Vol.2013-GN-86 No.17
Vol.2013-CDS-6 No.17
2013/1/16

講義進捗状況の可視化による プログラミング講義リフレクション支援の提案

堀口 悟史[1,a)]　井垣 宏[2,b)]　井上 亮文[3,c)]　星 徹[3,d)]　岡田 謙一[4,e)]

概要：本研究では，プログラミング講義中の受講生の状態を講義後に詳しく見直すことで，講義各回や全体の改善支援を目的とする．この目的を達成するため，著者らが開発したシステムで記録した受講生の講義時間中の状態ログを，講義後にさまざまな観点から見直すことができるリフレクション支援ビューを提供する．提供されるビューはウェブ上で動作するブラウザベースのシステムである．本稿では，実際の講義で取得したログからどのような改善点が見つかるかを議論する．評価実験として，受講生 12 名によるプログラミング講義において講義時間中の状態ログを収集し，分析を行った．結果として講義資料の内容が十分であるか，遅れている学生へのフォローが効果があったのか等の議論をすることができた．

キーワード：教育支援，プログラミング教育，LMS

A Reflection Support System for Programming Education using Visualization of Student's Behavior

Satoshi Horiguchi[1,a)]　Hiroshi Igaki[2,b)]　Akifumi Inoue[3,c)]　Tohru Hoshi[3,d)]　Kenichi Okada[4,e)]

1. はじめに

情報系大学においてプログラミング講義は必修科目である．優秀なソフトウェア開発者の育成のため，座学で学習した内容についての課題を，受講生が実際にコーディングする形式の講義が多くの教育機関で行われている [1]．

プログラミング講義では，講義中に講師が受講生に課題を与える．受講生はその日の学習内容を振り返りつつコーディングを行い，必要に応じて講師に質問をしつつ，与えられた課題を提出する．このように各受講生が個別に課題に取り組むため，受講生間のスキル差によって進捗状況に大きな差が生まれることが多い．理解度の不足により進捗が遅れる受講生の多くは積極的に挙手や質問等をしないため発見が難しく，講師が気がついた時には遅れを取り戻すことが難しい状況なこともしばしば発生する。

これまでの研究の多くは，講義時間中の各受講生の学習状況をシステム上で把握し，講師や TA などのスタッフが即座かつ適切に対応することを支援していた [2], [3]．しかし，講義は各回が完全に独立したものではない．前回の内容を踏まえた上で発展した内容を理解するといった知識の積み重ねが必要である．この観点から言えば，1 回の講義時間内だけでなく講義後にも見直しを行い，講義計画全体を改善していくことも重要である．

本研究では，プログラミング講義中の受講生の状態を講義後に詳しく見直すことで，講義各回や全体の改善支援を

[1] 慶應義塾大学大学院 理工学研究科
Graduate School of Science and Technology, Keio University
[2] 大阪大学大学院 情報科学研究科
Graduate School of Information Science and Technology, Osaka University
[3] 東京工科大学 コンピュータサイエンス学部
School of Computer Science, Tokyo University of Technology
[4] 慶應義塾大学 理工学部
Faculty Science and Technology, Keio University
a) horiguchi@mos.ics.keio.ac.jp
b) igaki@ist.osaka-u.ac.jp
c) akifumi@stf.teu.ac.jp
d) hoshi@stf.teu.ac.jp
e) okada@mos.ics.keio.ac.jp

目的とする．この目的を達成するため，著者らが開発したシステムで記録した受講生の講義時間中の状態ログを，講義後にさまざまな観点から見直すことができるビューを提供する．本稿では，実際の講義で取得したログからどのような改善点が見つかるかを議論する．

本論文の構成を以下に示す．2 章では関連研究，3 章では提案するプログラミング講義のリフレクション支援手法，4 章では評価実験，5 章では結果および考察，6 章をまとめとする．

2. 関連研究

2.1 講義時間中の支援に関する研究

宮地らはアセンブラプログラミング演習支援システム CAPES を提案している [2]．CAPES は受講生が提出した答案プログラムを自動的に解析してその正誤を判定し，正解者には次の問題を，間違った者には優しい問題を出すといったように，受講者のペースに合わせた出題が可能である．

藤原らは学習者の課題作成状況から行き詰まりを検出する手法を提案している [3]．この手法では学習者のコンパイル回数，実行回数，コンパイルエラー数，行数等を 1 分おきに計測する．その時間変化と講師が事前に設定した変化のパターンとが一致した場合に，システムはその受講生が行き詰まっていると判断する．

これら研究は，講義時間中の受講生の行動から状況を判断し，講義各回の進捗改善に有用である．一方で，特に大学などの教育期間では，教員が授業内容・方法を改善し向上させるための素子的な取り組みであるファカルティ・ディベロップメント（以降，FD）が重要視されている [4]．この観点からすると，講義時間中や講義各回といったミクロの視点からの改善だけでなく，全 n 回の講義計画全体からみて適切であるかといったマクロの視点からの支援も求められる．

2.2 講義後の支援に関する研究

講義内容および講師の質向上を目指した改善はリフレクション [5] と呼ばれる．一般的なリフレクション方法の 1 つでは講義を VTR 撮影し，その様子を後から複数人の講師とともに視聴する．各講師は，声の大きさや問題を抱えている受講生などを確認し，専用のシートに記入していく．最後に，記入したシートをもとに互いの解釈を照らしあわせて議論する．

この方法は効果が大きい反面，人的・時間的にコストがかかることから，様々な支援システムが研究されている．

三石らは授業計画と実施結果との差異に注目し，これらをふり返るべきポイントの候補として授業の全体像とともに提示する手法を提案している [6], [7]．この手法では，スライドの提示や板書の説明に要した時間や，講師が事前に作成した流れと異なる遷移をした箇所を，スライドや板書のサムネイル一覧にわかりやすく表示できる．結果として，リフレクションによる具体的な改善案の創出や次回講義への反映に一定の効果があることを確認している．

この研究が対象とするスライド提示型・黒板型の講義は，現在実施されている講義の大半を占め，適用可能性が高い．また，主に講師の行動をふり返りのキーとしている．一方で，プログラミングのような技術科目では受講生が手を動かす演習的要素が必須である．この種の講義では，先述の研究 [2], [3] にあるコンパイルの失敗や講義資料の閲覧動向といった受講生の行動が振り返りのキーとなる部分が大きい．

3. プログラミング講義のリフレクション支援手法

2 章の内容を踏まえた以下の 3 つが，我々が提案するリフレクション支援の前提条件である．

- プログラミングという受講生の行動が主体の科目を対象とする
- 人的・時間的コストがかからない. 講師が自分で見直せる
- 講義各回だけでなく, 前後・全体を横断して振り返る支援

3.1 キーアイデア

プログラミング講義において，座学と演習時の受講生の学習状況を取得し，そのログを講義後に様々な観点から表示する．表示したデータが講義リフレクション支援につながるか確認する．これにより，座学時における受講生の成長にあわせた講義資料の閲覧方法の変化や，演習時のコーディングにおける授業回ごとのエラーの回数の偏り理由を特定できる可能性がある．以降では，我々が開発したシステムである，座学における講義資料の閲覧状況の収集，演習時のコーディング過程の収集について詳述する．

3.2 講義資料の閲覧状況把握システム

我々はプログラミング講義の座学にて利用される HTML 形式で作成された講義資料に対し講義進捗管理をするシステムを開発した [8]．講師が講義資料にアクセス解析用のタグを埋め込むことで，受講生が講義資料をどのように閲覧しているかが分かる．システムは，閲覧ログ収集サーバおよびデータベース，閲覧ログに基づくずれ検出および遅れ理由分析を行うためのシートマップから構成されている．これにより，授業進度と比較して遅れている受講生を検出できる．取得できるログは以下のとおりである．

- ユーザ ID
- 時刻情報
- ユーザが閲覧している講義資料のアドレス

Vol.2013-GN-86 No.17
Vol.2013-CDS-6 No.17
2013/1/16

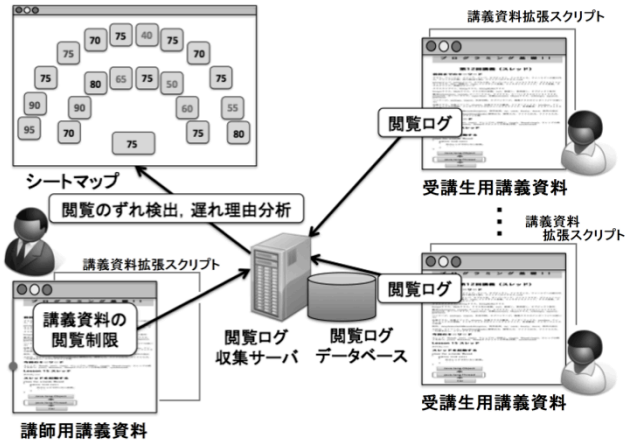

図 1 講義進捗管理支援システムのアーキテクチャ

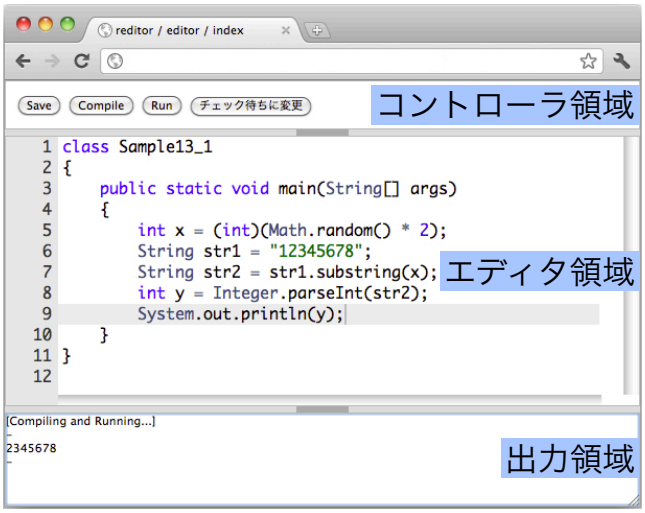

図 2 オンラインエディタによる Java プログラムのコーディング

- ログを記録した時点での講義資料の閲覧範囲

3.3 コーディング状況の把握システム

つぎに我々は演習時における受講生のコーディング状況の把握システムを開発した [9]．受講生はオンラインエディタ（図 2）にブラウザからアクセスし，コーディングを行う．システムはオンラインエディタを通じて，受講生一人ひとりのコーディング課程を記録する．オンラインエディタはコントローラ領域とエディタ領域，出力領域から構成されている．受講生はエディタ領域にソースコードを記述し，Save(保存)，Compile(コンパイル)，Run(実行)，チェック待ちに変更（課題提出）といった操作をコントローラ領域で選択する．受講生が Compile あるいは Run を選択すると，出力領域にコンパイル結果あるいは実行結果が表示される．このオンラインエディタをこれにより，問題を抱えた支援すべき受講生をリアルタイムに検出できるので，優先して支援すべき受講生へ対応できる．このシステムにより取得できるログは以下のとおりである．

- プログラミング演習中にエディタ領域に入力したソースコードの総行数
- 課題ごとのコーディング時間
- 単位時間あたりのエディタ操作数
- 課題ごとのエラー継続時間

3.4 プログラミング講義のリフレクション支援ビュー

3.2 節および 3.3 節のシステムで取得したログを様々な観点から比較できるブラウザベースのシステムを開発した．図 3 にその外観を示す．本システムは，横軸方向に時刻を，縦軸方向に資料閲覧箇所やコンパイルなどユーザ行動を定量化した値を表示する．この際，以下の項目を自由に切り替えることができる．

(1) ユーザ
(2) ユーザの行動
(3) 基準時刻
(4) 表示間隔

項目 1, 2 を切り替えて表示することで，それぞれのユーザがいつ，何を，どの程度していたのかの時間変化を直感的に把握できる．また，項目 3,4 を切り替えて表示することで，講義 1 回分から，前後の回，シラバス全体を通した分析が可能となる．

図 3 は講義 4 回分の時間変化を表示したものであるが，画面下部の時間軸をマウスで範囲指定することで，分析区間の拡大縮小が可能である．図 4 は，図 3 のうち，講義 1 回分（90 分）を拡大表示したものである．図 4 より，当該受講生は講義進行とともに資料を下方向に読み進めていくが，13:48 頃に資料の手戻りをしていることがわかる．

本システムを用いれば，ユーザ間での比較や，講義単体・全体を通した時間変化を容易に把握することができる．

4. 評価実験

プログラミング講義における受講生の学習状況を記録したログを授業後に解析することで，リフレクション支援につながるか確認するための実験を行った．学部 3 年生向けである Java プログラミングについての復習を行う集中講義（全 6 回）を対象に，我々が開発したプログラミング講義における状況把握システムを利用し講義資料の閲覧状況とコーディング状況を収集した．被験者は 12 名，講義時間は 1 コマ（90 分）である．講義では，概ね講義開始 40 分を座学．残り 50 分を演習として行った．講義終了後にリフレクション支援ツールにて受講生の講義時の行動分析を行い，そこから得られた知見について考察する．

5. 結果および考察

4 章において述べた Java プログラミング講義を対象に，リフレクション支援ツールで受講生の講義時の行動分析を行った．以降に特徴的な事例を挙げ，考察を行う．

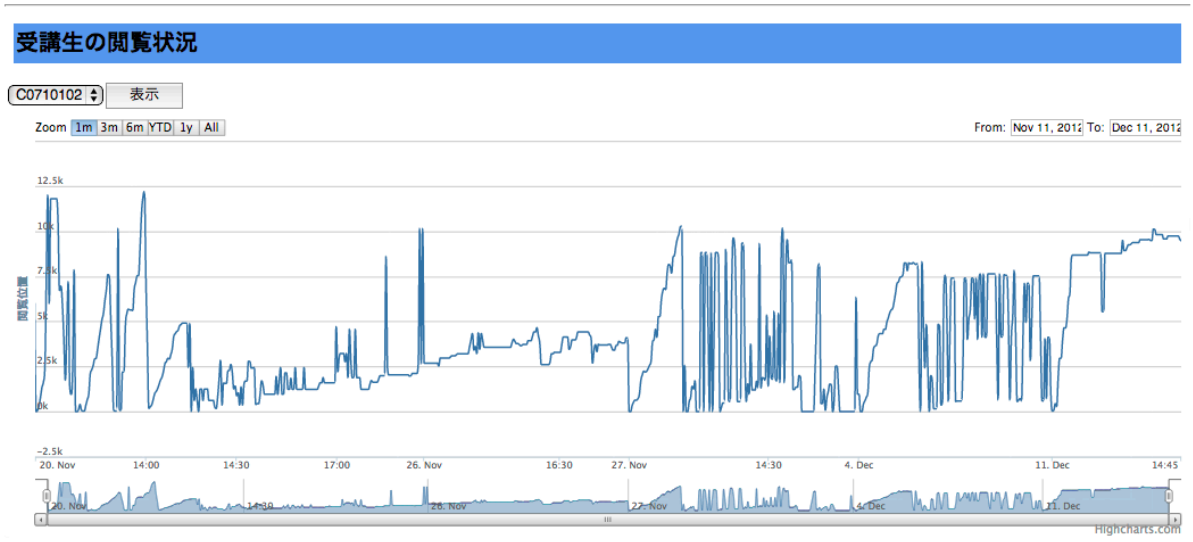

図 3 リフレクション支援ビュー：講義全体

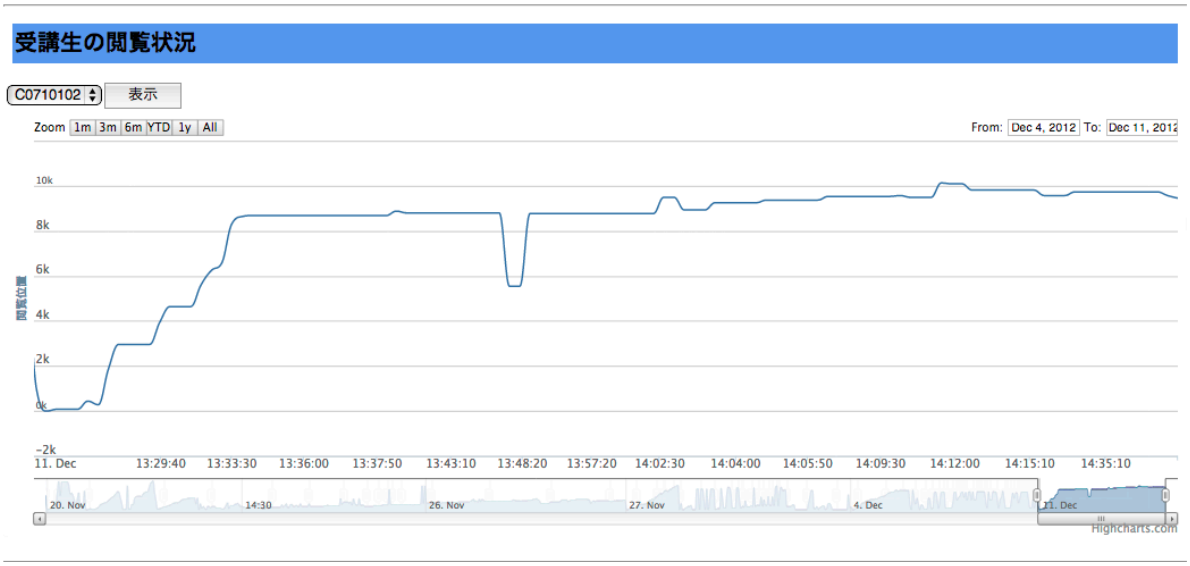

図 4 リフレクション支援ビュー：講義各回

5.1 講義 1 回分での考察

講義 1 回分での分析において，特徴的な事例を図 5 に挙げる．この例は，第 6 回における講義時間中の講義資料の閲覧箇所（HTML 資料の縦方向の座標）を表している．図 5 中の P1，P2，P3 において資料の手戻りが発生しているのが分かる．

この回は，13 時 33 分頃に座学が終了したので，手戻りが発生しているのはいずれも演習時間中である．当該受講生は，14 時 40 分にコーディング状況の把握システムで，プログラミングの文法が分からないことによる遅れが検知された．講義後に当該受講生にヒアリングしたところ，分からないプログラミングの文法を講義資料中から探していることによる手戻りであった．結果として目的の記述は以前の回の資料にあり，その回の資料中になかったため 3 回も手戻りが発生した．

この例では，授業進度と比較して遅れている受講生を検出するために取得する講義資料の閲覧箇所を，演習中も継続的にデータ表示をすることで，講義資料の内容が，演習問題を解く際の参考資料として十分かどうかを判断することができたと考えられる．

5.2 講義複数回分での考察

講義複数回分での分析において，特徴的な事例を図 6 に挙げる．この例は，演習を解くのが遅れがちな受講生の 3 回分の講義（第 4 回から第 6 回）における講義資料の閲覧箇所を表している．図 6 中の赤枠がそれぞれ講義の開始から終了までを囲んだものである．

当該受講生は，第 4 回，第 5 回の講義資料の閲覧状況において，座学，演習に関わらず手戻りを多くしているのが分かる．これにより次回以降に講師による積極的なフォ

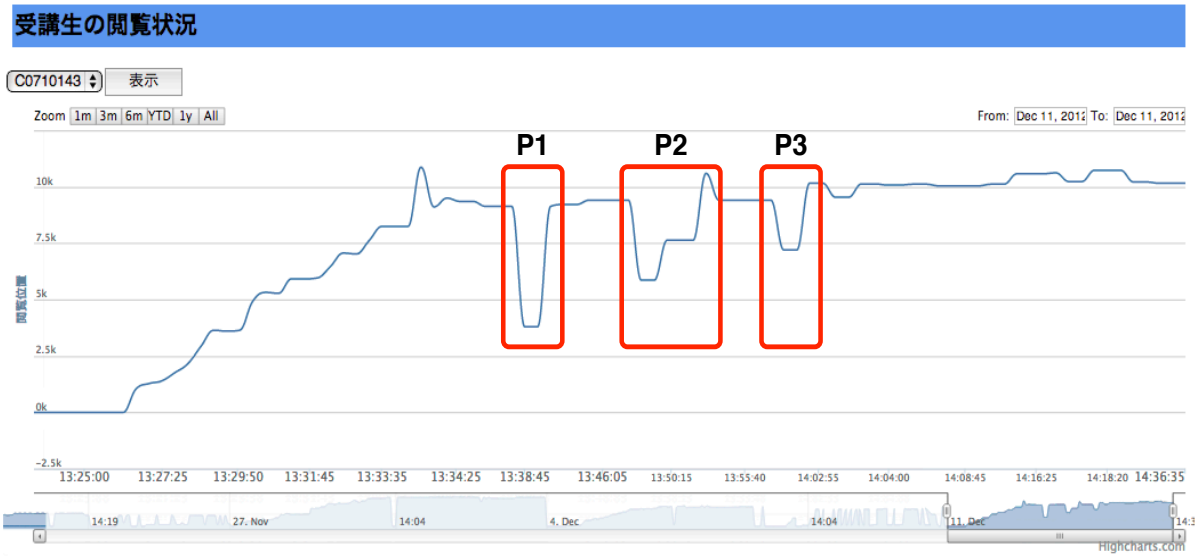

図 5 リフレクション支援ビューによる考察 1

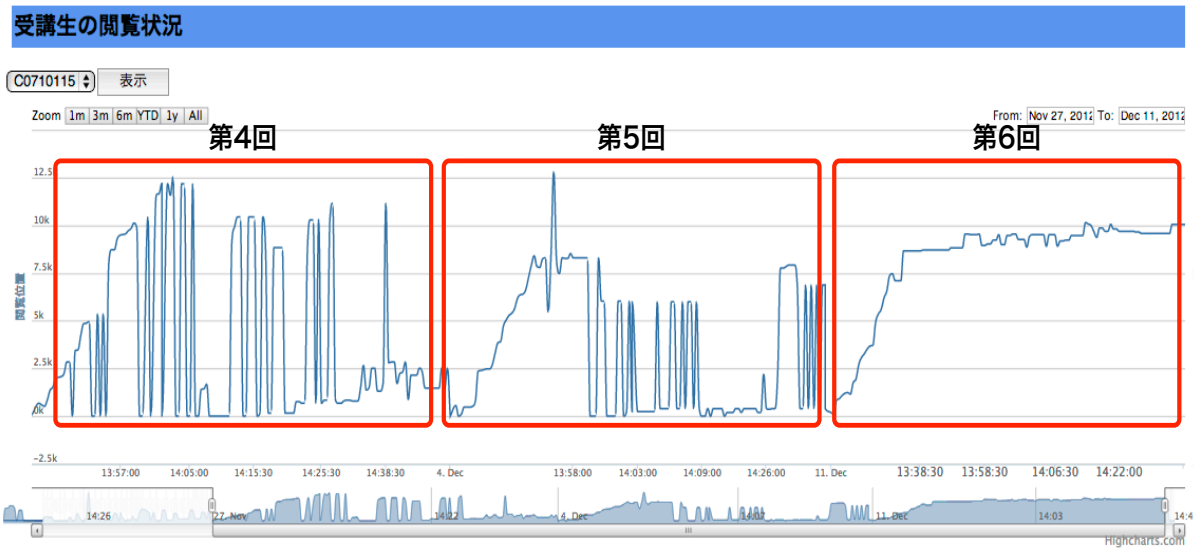

図 6 リフレクション支援ビューによる考察 2

ローが必要であると分かる．実際に第 6 回の講義において当該受講生へのフォローの回数を増やしたところ，図 6 中の第 6 回分の閲覧状況のように改善された．

この例では，複数回の講義を連続的にデータを表示することで，継続して問題を抱ている可能性がある学生がいないか確認することができた．また，講義後にリフレクションビューを閲覧することは，フォローが適切であったかのひとつの判断材料になると考えられる．

6. まとめ

本研究では，プログラミング講義中の受講生の状態を講義後に詳しく見直すことで，講義各回や全体の改善支援を行うリフレクション支援手法を提案した．この目的を達成するため，著者らが開発したシステムで記録した受講生の講義時間中の状態ログを，講義後にさまざまな観点から見直すことができるリフレクション支援ビューを提案した．評価実験として，受講生 12 名によるプログラミング講義において講義時間中の状態ログを収集し，分析を行った．結果として講義資料の内容が十分であるか，遅れている学生へのフォローが効果があったのか等の議論をすることができた．

今後の課題として，より FD の観点に立ったマクロの視点からの支援を増やすことが考えられる．例えば，講義計画全体からみて各回が適切であったかなどの判断をリフレクション支援ビューを利用することでサポートしたい．また，リフレクション支援ビューを継続的に利用することで得られる知見をもとに機能改善を行っていく予定である．

謝辞 本研究の一部は文部科学省科学研究費補助金（B）課題番号 23300049（2012 年）の支援により行われた．

参考文献

[1] 独立行政法人情報処理推進機構：IT 人材白書 2012 (2012).
[2] 宮地恵佑，高橋直久：構造誤り検出機能を有するアセンブラプログラミング演習支援システムの実現と評価 (¡特集¿ 教育システムにおけるプラットホームとコンテンツ開発論文)，電子情報通信学会論文誌. D, 情報・システム， Vol. 91, No. 2, pp. 280–292 (2008).
[3] 藤原理也，田口　浩，島田幸廣，高田秀志，島川博光：ストリームデータによる学習者のプログラミング状況把握，電子情報通信学会第 18 回データ工学ワークショップ (2007).
[4] 公益社団法人私立大学情報教育協会：私立大学情報環境白書（平成 23 年度版） (2012).
[5] 教師の資質向上を目指した授業リフレクション研究部：教師の資質向上を目指した授業リフレクションの在り方 (2012).
[6] 今野文子，樋口祐紀，三石　大：授業計画と実施結果の差異に着目した授業リフレクション手法の提案，日本教育工学会論文誌， Vol. 32, No. 4, pp. 383–393 (2009).
[7] 三石　大，今野文子，菅野裕佳，大河雄一：授業計画と実施結果の差異に着目した授業リフレクションの黒板利用型授業に対する実践と効果 (e-Learning と教育品質の保証・向上/一般)，電子情報通信学会技術研究報告. ET, 教育工学， Vol. 109, No. 268, pp. 35–40 (2009).
[8] 堀口悟史，井垣　宏，井上亮文，山田　誠，星　徹，岡田謙一：講義資料閲覧ログを用いたプログラミング講義進捗管理手法の提案，情報処理学会論文誌， Vol. 53, No. 1, pp. 61–71 (2012).
[9] 井垣　宏，齋藤　俊，井上亮文，中村亮太，楠本信二：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. 53, No. 1, pp. 61–71 (2012).