

# プログラミング授業支援システムとトラブル軽減について

松永 豊

情報教育講座

## For Programming Class Support System and Trouble Mitigation

Yutaka MATSUNAGA

Department of Information Sciences, Aichi University of Education, Kariya 448-8542, Japan

### 1. はじめに

演習系授業において日々の提出物は非常に重要である。学習者本人にとっては何らかのスキルを獲得するためのトレーニングの結果という意味合いもある。また、教師側にとっては学習の進捗を把握するための判断材料になることは言うまでもない。

筆者は本学情報コースにおいてプログラミング授業を複数担当しており、プログラミング授業を能率よく進めるために様々な授業支援システムを開発してきた[1][2][3]。基本的なコンセプトは学生側、教師側双方にメリットがあるシステムの構築である。学習者本人にとっての成長記録という側面を考慮して開発に取り組んでいる。

すでに運用中のシステムであり、システムの構築に当たってはかなり慎重に仕組みを考えたつもりであったが、問題点が発生したので一部大幅に変更することにした。そこで、本研究ではシステムの改良を通して授業支援システムで発生しうるトラブルの種類や軽減方法について議論・考察することを目的とする。

### 2. 提出物の管理に関して

いうまでもなく膨大な提出物は収集自体に一定の価値がある。様々な分析データになるだけでなく、不正チェック等にも使用可能である。ただし、それはあくまでも再利用可能なデータへの変換、すなわちデジタル化を前提としている。

提出物をデジタル化することのメリットは非常に多いが、無論、デメリットもある。システムの詳細に入る前に、まず、メリット・デメリットについて考察しておこう。なお、レポートに関するものだけでなく、一部、一般的な(普遍的な)ものも取上げて含めてある。

#### 【主なメリット】

- 分析等が容易になる。

- 最終的にデジタル化する前提であればデジタルデータを提出させることで入力の手間が省ける。
- 紙と違ってかさばらないため、数年前のデータと比較するなどにも簡単に行うことが可能。
- 遠隔地等の学生がわざわざレポート提出のためだけに大学に来る等の必要がなくなる。
- 様々な箇所で自動化が可能となる。

#### 【主なデメリット】

- デジタル化したデータは簡単にコピーができてしまうためレポートの模倣(いわゆるコピーペ)を助長する可能性がある
- 提出期間中(特にメ切直前)に、システムトラブルを起こさないよう注意が必要である。
- データの扱い自体に注意が必要。紙のレポートの場合は提出物が一瞬にして消えるということは考えにくいですが、デジタルデータの場合は何万人分であれ一瞬にして消滅する可能性がある。
- アンケートのようなデータ収集にも有効であるが、いうまでもなくデジタル機器が必要になるため、紙を配って回収のように手軽に実施できない。

レポートをデジタルデータとして提出させるというのは、見方を変えれば、学生にデータ入力をお願いしているとも言える。そのため、一般的には教師側の負荷の軽減につながるが、提出の仕組みで手を抜くと思わぬしっぺ返しを食らう。たとえば、あるルールを守って入力するように指示したのにそれを守らずに入力されてしまった場合、修正のため返ってコストがかかるかもしれない。もちろん、学生側に悪気はなくても、である。つまり、学生にとってミスを起こしにくいシステムの構築は教師にとっても有益となる。

### 3. プログラミング課題提出サーバ

次に課題内容がプログラミングに特化した場合のレポート提出サーバについて考察する。まず、構築すべきサーバについて列挙しておく。

- プログラミングにおいては1つの課題（プロジェクト）に対して複数の提出物（ファイル）が生成されることが多い。
- 即ち、プロジェクトごとにファイルを固めておくことが望まれる。（フォルダ等）
- 複数の提出物が同時に発生するため、提出物の部分的欠損が生ずる可能性が高くなる。なぜなら、学生自身レポート提出操作自体は行っているわけだから、完了したつもりになってしまうことも充分あり得るからである。（すなわち、悪気なく未完成提出になることもあり得る）
- 必要に応じてミスは体験させた方がよい。無論、ミスであることは何らかの形で学生に伝える必要がある。
- 単なる提出ミスに関しては学生の自助努力で再提出させるべきである。提出ミス発生のアナウンスが自動化できればなおよい。
- 同じようなミスが多数発生する可能性がある場合は、あらかじめ何らかの工夫をしておくべきである。共同ページによる全体コメント、過去コメントの再利用などが考えられる。スタンプのようなものの利用も有効である。

筆者はいくつかのプログラミング授業を担当しており、授業ごとに提出物も異なるが、主な提出物の種類は以下のとおりである。

表1 主な提出ファイルの種類

ファイルの内容	ファイルの形式
PEN ソースコード	テキストファイル
C++ ソースコード	テキストファイル
仕様書	テキストファイル
実行結果	テキストファイル
データファイル	テキストファイル
C# ソースコード	テキストファイル
画像ファイル	バイナリファイル
音声ファイル (BGM等)	バイナリファイル

大雑把には以下の特徴を満たす必要がある。（あるいは満たすのが望ましい）

- コメントシステムの利用可能条件がテキストファイルであるため、特別な理由がない限りテキストファイルを基本とする。

- 実行ファイル（EXE ファイル）自体は提出させない。必要に応じて教師パソコン側でコンパイルしなおして実行するため、ソースファイルや実行時に必要なデータファイル等は無加工の状態ですべて提出させたい。

なお、初学者向けの学習言語であるPEN[4]のソースコードファイルは拡張子が.penであるが、ファイルの中身は通常のテキストファイルと変わらない。（つまり、C++等と同様である。）

### 4. システムの改造

前述のとおり、本システムは数年前から開発を始めており、すでに実際の授業で使用している。現在に至るまでに数々の改良を重ねたため、授業支援としてカバーできる部分はかなり広範囲かつ多機能になっている。主に以下のような機能がある。

- 授業内容のアナウンス
- ログインシステム
- 自動出欠記録
- レポート提出システム
- レポートコメントシステム
- 提出レポート保管庫
- 面接予約システム
- 電子カルテ
- プログラミング授業間の連動
- 教員間スケジュールの連動

システムは大きく分けて、「ログインシステム」「レポート提出システム」「コメントシステム」「面接予約システム」4つの部分で構成されているが、今回は「レポート提出システム」「コメントシステム」を主に変更した。

本システムを用いて課題を提出する際の提出する際の学生・教師間のレポート提出や評価など典型的な流

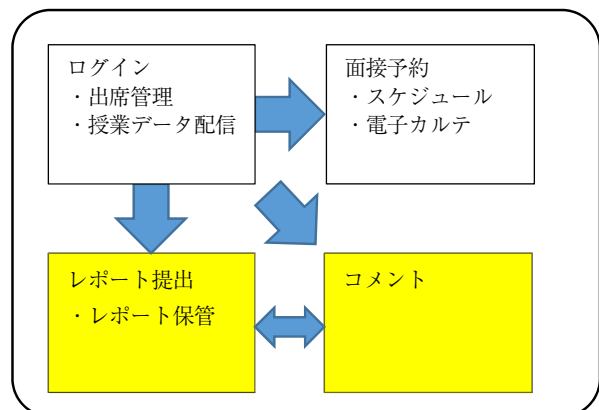


図1 システムの概要

これは図2のとおりである。

与えられた課題に対し、学生はまず仕様書を作成する。前述のように通常課題に関しては仕様書もテキストファイルで作るよう指導している。ワープロ等の使用を禁止しているのはコメントシステムとの連携のためであるが、図表を使うことは多くないので通常課題では特に問題ないと判断している。ただし、最終課題に関しては図表（特に図）を用いた仕様書がわかりやすいため、授業によっては最終課題の仕様書のみ別の方式を用いている。

本システムはもともと PukiWiki[5] を改良して構成されているため、PukiWiki 本来の機能は使うことは簡単である。そこで、各学生に課題テーマを決めてもらうタイプの最終課題などでは Wiki ページとして仕様書を提出してもらおう場合もある（図3）。仕様書のひな型をセットしておいて、仕様書ファイルの提出の代わりに Wiki ページを編集してもらう。この方法の場合は画像データ、色や大きさなどが異なるフォント、表、箇条書きなどを用いることで様々な表現が可能となる。

続いてソースコードの生成、コンパイル、デバッグなどを繰り返し、最後に実行結果ファイルを作成してプロジェクトに必要なすべてのファイルを揃える。

提出の際にはプロジェクトごとにファイルがままとまっているほうが好都合である。そこで旧システムにおいては学生側でプロジェクトごとにフォルダを作ってもらいフォルダごと zip 圧縮したファイルを提出してもらっていた。この方法の場合、課題ごとに一つの zip ファイルが提出されることになるため、zip ファイル内に含まれるファイルの種類をファイル名から自動解析し、提出漏れや不要ファイルの警告を表示するようにしていた。しかしながら、この方法ではいくつか

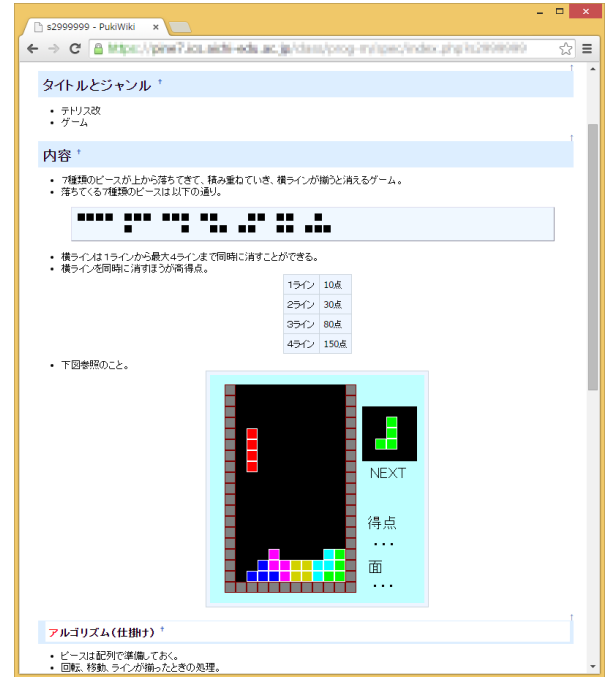


図3 最終課題の仕様書例

の欠点があることが判明した。

- そもそもフォルダの zip 圧縮という操作そのものが面倒である。
- この方法では不必要なファイルまで圧縮する学生が多数発生してしまった。理由はコンパイル時に中間ファイル（obj ファイル等）が生成されるためである。
- 提出不足やコメントの指示に従ってファイルを作り直した場合など、プロジェクトの一部のみ（たとえばソースコードのみ）を再提出した場合、たとえ一つのファイルを提出したいだけでも zip 圧縮が必要になって面倒。
- 再提出において部分的なファイルを zip 圧縮して提出すると、必要なファイルが足りない旨のメッセージが毎回出るため、煩わしい。

特に不要ファイルを消そうとして誤ってソースファイルを消してしまうトラブルも発生したため、不要ファイルの削除に関しては任意に変更せざるを得なくなった。その結果、サーバ上には不要なファイルが大量に送られることになってしまった。

そこで、新システムにおいては、複数ファイルを同時にアップロード可能にし、必要ファイルと不要ファイルをファイル名から解析したのちに必要ファイルのみアップロードする仕組みに変更した。具体的には HTML5 の機能を用いてブラウザ上の特定エリアへのドラッグドロップで実現している。

この改造により、プロジェクトフォルダ内のファイルをすべて選択してドラッグドロップしても中間ファ

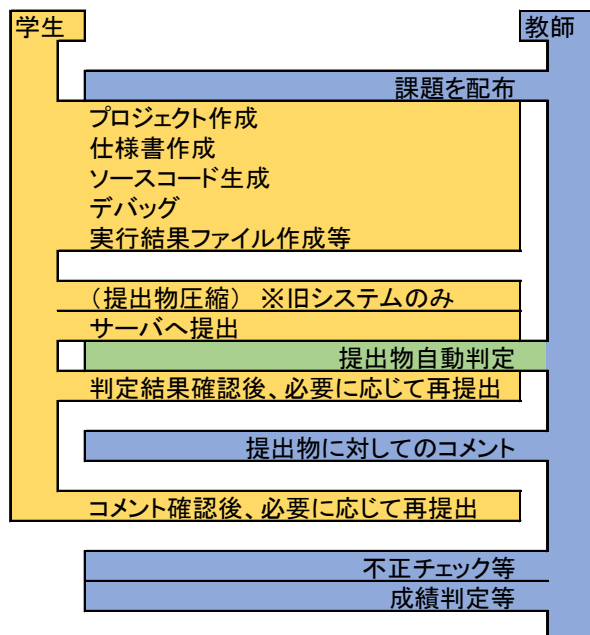


図2 典型的な流れ

イル等がアップロードされることはなくなったので、不要ファイルのアップロードが大幅に軽減した。また、提出不足も即わかるので、学生の自助努力で提出漏れもほとんどなくなった。

ただし、これを実現するにあたりファイル保管庫の仕組みもいくつか変更している。本システムはファイル保管庫の機能を有するため[2]、提出ファイルは一定時間が経過すると上書きできないようにファイルがロックされる仕組みが施されている。旧システムの場合、原則プロジェクトは一つのzipファイルに保存されているためロック済みファイルと同名のファイルはアップロードできないようにするだけで充分だった。(再提出の際には圧縮ファイル名を別名にする必要があるが、それさえ行えば特に問題は発生しない。)

しかしながらプロジェクトの一部のみを再提出可能にする場合、学生サイドでファイル名を変更する仕組みでは利便性が激減する。そこで、ファイルロック時に提出日時から生成した数字(ロックナンバー)をファイル名に付加することで同名ファイルのアップを可能にした。すなわち同名ファイルでもロックナンバーを含めればユニークなファイル名としてすべて保存する仕組みである。

教師はロックされたファイル(新システムではロックナンバーが付加されたファイル)にコメントを書くことができるが、このコメントシステムも大幅に変更した。

- 一つのプロジェクトで複数のファイルが提出されるため、各学生のプロジェクトごとにコンパクトに一覧表示した。ファイルの種類だけを記号で表現し、クリックした場合にはファイルの種類ごとに別ウィンドウで表示するようにし

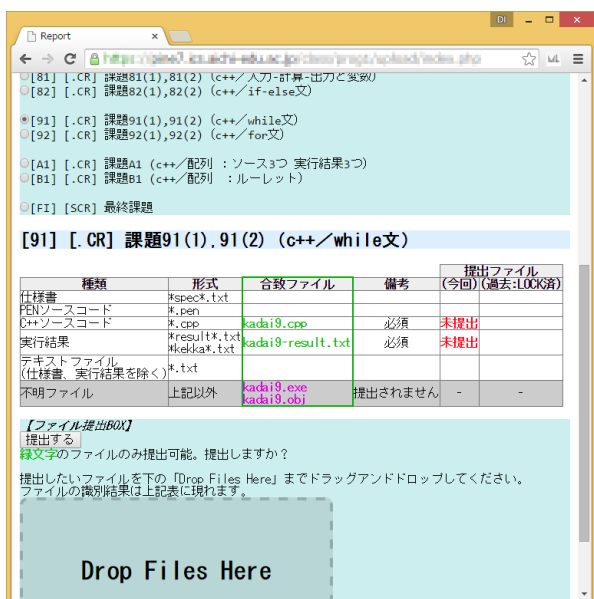


図4 HTML5を用いたファイルのアップロード

た。また、提出日時は色で表現している。(図5)

- コメントに関しては旧システムの埋め込み方式[2]は止めにして、メッセージを好きな場所に重ねられるように変更した(図5)。コメント枠をペタペタと貼り付け、ウィンドウのように大きさを変えたり移動したりできるようにした。実現にはjQuery[6]等の技術が使用されている。また、同じファイルに1つではなく複数のコメントを書くことも可能である。
- なお、学生用ページではコメントの描画を自由にON/OFFできるため、教員は文字が隠れるなど気にせずにコメントを書くことができる。
- よく出てくるコメントもあるため、すでに関心したコメントを一覧表示し、再利用する仕組みを作成した。(図6)
- さらによく使われるコメント等はスタンプで対応できるようにした。(図7)

コメントに関しては、手軽さだけを考慮した場合は印刷物に直接指示を書き込む方が早いという意見が無いわけではない。しかしながら、データのデジタル化は様々な副作用的効果も期待できると考えている。例えば、間違いやすい箇所は同じミスを起こす学生が複数名現れることがよくある。無論、デジタル化の恩恵としては、まず、「コメントの再利用」という分かりやすいものが挙げられるが、それだけではない。一つのファイルで複数のコメントを全く同じように書く羽目になったり、指示の出現パターンによってはある種のデジャブを感じたりするなど、コメントを付けている段階で異常性に気が付く場合がある。すなわち、複数の指示に対するの同時出現率(指示間の関連密度)の高さから異常性を客観的に知ることが可能となる。このときは、類似度チェックを行う。受講生同士のチェックはもちろんのこと、過去の膨大なデータとの

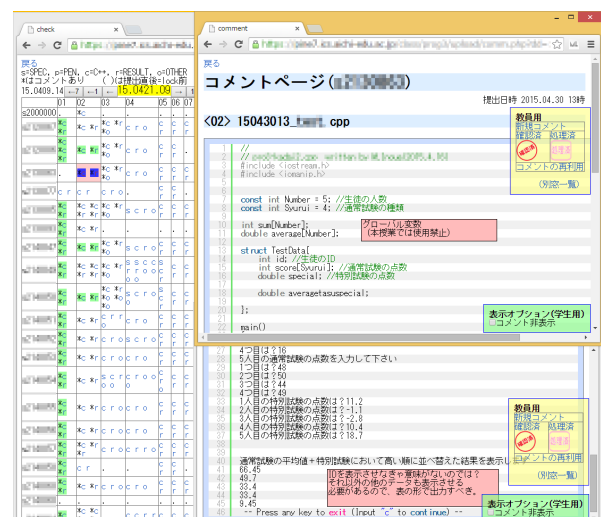


図5 コメントシステム

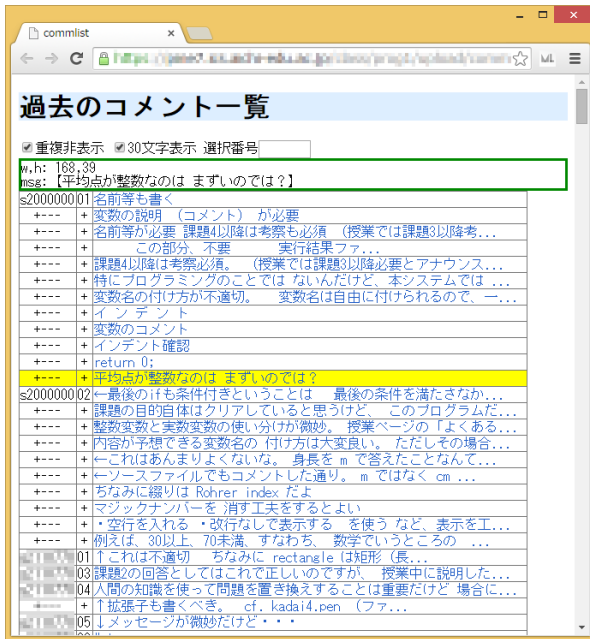


図6 過去コメントの利用

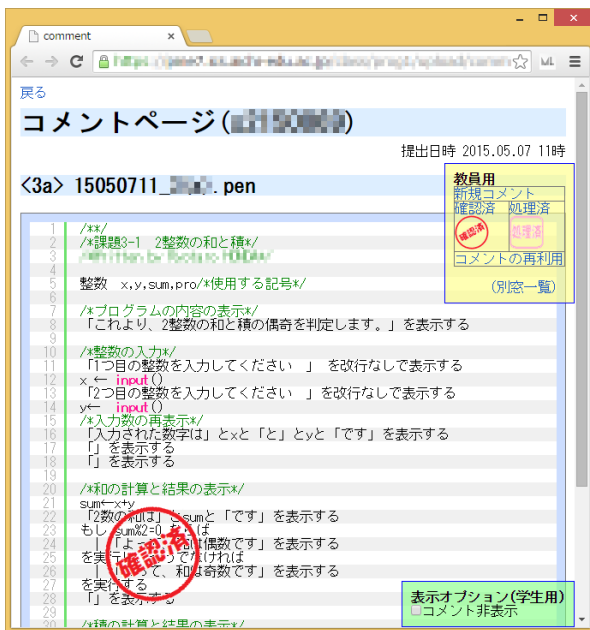


図7 スタンプの利用

## 5. おわりに

本研究ではトラブル軽減を考慮したプログラミング授業支援システムの開発を行った。提出物が複雑になることによって引き起こされるトラブルを先回りして回避するシステムである。その結果、提出時のトラブル軽減などに役立つシステムが構築できた。

本論文では過去に作成したシステムの問題点を指摘することでトラブル軽減についての議論を深めたが、提案システムは従来のバージョンの段階ですですに本学のカリキュラムや授業運営方針に極めて特化した進化をさせてしまっている。例えば、PEN, C++, C#など扱う言語が複数、テキストベースのプログラミングとGUI系のプログラミング、授業時間外に個別面談を行っている、それに向けた面接予約が必要、面接予約の前に課題の完成物の提出が必要（面接枠のキープ禁止）、複数の授業担当者に見せる必要があるため教員間の連携が必要、通常課題と最終課題の関連性、などである[1][2]。プログラミング支援システムに関する研究は多々あるが、このような事情から他の研究との比較は困難であったため、旧バージョンとの比較という形で議論を進めた。

学生、教師、の双方ともに一定以上の高評価は得られたが、さらなる改良点も見つかった。これまでのシステムでは学生への指示、すなわちコメントシステムではテキストファイルのみを扱う形で進化しているが、GUI画面の指示等も有効であるかもしれない。例えば、河田らのGUIプログラミング教育支援システム[7]などは参考になると考えている。また、類似度チェックシステムの向上が挙げられる。また、コメントシステムにおける描画形式の多様化も必要であり、本システムを利用した担当教員からの要望もあった。具体的には提出ファイルのソーティングの順番やコメントシステムにナビゲーション部分の改良などが主要な要望である。これらの改善が今後の課題である。

## 参考文献

- [1] 松永豊, プログラミング演習授業のための面接予約システムの開発, 愛教大研究報告 58 輯 (教育科学編), 2008
- [2] 松永豊, プログラミング演習授業支援システムの開発, 愛教大研究報告 59 輯 (教育科学編) 2009
- [3] 松永豊, Node を用いたプログラミング授業支援システムの開発について, 愛教大研究報告 63 輯 (教育科学編), 2013
- [4] PEN: <http://www.media.osaka-cu.ac.jp/PEN/>
- [5] PukiWiki: <https://pukiwiki.osdn.jp/>
- [6] jQuery: <http://jquery.com>
- [7] 河田進, GUIプログラミング教育支援システム—正しい実行結果と正しくない実行結果の理解支援—, 香川高等専門学校研究紀要6, 2015

(2015年9月24日受理)