンプトからコンパイラを起動しソースコードをコンパイルする方法をとる演習とする。

Visual Studio は，Microsoft 社製のソフトウエア開発ツールであり，アプリケーションを効率よく開発することができる。そのため，初心者からプロのプログラマまで利用する。Visual Studio を用いて実行ファイルを作成する方法は，ソースコードの作成からビルドまでのすべての工程を Visual Studio IDE 上で実施する方法と，ソースコードは別途作成し Visual Studio コマンドプロンプトからソースコードをコンパイルする２つの方法がある。

Visual Studio IDE 上で実行ファイルを作成する方法は，ソースコードを作成する前にプロジェクトを準備したり，Visual Studio IDE の操作を習得したりする必要がある。そのため初心者には，プログラミング以外の要素も習得する必要はあるが，眼には見えないメモリ空間上の変数の値を可視化したり，自分が作成したソースコードをどのような順番で実行しているのかトレースしながら確認したりすることができる。

Visual Studio コマンドプロンプトからソースコードをコンパイルする方法は，プログラミング以外の最低限の操作のみで演習を行うことができ，プログラミング初心者を対象とした授業で用いられることが多い。

本研究では，これら二つの方法において，プログラミング演習を行うすべての過程を取得することを目的とする。

3. プログラミング過程可視化システム

3.1 システム構成

本システムは，プログラミング過程取得モジュール，プログラミング過程蓄積モジュール，プログラミング過程再現モジュールから構成される。

プログラミング過程取得モジュールは，受講者の PC 上で動作し，受講者が PC を操作するイベントの取得とその時の受講者の PC 画面をキャプチャした画像をプログラミング過程蓄積モジュールに送信する。プログラミング過程蓄積モジュールとプログラミング過程再現モジュールは，Web サーバ上で動作し，蓄積モジュールは，プログラミング取得モジュールから送られたデータをサーバ上に蓄積し，再現モジュールは，Web ブラウザ上で受講者の PC 画面を再現するための HTML を形成する。

3.2 プログラミング過程取得モジュール

本モジュールは，受講者の演習過程を取得するアプリケーション OperationLog.exe とグローバルフックを行うための OperationLog.dll，Visual Studio IDE の機能を拡張するアドイン，Visual Studio コマンドプロンプト上での操作を取得するバッチファイル cl.bat から構成される。

本研究で開発した OperationLog.dll は，OS に対して SetWindowsHookEx 関数を用いてグローバルフックを行いてウインドウメッセージを監視する。これにより，マウス操作やキー操作を操作対象となるウインドウに反映させながら，その操作イベントを取得することができる。本 OperationLog.dll が監視する項目は，マウス左ボタン押し下げ，マウス左ボタンダブルクリック，マウス右ボタン押し下げ，キーボードからの文字の入力である。

本研究で開発した OperationLog.exe は，DLL が監視中に該当操作を取得したタイミングで受講者の PC 画面のキャプチャを行う。さらに，その該当の操作が，画

面全体のどの部分に対しての操作であるかを確認しやすくするために，キャプチャ画像の対応箇所に赤枠でハイライト処理を施す。これにより，キャプチャした画像を後から振り返った時に，ウインドウ全体に対してどの部分に行った作業であるのかを確認することができる。

テキストエディタ上のソースコードの取得法

メモ帳やTerapadなどのテキストエディタは，ウインドウ内にEditクラスのエディットコントロールが配置されており，ユーザは，その領域内で文字列の編集を行う。エディットコントロール内にある文字列は，WM_GETTEXTメッセージにより取得できる。この特徴を利用して，OperationLog.exeは，テキストエディタになりすましてOSにWM_GETTEXTメッセージを発行し，受講者が作成しているソースコードを取得する。取得したソースコードは，同時に取得したキャプチャ画像と関連付けを行い，プログラミング過程蓄積モジュールへ送信する。

Visual Studio IDE上のソースコードの取得法

Visual Studio IDE上のコードエディタは，VsTextEditPaneクラスから構成される。そのため，コードエディタ上で編集されるデータは，メモリ空間に存在するデータを基に描画データとして，つまり，画像としてユーザに提供される。そのため，役割ごとに色分けを施したり，あるまとまりを折りたたんだ状態で表示したりすることができる。その一方で，テキストエディタのようにWM_GETTEXTメッセージではコードエディタ上のソースコードを取得することができない。本研究では，コードエディタ上のソースコードを自動保存するアドイン開発することにより，ソースコードの作成過程をテキストデータとして取得できるようにした。

コマンドプロンプトのコンパイルの結果を取得法

受講者がコンパイルしたタイミングでコンパイルメッセージなどの情報をプログラミング過程蓄積モジュールへ送るために，コンパイルコマンドと同じ名前のバッチファイルcl.batを記述した。バッチファイルに記述した内容は，通常のコンパイル処理に加え，コンパイルメッセージをファイル化し，ソースコード，メッセージファイル，ユーザ識別情報をプログラミング過程蓄積モジュールへ送信するための処理である。

Visual Studio IDEのコンパイルの結果を取得法

Visual Studio IDEを用いてコンパイルを行う場合には，コンパイルメッセージは，該当のフォルダにHTMLファイルとして自動保存される。本研究で開発したアドインは，コンパイルメッセージが記述されたHTMLファイルをユーザ識別情報と共にプログラミング過程蓄積モジュールへ送信する。

3.3 プログラミング過程蓄積モジュール

プログラミング過程蓄積モジュールは，Web サーバ上で動作する。本モジュールは，受講者のPC上で動作するプログラミング過程取得モジュールから送られたデータを，ユーザ情報ごとに，画像データとその画像に関するウインドウ情報やソースコードなどのテキスト情報を時系列で管理する。

3.4 プログラミング過程再現モジュール

Web サーバ上で動作するプログラミング過程再現モジュールへのアクセスは，Web ブラウザを用いる。本モジュールが提供する情報は，HTML で構成されており３階層構造となる。

1 階層目である TOP 画面は，受講者全員の現在の状況が確認できるリストが提供される。リストは，テーブルタグを用いて構成されており，その要素は，学生証番号，最後に取得した画像に関連付けられたテキストデータである。テキストデータがソースコードの場合には，カーソル行の１行が表示される。これらの情報は，Ajax を用いて常にプログラミング過程再現モジュールと通信を行う。これにより，再読み込みの操作を行うことなく情報を更新し提供することができる。操作の過程を確認するには，学生証番号をクリックすることにより次の画面に移動する。

2 階層目で提供される情報は，TOP 画面で選択した受講者のテキスト情報を時系列から逆順でリスト化したものである。リストの構成要素は，プログラミング過程蓄積モジュールが取得したタイムスタンプ，画像に関連付けられたテキスト情報であり，その情報がソースコードの場合には，編集行を，コンパイル項目の場合には，コンパイラのメッセージが表示される。コンパイラのメッセージに対しては，エラー時は赤色，成功時は青色で色分けが施される。確認したい時間帯のタイムスタンプをクリックすることにより受講者のPC画面のキャプチャ画像が表示される 3 階層目へ移動する。

3 階層目で提供される画面は，受講者のPC画面のキャプチャ画像を，サムネイル形式で複数枚表示したり，1 枚ずつ表示したりすることができる。画像は，ボタンにより進めたり戻したりすることができる。

4. おわりに

本研究では，プログラミング演習の授業において，受講者が行ったすべての操作を監視し取得するシステムを開発した。具体的には，特別なエディタを用いることなくソースコードの編集過程を取得したり受講者のPC画面を取得したりすることができ，必要に応じてWeb ブラウザで演習過程を再現できるシステムである。

本システムには，プログラミング演習の過程がテキストベースの情報として豊富に蓄積される。これらの情報を分析することにより，さまざまな支援を行える可能性がある。これらのデータの有効活用や，本システムを長期的に利用し定量的な有効性の検証を行うことを今後の課題とする。

参考文献

(1) 井垣宏，斎藤俊，井上亮文，中村亮太，橋本真二：”プログラミング演習における進捗状況把握のためのコーディング過程可視化システムC3PVの提案”，情報処理学会論文誌，Vol.54 No.1，pp.330-339，(2013).

(2) 片桐由裕，立岩佑一郎，山本大介，高橋直久：”受講者の操作履歴の分析機能を用いたプログラミング指導者支援システム”，信学技報，ET2009-83，pp.181-186，(2009).