

PDF ドキュメント添削ツール GABA と それを用いたオンラインレポート提出・添削環境

掛下 哲郎[†] 本山 翔一[†]

学生が作成したレポートの提出や、教員による添削結果の返却・修正指示をオンラインで行う形式の教育指導は、紙媒体でのレポート提出・添削を用いた方法と比較してターンアラウンド時間が短縮されるため、学生に対する指導機会が増大し、レポート達成度も向上する。本論文では、レポート添削作業の効率化および学習指導の高度化を目的として、PDF ドキュメント添削ツール GABA を開発し、Moodle 課題モジュールと組み合わせることで、レポート提出・添削および返却をオンラインで行えるようにする。複数の学生に同一または類似した課題を出題する場合、学生が作成したレポートにも同種の誤りが繰り返し現れる場合が多い。そこで、添削の際に使用したコメントを GABA に登録して再利用することにより添削時間を短縮する。GABA は Adobe Acrobat 8 のプラグインとして実装されており、PDF 形式に変換できる各種のレポートの添削に広く活用できる。また、GABA を用いることで、コメントのカテゴリ分類、添削位置に応じたカテゴリの自動選択、使用頻度に応じたコメントの並べ替え、コメント数のクロス集計を用いたレポートの評価支援なども可能になる。

An Annotation Tool GABA for PDF Document and an Online Environment for Report Submission and Correction using GABA

TETSURO KAKESHITA[†] SHOICHI MOTOYAMA[†]

Comparing online and paper-based environments for report submission and correction, the former supercedes to the latter, since (1) the turn-around time becomes shorter, (2) teaching opportunity increases, and (3) as a consequence, the student's achievement level becomes higher in the online environment. In this paper, we develop an annotation tool GABA for PDF document and combine it with Moodle assignment module in order to reduce correction time by the instructors and to facilitate instruction to students. In a usual class, the same or similar assignments are given to the students. Then it is often the case that many students make similar mistakes. We thus register common correction comments to GABA and reuse them. GABA is developed as a plug-in of Adobe Acrobat 8 and can be used to put comments to various PDF documents. In addition to reuse comment, it also becomes possible to categorize correction comment, to automatically select appropriate category using mouse location, to sort comments using their frequency and to provide cross tabulation of comments for report evaluation by using GABA.

1. はじめに

大学等の教育機関では、学生に演習課題を出題し、提出されたレポートの添削を通じて学んだ知識を実践できる能力を育成する教育方法がしばしば行われている。角らはレポートの作成・提出および教員によるレポートの添削をオンラインで行うシステムを開発し、ターンアラウンド時間を評価した[1]。その結果、紙媒体でのレポート提出・添削・返却の場合と比較してターンアラウンド時間が短縮され、結果として指導機会の増大およびレポートの達成度向上が認められた。

本論文では、添削対象となる電子ファイルの範囲の拡大、ターンアラウンド時間の短縮および添削を通じ

た学習指導の高度化を目的として、PDF ドキュメント添削ツール GABA を開発し、Moodle 課題モジュールと組み合わせることで、レポート提出・添削および返却をオンラインで行えるようにする。GABA は以下に挙げる特徴を持つ。

- (1) 文献[1]ではレポート作成および添削のために専用のソフトウェアおよびファイル形式を使用しているが、添削対象のファイルを PDF ドキュメントとすることで、様々な形式のレポートに柔軟に対応できる汎用性を確保する。
- (2) 教員がレポートに書き込む添削コメントをシステムに登録することにより、コメントを再利用し、添削時間の短縮を図る。
- (3) 添削コメントに付加した重要度情報、添削者の操作履歴、マウスの位置情報などを活用することで、

*[†] 佐賀大学
Saga University

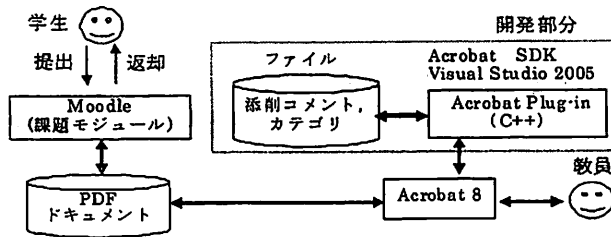


図1 レポート提出・添削環境の全体構成

Figure 1 An Online Environment for Report Submission and Correction

添削の効率化・高度化を図ることができる(未実装)。

Adobe Acrobat 8 Professional (以下, Acrobat 8) の注釈・レビューツールは, PDF ファイルの任意の位置に様々な注釈(ノート注釈, テキスト注釈, ハイライト表示・線・長方形・多角形・楕円・自由曲線等への注釈)をつけることができる。これを活用することで(1)の汎用性は確保される。しかし, それぞれの注釈には, キーボードから直接コメントを記入する必要がある。一度使用したコメントを再利用することはできない。一方, 実際のレポートでは, 多数の学生に対して同一または類似した課題を出題するケースが多いため, 提出されたレポートにも同じような誤りが繰り返し出現する場合が多い。そこで, 注釈で使用した添削コメントをファイルに保存して再利用することで, 添削時間の削減を図る。添削コメントの再利用だけを目的とした場合, 当該コメントをテキストファイル等に保存しておき, 必要に応じて Cut & Paste する方法もあるが, GABA では, 添削コメントに付加した重要度情報や, 添削者の操作履歴を活用することで, 添削作業の効率化や高度化を図ることも可能な設計にしている。

従来の添削システムでは, 例えば CoCoA[2]のように教員・学生の双方が専用のソフトウェアを使用する必要があり, 添削対象も限定されていることが多かった。また, 注釈を含むアノテーションが知識共有に有用なことは[3]等で知られているが, 授業での活用事例は CoNote[4]等のように講義資料等を対象として教員や学生がコメント付けを行うものが多かった。

以下, 2 節では, レポートの作成・提出・添削等を効率化するための仕組みを説明する。本論文では Moodle の課題モジュールと GABA を組み合わせてこれを実現している。3 節では GABA の添削機能およびその操作性を向上させるための工夫について説明する。4 節では GABA の実装方法を説明する。5 節では, GABA の操作時間を計測し, Acrobat 8 単体で添削した場合と比較する。添削コメントを再利用することでタ

イピング量を削減し, コメント数に応じて添削時間を大幅に削減できる。6 節では, 添削作業の効率化や高度化についての展望を示す。

2. GABA と Moodle を用いたレポート提出・添削環境の構築と運用

オンラインでのレポート添削を実施するためには, 学生がレポートを作成・提出するための環境と, 教員が提出されたレポートを効率よく収集・添削・フィードバックするための環境が必要である。本研究では, そのためにオープンソース LMS として普及している Moodle[7,8]を用いる。レポート提出・添削環境の全体構成を図1に示す。

Moodle には課題モジュールが組み込まれており, 教員は, 授業ページの任意の週に課題を追加できる。また, 課題提出ページでは, 以下に示す各種のパラメータを指定できる。

- 課題名
- 学生に対する指示(具体的な課題および各種の注意事項など)
- 評点(満点の点数を指定する)
- レポートの提出期限(開始日時, 終了日時)
- 課題のタイプ(「ファイルの高度なアップロード」を指定すると, 複数ファイルの提出やレスポンスファイルの返却が可能になる)
- グループモード(学生をグループ分けして, グループ別に課題を与えることができる)

学生は, 指示されたページからレポートファイルをアップロードする。Moodle にはユーザー登録機能があるため, 学生は授業開講時に講義 HP に登録する必要がある。また, 学生はユーザー ID とパスワードを用いてログイン講義 HP にログインするため, 個人単位で管理できる。

学生が提出したファイルは, Moodle の教員用ページからダウンロードできる(一括ダウンロードも可能だ

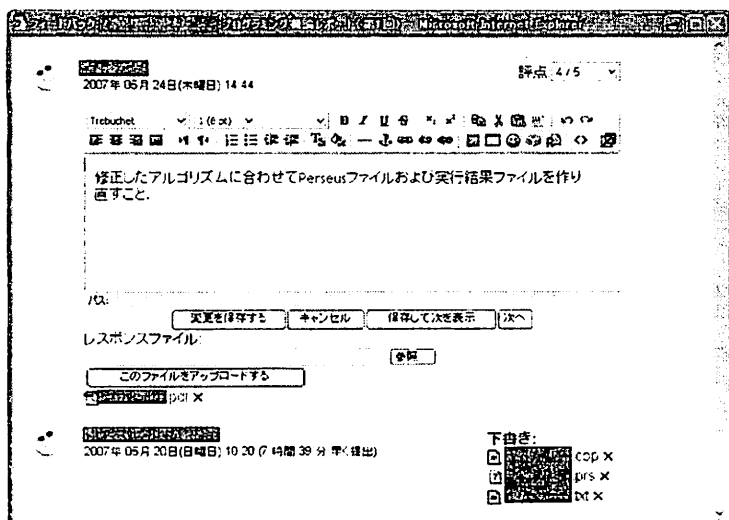


図2 教員用画面：成績通知と添削結果の返却
Figure 2 Sending Score and Feedback File to Students

が、提出期限を過ぎたレポートは個別にダウンロードする必要がある)。このページでは、学籍番号、評価、レポート提出日時等の条件を用いてデータをソートできるため、教員はレポート未提出者や締め切りの遵守状況を容易に把握できる。なお、Moodleは学籍番号を直接には管理していないため、各学生には個人ページで姓の欄に学籍番号を、名の欄に氏名をそれぞれ記入するように指示している。

学生がAcrobat 8等のPDF作成環境を所有している場合には、学生にPDFファイルを作成させ、それを直接提出させることもできる。しかし、Acrobat 8は有償ソフトウェアなので、学生にはcppファイル(C++プログラム)やdocファイル(Word文書)等を提出させておき、ダウンロード後に教員側でPDFファイルに変換する運用も可能である。当然ながら、PDF変換のための手間が教員側で必要になるが、所要時間を計測したところ、1ファイル当たり10秒程度でファイル変換できるため、後述の添削時間と比較すると無視できる。また、必要に応じて、TA等に変換作業を依頼することもできる。

このようにしてPDF化されたファイルを対象としてGABAを用いた添削を行う。添削済みのファイルはMoodleの課題モジュールを使用し、レスポンスファイルとして学生に返却する(図2を参照のこと。なお、個人情報保護の観点から、学生の氏名や学籍番号が特定される恐れのある情報をマスキングしている)。これにより、学生と教員の双方に添削結果が残るため、指

示内容を確認するのも容易である。また、認証評価やJABEE等のア kredィテーション審査に備えて根拠資料を蓄積する作業も電子化されるため、紙媒体での資料保存(学生にレポートを返却する場合、コピー等の手間がかかる)と比較してはるかに容易になる。課題モジュールを用いることで、学生にレポートの評価を個別に通知することもできる。入力した評価は学期末の成績評価の際にも活用できる。

なお、図2の画面では、教員がレポート全体に対するコメントを入力して学生にフィードバックできる。しかし、GABAを用いた添削のように、レポート内容に関する詳細な指摘を行う場合、指摘箇所を文章のみで具体的に説明するのは難しい。

学生は、講義HPの課題ページをアクセスして添削結果を受け取り、その指示に従ってレポートを修正し、再提出する。フィードバックファイルと同時にレポート評価も個別に通知される。

以上のサイクルを繰り返すことで、レポートの作成・提出・添削・フィードバックはもとより、資料の保存や成績評価も効率よく実現できる。

3. GABAの添削機能

GABAはAcrobat 8のプラグインとして実装した。Acrobat 8のプラグイン用フォルダにGABAの実行形式ファイルをコピーしてAcrobat 8を起動すると、GABAがAcrobat 8に組み込まれる。

コメント文字列	カテゴリ番号
同様の指図に該当する箇所が複数ある場合でも、原則として1箇所だけにコメントを付けています。コメントを読んで、同様の指図が該当する箇所は各自で修正すること。アルゴリズム等を修正した場合には、それに合わせてPerseusファイルおよび実行結果ファイルを作り直すこと。	1
プログラムが完成していない。	1
Perseusで作成したファイルを提出すること。	1
プログラムの実行結果を提出すること。(※ プログラムと一致しない実行結果を提出した場合には、悪質な行為として減点します)	1
アルゴリズムが誤っている。	3
アルゴリズムの具体化が不足しており、手作業ではトレースできない。場合分けや繰り返しを行なう場合には、アルゴリズムを明記すること。「以下の処理」の具体的内容を記述すること。このままでは「アルゴリズム作成のガイドライン」に従っていない。	3
オーバーコメント(〇++コードから自明に分かるコメント)は不可。アルゴリズム等では、コードの意図を表現すること。	3
アルゴリズムの書式が誤っている。アルゴリズム作成のガイドラインに従うこと。	3
アルゴリズムとプログラムが対応していない。	3

図3 添削コメント・ファイル (一部)

Figure 3 Correction Comment File

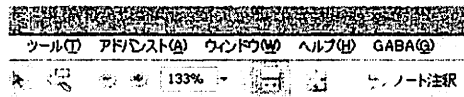


図4 GABAのメニューバー

Figure 4 GABA Menu Bar

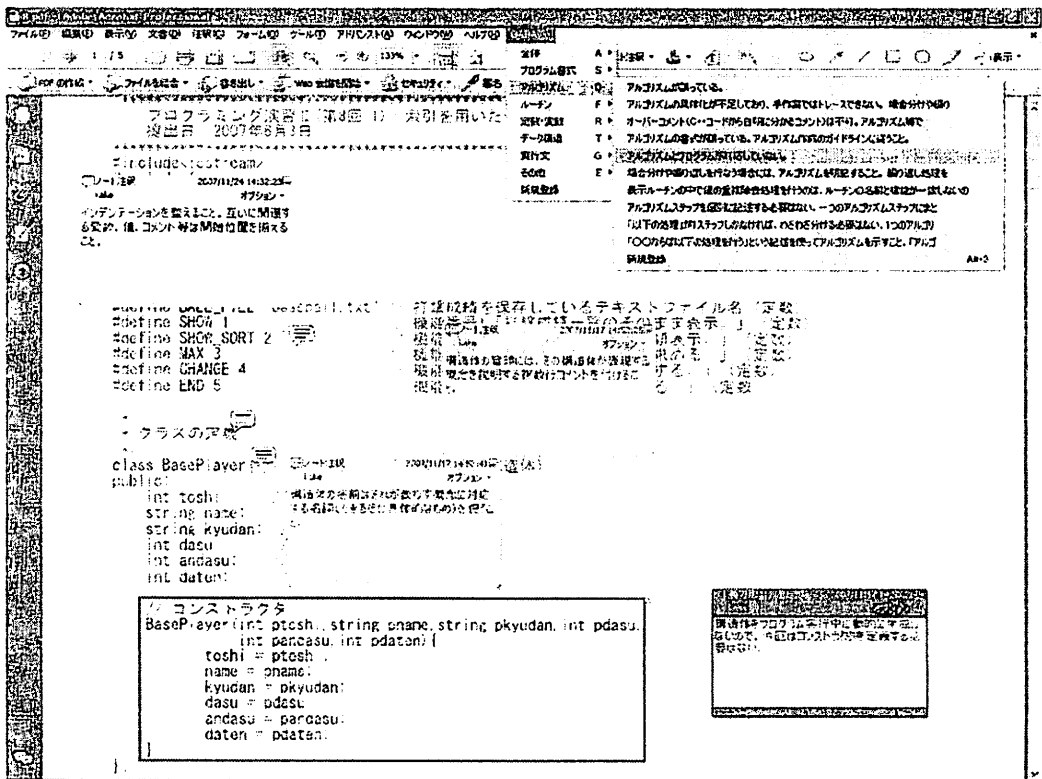


図5 カテゴリと添削コメント

Figure 5 Categories and Correction Comments

GABAを組み込んだ状態でPDFファイルを開くと、当該PDFファイルが置かれているフォルダに配置された添削コメント・ファイル（添削コメント本文とカテゴリ番号から構成されるCSV形式ファイル、一部を図3に示す）およびカテゴリ・ファイル（カテゴリ名とカテゴリ番号から構成されるCSV形式ファイル）から添削コメントおよび添削コメント・カテゴリが読み込まれ、GABAのメニューに登録される。学生が提出したレポートファイルを1つのフォルダにまとめておくことで、異なる学生のレポートに対して同一のカテゴリおよび添削コメントを使用できる。逆に、異なる授業科目や課題に対しては、別のフォルダを用いて分離することで、科目や課題に応じたカテゴリおよび添削コメントの変更も容易に行える。

メニューバーの「GABA」項目（図4）を選択すると、添削コメントのカテゴリが表示される。利用者がカテゴリを選択すると、そのカテゴリに属する添削コメントが表示される。利用者が指定した添削コメントは、マウスで選択されている注釈の末尾に追加される（図5）。1つの注釈には、必要に応じていくつでも添削コメントを追加できる。図5にはAcrobat 8のノート注釈に添削コメントを追加した例を主に示しているが、他の種類の注釈（テキスト注釈、長方形への注釈等）に添削コメントを追加することもできる。

そのため、利用者はまずAcrobat 8の機能を用いて空の注釈を追加する。次に、GABAを用いてその注釈に添削コメントを挿入する。さらにGABAでは、必要に応じてカテゴリや添削コメントの追加・編集も行える^{*}。代表的な添削コメントをあらかじめ添削コメント・ファイルに登録しておくことで、添削の採点基準を具体的に示せるため、複数人での添削の際にも、整合性を保ちやすくなる。これは、多数のレポートをTA等が分担して添削する際などに有効に機能する。

添削コメント・ファイルは、Excel等を使用して採点者が作成・編集できる。なお、採点の客観性を高めるためには、過去の類似レポートに対するコメントを参考に作成する方法や、学生に公開している成績評価基準ないしは各種ガイドラインに基づいて作成する方法を採用することが望まれる。例えば、図3に示す添削コメント・ファイルは、著者が作成・公開しているC++プログラミングのガイドライン[5]に基づいて作成したものである。

また、GABAの操作性を向上させるために、各カテゴリおよび各添削コメントにはキーボードショートカ

ットを割り当てた。利用者が空の注釈を追加する際には、マウスを用いて注釈の位置を指定する。一方、GABAメニューを選択する操作や、選択した注釈に添削コメントを追加する操作は、マウス操作の他に、キーボードショートカットを用いて行うこともできる。これにより、マウス操作（右利きの利用者の場合、主に右手で行う）とキーボード操作（主に左手で行う）の役割を分担し、添削時間の短縮を図る。また、キーボード操作を左手で行いやすくするために、ショートカットキーは、左手のホームポジションから操作可能な範囲にある中段キー（A, S, D, F, G）、上段キー（Q, W, E, R, T）、下段キー（Z, X, C, V, B）および数字キー（1, ..., 5）を用いる。

4. GABAの開発

GABAの開発はAcrobat SDK 8[6]およびVisual Studio 2005を用いて行った。開発言語はC++である。Acrobat SDKには、メニュー作成、イベント処理、PDFファイル操作等を実装するための各種APIが用意されている。

添削の際に使用するコメント、コメントのカテゴリ名、カテゴリ番号はCSVファイルに保存されており、GABAの起動時に読み込む。GABA起動時には、この他にメニューの初期設定が行われる。

GABAは、Acrobat 8の注釈・レビューツールを用いて作成された注釈を利用者が選択した時に、メニューで指定された文字列をその注釈に追加する。このために、注釈が選択されているか否かを判定する関数、および指定された文字列を選択されている注釈に追加する関数を実装した。また、コメント文字列を新規登録するため、指定されたコメントが保持する文字列をコメント・ファイルに追加する関数を実装した。

上記の関数は、GABAの個別メニュー項目（図5）に設定されたイベントハンドラによって呼び出される。コメント文字列が新規登録されると、それに合わせてメニュー項目が更新される。

5. 添削機能の操作性評価

GABAのユーザインタフェースを設計する際には、GUI部品やキーボードの操作時間[9]を考慮して添削者の手間をできる限り減らし、操作性の良いものを開発することを目指した。本節では、GABAを組み込んだ場合とそうでない場合についてAcrobat 8の操作性を計測し、両者を比較する。また、GABAの改善に向けて被験者から収集したコメントを紹介する。

5.1 Acrobat 8の操作性評価

Acrobat 8の注釈・レビューツールを用いて添削を行

^{*} カテゴリやコメントの削除機能や編集機能などはレポートの添削作業中に使用するケースは少ないため、別メニューでの実装を予定している。

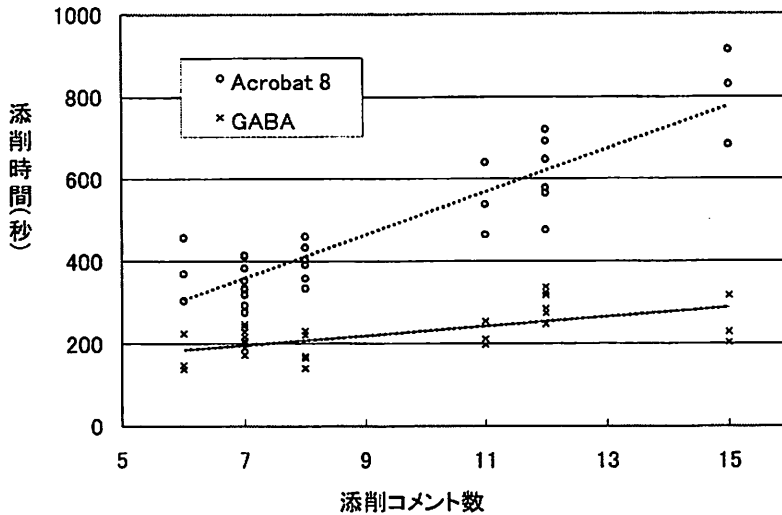


図6 Acrobat 8 と GABA を用いたコメント追加に伴う操作性評価

Figure 6 Operation Time of Acrobat 8 and GABA for Comment Addition

う場面を想定し、同一課題について学生から提出された 10 通のレポートを添削し、その所要時間をレポート毎に計測した。各レポートは 350 行程度の C++ プログラムであり、PDF 化したものは 5 ページ程度になる。各レポートの完成度は様々なので、添削コメントの数も 6~15 個の範囲で分布している。また、各コメント文字列の長さは 4~125 文字の範囲にある。被験者は 9 名（教職員 6 名および TA 経験を持つ大学院生 3 名）である。

各レポートに対する具体的な添削内容（コメントを付けるべき位置と内容）はあらかじめ指示した。これによって、純粋な編集時間のみを計測した。実際のレポート添削においては、添削対象部分の特定やコメントの選択を行うための時間も必要になるが、本実験では、ツールの操作性のみを評価する観点から、これらの所要時間は評価の対象に含めていない。

評価実験の結果を図 6 に示す。実験結果データの線形近似を行うと、コメント数 x に対して添削時間 $y = 52.243x - 6.2896$ となる。添削コメントが 1 個増えるに従って、添削時間は 52.2 秒の割合で増加する。

Acrobat 8 の注釈・レビューツールは注釈のコピー・ペースト機能も備えているが、各レポートには同一コメントは現れないため[†]、同一ファイル内での注釈の再

利用はほとんど発生しない。異なるファイル間での注釈の再利用は可能だが、注釈を探すために画面の切り替えやスクロール等が必要になり、そのために時間を要している。

5.2 GABA の操作性評価

5.1 節の評価実験と同一のレポート集合を対象として GABA を用いた添削を行い、その所要時間をレポート毎に計測した。被験者は 5.1 節の評価実験と同一である。

同一レポートを同一被験者が連続して添削することから、被験者が添削内容を覚えることにより、2 回目の添削作業の時間が短縮される可能性が考えられる。これによる影響を排除するため、レポート毎に添削順序（Acrobat 8 と GABA のいずれを先に行うか）を被験者に指示し、2 つの場合が均等に出現するように工夫した。

評価実験の結果を図 6 に示す。実験結果の線形近似を行うと、コメント数 x に対して添削時間 $y = 11.466x + 117.33$ となる。コメント数の増加に伴う添削時間の増加の割合は 11.5 秒であり、Acrobat 8 だけを用いた場合の 21.9% に留まる。被験者によるばらつきが少ないのは、コメントの再利用により、タイピング能力の差が出にくいためである。GABA を用いても、PDF ファイル内でコメントを追加すべき場所を探す手間および GABA に登録されていないコメントをタイプする手間は必要だが、十分な時間短縮効果がある。

ただし、Acrobat 8 の操作性評価においては、全ての

[†] レポート冒頭で、「同様の指摘に該当する箇所が複数ある場合でも、原則として 1 箇所だけにコメントを付けています。コメントを読んで、同様の指摘が該当する箇所は各自で修正すること。」との指示を行った。これにより、単一レポート中に同一の添削コメントが複数回出現しないようにしている。

コメントをキーボードから入力させている。より現実的な添削では、テキストファイル等で添削コメントを保持しておき、必要に応じて Cut & Paste する場合がありますと考えられる。その場合、Acrobat 8 と GABA の操作時間の差はより小さくなると予想される。

5.3 被験者のコメント

評価実験を行った被験者から出されたコメントを以下に示す。このうち(1)～(3)では GABA の有用性が確認できた。(4)～(7)への対応については 6 節で述べる。

- (1) コメント入力が簡素化されていて便利。
- (2) コメントが予め登録されているので、ミスが減少する。
- (3) GABA を用いることで添削作業は大幅に効率化される。
- (4) 添削コメントを追加したい箇所にマウスカーソルを持っていき、その場で右クリックすると GABA メニューが展開されて、コメントを選択できるようになっていると、マウス操作だけでコメントが追加できるので望ましい。
- (5) 添削コメントのソート機能（利用頻度順）があった方が望ましい。
- (6) コメント文字列が配置されているカテゴリを理解するまではまごついた。
- (7) コメントしたい箇所の一部をマウスで選択した時に、コメント候補を自動的に絞り込んで表示するように改善すると操作性が向上する。

6. GABA を用いた添削作業の効率化および高度化に向けた展望

5 節で述べた添削コメントの再利用は、テキストファイル等で添削コメントを保持しておき、必要に応じて Cut & Paste することでも実現できる。その意味で、添削コメントの再利用による操作時間の短縮は GABA の本質的な長所とは言えない可能性がある。

しかし、GABA は Acrobat 8 のプラグインとして開発されているため、以下に列挙するような、単なる添削コメントの再利用よりも高度な機能を実現できる。これらの機能を活用することで、5.3 節で紹介した被験者のコメントも踏まえた上で、添削作業の効率化および高度化を図ることができる。

- (1) PDF ファイルに追加された注釈を右クリックして添削コメントメニューを表示する。これにより、注釈を追加するための手間を減らすことができる。
- (2) マウスカーソルの位置情報を用いて、その付近にある添削対象（例：#define, if, while, for, 変数, ルーチン, コメントなど）に応じて、添削コメントメニ

ューのカテゴリを自動的に切り替える。

- (3) レポート課題毎に添削コメントの使用頻度を集計する。レポートの添削カテゴリ毎に分類したコメントの並び順を、当該カテゴリ内で使用頻度の多い順に自動的に並べ替える。これによって、しばしば使われる添削コメントを、最小限の手間で入力できる。
- (4) コメントに重要度を付加する。重要度およびカテゴリを用いて、当該レポートを採点する際に用いた添削コメント数のクロス集計を生成する。生成したクロス集計は、レポートを採点する際の基本情報として、レポートに書き込む。
- (5) 添削済みの PDF ファイルに書き込まれた添削コメントをカテゴリや重要度毎に集計することで個別の学生やクラス全体の理解度を把握し、授業内容やレポート指示の改善に活用する。

現在、(1)～(5)の機能の実現に向けて Acrobat SDK の機能調査を行っている。(2)～(5)の機能の実現は可能な見込みである。しかし、マウスの右クリックに対応したイベントハンドラの定義や、クリック位置におけるメニューの表示は、Acrobat 8 のプラグインには許されていない。そのため、(1)の実装は困難と思われる。

7. おわりに

GABA を用いることで、PDF に変換できる広範囲のレポートに対して効率的にコメントを付けることができる。教員が多くの学生に対して丁寧なコメントを返そうとすると、タイピング量が必然的に多くなるため、GABA の有用性が増すことが期待される。

60～100 通のレポートを添削する場合、1 通当たりの添削時間を 10 分程度としても、教員は 10～17 時間程度の時間を添削作業に費やす必要がある。図 6 にも示したが、GABA を用いることで 1 通当たりの添削時間を高々 5 分以下に抑えることができる。その分だけ、教員は学生に対する指導時間をより多く確保できることになり、教育改善を行う機会も生まれてくる。

今後の課題としては、6 節に挙げたような添削の高度化・効率化に向けた GABA の改善およびその評価が挙げられる。

参考文献

- 1) 角 世元, 李 曉永, 出口 博章, 太田 剛, 酒井 三四郎, “オンラインレポート添削支援システムにおけるターンアラウンド時間の分析”, 情報処理学会論文誌, Vol. 48, No. 8, pp. 2781-2790 (2007).
- 2) 矢野 米雄, 緒方 広明, 榊原 理恵, 脇田 里子, “日本語作文教育のためのネットワーク型添削支援

- システム CoCoA の構築”, 教育システム情報学会誌, Vol. 14, No. 3, pp. 21-28 (1997).
- 3) 伊藤 禎宜, 角 康之, 間瀬 健二, 國藤 進,
“SmartCourier: アノテーションを介した適応的情報共有環境”, 人工知能学会論文誌, Vol. 17, No. 3, pp. 301-312 (2002).
 - 4) Jim Davis, Dan Huttonlocker, “CoNote system overview” (1995), <http://www.cs.cornell.edu/home/dph/annotation/annotations.html>
 - 5) 掛下 哲郎, C++プログラミングのガイドライン (2003), <http://www.cs.is.saga-u.ac.jp/syllabus/GuideLine/Cguide.html>
 - 6) Adobe Acrobat 8.1 SDK (2006), <http://www.adobe.com/devnet/acrobat/>
 - 7) Moodle ホームページ, <http://moodle.org/>
 - 8) Jason Cole, Helen Foster, “Using Moodle, Second Edition: Teaching with the Popular Open Source Course Management System”, O'Reilly (2007).
 - 9) 船戸 康徳, 宮浦 慎司, 坂本 忠明, 今宮 敦美,
“ボタン操作の評価実験 (インタフェース評価とデザイン技法, および一般)”, 情報処理学会研究報告 1996-HI-066, Vol. 1996, No. 48, pp. 35-40 (1996).

著者紹介

掛下 哲郎 (正会員)

佐賀大学 理工学部 知能情報システム学科 准教授。
ソフトウェア工学およびデータベースを専門とする。
JABEE によるアクレディテーション活動の他,
情報処理学会 高度 IT 人材育成フォーラム等で活動している。
本会の他, 電子情報通信学会, ACM, IEEE-CS 等各会員。

本山 翔一 (非会員)

佐賀大学大学院 工学系研究科 知能情報システム学
専攻 (博士前期課程) 修了。現 NEC。大学院時代は
オンラインレポート添削システムに関する研究・開発に従事。