



Title	コンピュータ活用による運用負荷軽減を指向した授業支援環境の実現に関する研究
Author(s)	長瀧, 寛之
Citation	大阪大学, 2009, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/2736
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

コンピュータ活用による運用負荷軽減を指向した 授業支援環境の実現に関する研究

提出先 大阪大学大学院情報科学研究科

提出年月 2009 年 1 月

長 瀧 寛 之

発表論文リスト

学術論文

1. 長瀧寛之, 伊藤亮太, 大下福仁, 角川裕次, 増澤利光: “アルゴリズム学習における間違い探し形式の演習課題を自動生成する手法の提案と評価”, 情報処理学会論文誌, Vol.49, No.10, pp.3366–3376, Oct 2008.
2. 長瀧寛之, 永井孝幸, 都倉信樹: “学生が作業スタッフとして参加する学科内全科目の講義撮影・配信への取り組み”, 工学教育, Vol.55, No.5, pp.8–14, Sep 2007.

国際会議

3. Ryota Itoh, Hiroyuki Nagataki (presenter), Fukuhito Ooshita, Hirotugu Kakugawa and Toshimitsu Masuzawa: “A Fault Injection Method for Generating Error-correction Exercises in Algorithm Learning”, 8th International Conference on Information Technology Based Higher Education and Training (ITHET 2007) - Proceedings (CD-ROM), pp.200–205, Jul 2007.
4. Hiroyuki Nagataki, Nobuki Tokura, Takayuki Nagai, Hideo Masuda, Michio Nakanishi and Toshimitsu Masuzawa: “An Interactive Lecture Support System in a Classroom”, Proceeding of 3rd International Conference on Information Technology Based Higher Education and Training 2002 (CD-ROM), NAGATAKI.PDF, Jul 2002.

レターズ

5. 長瀧寛之, 永井孝幸, 都倉信樹: “講義中の学生と教師のやりとりを支援するシステムの構築と運用”, 情報技術レターズ (情報科学技術フォーラム (FIT 2003)), pp.387–388, Sep 2003.

研究報告

6. 長瀧寛之: “CMS 上でリスニング試験問題を作成する際の問題点とその解決手段の検討”, 情報処理学会研究グループ報告 第9回 CMS 研究発表会, pp.48–55, Sep 2008.

7. 長瀧寛之, 永井孝幸, 都倉信樹: “学科内全講義のビデオ撮影・蓄積・配信への取り組み”, 平成 17 年度工学・工業教育研究講演会講演論文集, pp.104–105, Sep 2005.
8. 長瀧寛之: “授業内のやりとり支援システム inLec の設計と高校での活用事例”, 第 1 回鳥取環境大学情報教育セミナー (高校・大学連携企画), Nov 2004.
9. 長瀧寛之, 北村英純, 永井孝幸, 都倉信樹: “遠隔ペアプログラミング支援システム SATORI の開発とプログラミング教育への適用”, 平成 16 年度工学・工業教育研究講演会講演論文集, pp.445–446, Jul 2004.
10. 長瀧寛之, 永井孝幸, 都倉信樹: “大学内でのアンケートシステムの運用事例と考察”, 平成 16 年度工学・工業教育研究講演会講演論文集, pp.191–192, Jul 2004.
11. 長瀧寛之, 永井孝幸, 都倉信樹: “学生による講義ビデオのしおり付け実験の報告”, 情報処理学会研究報告 Vol.2003, No.103 (コンピュータと教育研究会 2003-CE-71), pp.27–34, Oct 2003.
12. 長瀧寛之, 永井孝幸, 都倉信樹: “授業内の学生の反応を記録・解析するシステムの運用報告”, 情報処理学会研究報告 Vol.2004, No.13 (コンピュータと教育研究会 2003-CE-68), pp.47–54, Feb 2003.
13. 長瀧寛之, 永井孝幸, 都倉信樹: “生徒と教師の授業時間内のやりとりを支援するシステムの実装と評価”, 平成 14 年度工学・工業教育研究講演会講演論文集, pp.47–50, Jul 2002.
14. 長瀧寛之, 増澤利光, 都倉信樹: “生徒と教師の協調学習授業を支援するインタラクティブなシステムの提案”, 情報処理学会研究報告 Vol.2001, No.98 (グループウェアとネットワークサービス研究会 2001-GN-41), pp.61–66, Oct 2001.
15. 長瀧寛之, 都倉信樹: “生徒と教師の協調学習授業の支援を目指して -インタラクティブな支援システムの提案-”, 情報処理学会第 62 回全国大会 講演論文集, 第 4 分冊 pp.231–232, Mar 2001.
16. 長瀧寛之, 都倉信樹: “履歴を利用した問題選択を行う演習・試験システムの実現”, 情報処理学会第 61 回全国大会 講演論文集, 第 4 分冊 pp.367–368, Oct 2000.

その他論文リスト

研究報告（共著）

17. 林昌弘, 長瀧寛之, 大下福仁, 角川裕次, 増澤利光: “議論活動における調査資料の活用を支援するシステム HAKASE の構築”, 情報処理学会研究報告, Vol.2008, No.13 (コンピュータと教育研究会 2008-CE-93), pp.119–126, Feb 2008.
18. 伊藤亮太, 長瀧寛之, 大下福仁, 角川裕次, 増澤利光: “アルゴリズム学習における誤りからの学習を実現する演習課題の自動生成手法”, 電子情報通信学会研究報告, Vol.106, No.507 (教育工学研究会 ET2006-97), pp.81–86, Jan 2007.
19. 齊藤明紀, 長瀧寛之, 永井孝幸: “個人所有ノート PC を用いた計算機環境”, 情報処理学会研究報告, Vol.2005, No.83 (分散システム／インターネット運用技術研究会 2005-DSM-38), pp.29–34, Aug 2005.
20. 濱田正人, 長瀧寛之, 都倉信樹: “演習内に複数の講義室で情報を共有し蓄積できる教員用システムの開発と運用”, 電子情報通信学会技術研究報告 Vol.104 No.643 (教育工学研究会 ET-2004-97), pp.13–18, Jan 2005.
21. 北村英純, 長瀧寛之, 都倉信樹: “高等学校教育におけるポートフォリオシステムの研究”, 電子情報通信学会技術研究報告 Vol.103 No.536 (教育工学研究会 ET-2003-68), pp.47–52, Feb 2004.
22. 岡本幸, 長瀧寛之, 齊藤明紀, 都倉信樹, 増澤利光: “大学教育に適した学生ポートフォリオシステム”, 情報処理学会研究報告 Vol.2004, No.13 (コンピュータと教育研究会 2003-CE-68), pp.55–62, Feb 2003.
23. 梶田秀夫, 鈴木未央, 長瀧寛之, 中村聡史, 海貝明道, 小川剛史, 中西道雄: “Linux システムを中心とした授業支援・運用支援システムの構築”, 2001 年 PC カンファレンス論文集, pp.34–35, Aug 2001.

雑誌記事

24. 永井孝幸, 長瀧寛之: “特集 大規模分散ネットワーク環境における教育用計算機システム 2. 教育用計算機環境の事例 6. 必携パソコン化編”, 情報処理 45 巻 3 号, pp.255–258, Mar 2004.

概要

高性能な計算機や広帯域ネットワークの普及を背景に、近年様々な教育機関で、科目管理システム（Course Management System : CMS）導入など学習支援にコンピュータを活用する取り組みが行われている。しかしシステムの利用や運営に伴う人的あるいは金銭的コストの高さから、導入後の利用拡大が容易に進まず、部分的あるいは一時的な運用にとどまっているのが現状である。学習支援システムの有効活用を促進し継続的な運用を実現するには、システム運用にかかる運用者の負担を軽減するサポート体制が不可欠である。

著者は以前より、教育機関における計算機を活用した学習支援システムの構築、特にその運用に伴う負荷を軽減するための工夫に焦点を当てた研究を大阪大学と鳥取環境大学で行ってきた。本論文では教育機関における授業の支援を軸として、授業外学習、授業内活動、授業内演習の三つの視点それぞれについて、計算機による学習支援の研究成果を紹介する。

1つ目に授業外学習の支援として、大量の授業を撮影し学内配信するプロジェクトの運用手法とその実践結果について紹介する。著者は鳥取環境大学にて、授業を電子コンテンツとして授業外でも利用できる環境の実現を目指し、一学科の全科目全講義のビデオ撮影と学内オンデマンド配信を行うプロジェクトを実施した。当初から継続運用を目標としていたため、プロジェクト実現の際にかかる運用コストを下げるため、作業の単純化や計算機の支援による自動化などの工夫を行った結果、年間100万円程度の運用コストで2004-2005年度を通して1000を超える講義映像を蓄積し、かつ蓄積した講義映像が有効活用される体制を整えることができた。本論文では、講義映像撮影・学内配信システムの概要と工夫点を紹介し、実践結果の解析と考察について述べる。

2つ目に、授業内活動の支援として、授業内での学習者と教師のやりとりを支援するシステムの構築と運用について紹介する。本システムは授業中における学習者からのレスポンス取得を支援することで授業の活性化を促進し、同時に学習者と教師の活動履歴を電子的に保存し、授業を振り返る資料として履歴を活用することをねらいとしている。本システムは授業中の利用を想定し、また計算機の性能にできるだけ依存しないよう、簡易なインタフェースを備えたWebアプリケーションとしてシステムを構築した。本システムを2001-2004年に実際の授業で運用し、その結果を解析することで、目的とする学習者からのレスポンスの活発化と、授業活動の履歴取得が実現

できていることを確認した。本論文では、システムの概要と工夫点について説明し、実際の授業での実践結果を基に本システムの有用性について議論する。

3つ目に、授業内演習の支援として、演習問題の自動作成手法の提案とその評価について紹介する。ここでは、情報科学の基礎科目であるアルゴリズムの授業を対象とし、アルゴリズムの知識定着を目的とした演習として、著者らが提案する、提示されたソースコードの誤りを発見し修正する“間違い探し演習”の問題作成に焦点を当てる。本研究では、間違い探し演習問題の作成にかかる教師の作業負担を軽減する手法として、アルゴリズム設計パラダイムを利用したソースコードへの誤り自動挿入による、演習に適した問題の自動作成手法を構築した。本手法で生成される演習問題について複数人による評価を行った結果、教師が手作業で作成する場合と同等の品質の問題を本手法で生成できることを確認した。

目次

第1章 はじめに	1
1.1 背景	1
1.2 本論文の概要	4
1.2.1 運用コストを抑えた学科内全講義のオンデマンド配信への取り組み	4
1.2.2 授業内の学習者と教師のインタラクションを支援するシステムの構築と運用	4
1.2.3 アルゴリズム学習における間違い探し形式の演習問題を自動生成する手法 の提案と評価	5
第2章 運用コストを抑えた学科内全講義のオンデマンド配信への取り組み	7
2.1 はじめに	7
2.2 背景	8
2.3 鳥取環境大学での取り組み	9
2.3.1 全科目撮影・配信の背景	9
2.3.2 運用ポリシー	9
2.3.3 作業の概要	10
2.3.4 運用上の工夫	10
2.4 運用結果	15
2.4.1 配信講義数	15
2.4.2 閲覧者数	16
2.4.3 学生スタッフ	16
2.4.4 機材メンテナンス	17
2.4.5 教員の作業コスト	19
2.5 評価	20
2.5.1 機材投資	20
2.5.2 学生のスタッフ参加の意義	21
2.5.3 教員による映像の活用	21

2.6	おわりに	22
第3章	授業内の学習者と教師のインタラクションを支援するシステムの構築と運用	25
3.1	はじめに	25
3.2	inLec の概要	26
3.2.1	運用環境	26
3.2.2	提供する機能	27
3.2.3	その他の機能	32
3.3	実践	32
3.3.1	実践事例：同一科目での inLec 継続利用	32
3.3.2	実践事例：遠隔学習者を含む環境での inLec 利用	36
3.4	おわりに	42
第4章	アルゴリズム学習における間違い探し形式の演習課題を自動生成する手法の提案と評価	43
4.1	はじめに	43
4.2	演習形式の検討	44
4.2.1	演習問題の形式	44
4.2.2	演習に適した誤り	45
4.2.3	演習問題の自動作成	45
4.2.4	関連研究	46
4.3	誤り自動挿入の手法	46
4.3.1	誤り挿入対象の特定	47
4.3.2	誤りパターン挿入	50
4.4	実装	51
4.5	間違い探し演習の実践	55
4.6	提案手法の評価	58
4.6.1	評価方法	59
4.6.2	評価結果	59
4.6.3	提案手法の改善	60
4.6.4	提案手法の再現率評価	62
4.6.5	再現率向上への検討	63

4.7 おわりに 65

第 5 章 おわりに **67**

目 次

2.1	講義配信までの流れ	11
2.2	各作業の所要時間	12
2.3	レンダリング・オーサリングの流れ（括弧内は所要時間）	13
2.4	講義映像タイトル例	14
2.5	キャプション（画面右上）	14
2.6	作業記録表	15
2.7	科目あたり述べ閲覧数（昇順：横軸は科目名）	16
2.8	2005 年度学生スタッフの活動時間（単位:hour）	18
2.9	教員の講義映像閲覧回数（ただし管理者を除く）	22
3.1	inLec 概略図	26
3.2	簡易レスポンス機能インタフェース（学習者用）	28
3.3	簡易レスポンス機能インタフェース（教師用）	28
3.4	投影画面：簡易レスポンス集計状況表示	29
3.5	いつでも返答ボタン	30
3.6	教師用インタフェース：いつでも返答ボタン集計結果	30
3.7	投影画面：いつでも返答ボタン集計結果（画面下）	30
3.8	出欠確認画面	31
3.9	回答の推移（経過時間 1 分 35 秒あたりから「はい」と「いいえ」の回答数が逆転）	36
3.10	座席列とクリック数の関係（折れ線：平均，ひげ：分散）	37
3.11	簡易レスポンス機能による返答率：青谷高校（凡例の数字はクラス名）	39
3.12	質問「inLec では自分の意見を出しやすいか？」への回答	41
3.13	質問「inLec と通常講義のどちらがよかったか？」への回答	41
3.14	質問「対面と遠隔のどちらの授業がよかったか？」への回答	42
4.1	マージソートを実装したソースコードの例	48

4.2	試作システムの構成	52
4.3	中間ノード（構文木）の例	53
4.4	MIS ノード挿入と誤りパターン適用	54
4.5	演習問題の例	56
4.6	手作業誤りの例：配列引数 $[A-B] \rightarrow [A]$	64
4.7	手作業誤りの例：純変数 $\rightarrow$ 配列変数	65

表 目 次

2.1	当学科の開講科目数	9
2.2	撮影・配信講義数	15
2.3	学生スタッフ人数（括弧内：新規参加スタッフの人数）	17
2.4	購入機器（2004–2005 年度総計）	18
2.5	機器の故障点数（2004–2005 年度）	19
2.6	作業トラブル（括弧内は発生件数）	20
3.1	inLec 利用授業の諸情報	33
3.2	inLec 総クリック数：鳥取環境大学（日付は全て 2002 年）	34
3.3	簡易レスポンスボタンのクリック数	35
3.4	いつでも返答ボタン利用回数（一行入力欄は 10/24 から提供）	35
3.5	青谷高校の inLec 利用授業概要（日付は全て 2004 年）	37
3.6	inLec 総クリック数：青谷高校（日付は全て 2004 年，*:遠隔室）	38
3.7	いつでも返答ボタンクリック数:青谷高校（*:遠隔室）	40
4.1	誤り挿入対象と誤りパターンの対応表	50
4.2	講義での間違い探し演習実施状況	56
4.3	Q4 の回答の人数分布	57
4.4	Q4 の回答結果ごとの正答率	58
4.5	自動誤りサンプルの有用性支持率	60
4.6	有用評価が 50%以下の誤りパターン	61
4.7	自動誤りサンプルの有用性支持率（改善版）	62
4.8	提案手法の再現率（アルゴリズム別）	63
4.9	支持率別の再現率	63
4.10	支持率 50%以上の再現不可手作業誤りの誤りパターン	64

第1章 はじめに

1.1 背景

計算機やネットワークを活用した学習支援の研究は、教育学における主要な研究テーマの一つとして、理論、実践の両面から様々なアプローチが提案されている。古くはプログラム学習理論を背景としたドリル型演習のCAI（コンピュータ援用教育）が主流であったが、次第に問題を解くだけでなく、学習履歴の記録や学習者間コミュニケーションなど、学習に伴うあらゆる活動をコンピュータで支援しようとするアプローチへ広がっていった。特に近年は、広帯域の計算機ネットワークが一般家庭まで広く普及したことを背景として、いわゆる e-Learning に代表されるインターネットを活用した学習支援システムが注目を集めるようになり、企業や教育機関を中心に導入事例が増加している [1, 2]。

計算機を活用した学習支援システムへの教育機関の期待は大きい。一つには、これまで手作業で行ってきた学習・教育活動を計算機によって自動化することで、教育活動にかかる負荷を軽減することへの期待がある。Faculty Development など教育の改善や厳密な授業評価が求められる現在、教師の負担はますます増加する傾向にある [3]。教育活動にかかる負荷の軽減は、単に時間的余裕を生み出すだけでなく、教師の労力を事務作業でなく教育の質の向上へ注力できる効果も期待でき、教育評価と改善を求められる現場においては急務の課題と言える。

さらに、従来は実現が困難であった新しい学習環境を、計算機の活用によって提供できる可能性への期待がある。例えば計算機を活用した遠隔教育は、地理的・規模的なハンディを乗り越え多くの学習者に対面教育と同等の教育コンテンツを提供する手段として注目されており、その支援システム構築や運営の手法が提案されるだけでなく、実際にインターネットを介した遠隔学習で学位を取得できる教育機関が現れてきている [4]。

また学習活動のサポートをより効率的、効果的に行うツールとして、学習活動を統合的に管理する科目管理システム (CMS) が近年注目を集めている。CMS は教材の掲載やテスト、成績管理機能など、学習・教育活動に有用な様々な機能が一括提供されているだけでなく、システム利用を通して蓄積された、個々の学習者の詳細な学習履歴を解析し、授業担当者へのフィードバックやきめ細かい個別学習サポートを実現する機能も提供されている。CMS で提供される機能は、最

新の教育理論や実践結果を踏まえたものが多数含まれている。例えば主要な CMS の一つである moodle は、「知識は個人の内ではなく他者とのコミュニケーションを通して形成される」という、社会的構成主義の理論を背景として作成されたシステムである [5]。また過去に作成された教育コンテンツを新しい CMS で再利用しにくいという問題が起こったことから、コンテンツの共通規格として SCORM[6] が制定され、現在の多数の CMS が SCORM に対応している。このように、理論と実践両面からのアプローチにより、効果的な学習支援環境づくりのノウハウは着実に蓄積されてきており、今後学習支援システムのさらなる発展が期待される。

ただし、現在でも十分に解決されていない問題の一つとして、学習支援システムの運用にかかる利用者や管理者の負担増加が挙げられる。

コンピュータ操作を伴う学習支援システムの活用には、手作業とは勝手の違う作業内容の習得や、場合によっては前提となる情報リテラシ能力の習得が求められる。また学習効果の高いコンテンツを作るには、慎重な設計と膨大な作業量が要求される。結果としてシステム利用において、現在以上の負担を教師に要求することになる。学習システムの活用促進の方策としては、インストラクショナルデザイナーなど科目設計の専門スタッフと教員が連携して、学習コンテンツ作成を効率的に進める体制の必要性が指摘されている [7]。具体例としては、MIT の OpenCourseWare が挙げられる。コンテンツ公開作業にかかる負担を専属スタッフで請け負う体制を作ることによって、教師の負担を極力減少させ、プロジェクト参加への敷居を下げた。その結果、全科目の7割以上という学習支援プロジェクトの実践事例としては非常に高い教員の参加率を得、授業コンテンツが充実した結果、OpenCourseWare 本来の目的であった、コンテンツ公開による知識の共有促進や MIT 自身の宣伝効果の向上にもつながっていった [8]。しかし専門スタッフの確保などは教育機関の組織としての基礎体力に依存する部分があり、全ての教育機関がそのような体制を維持する余裕があるわけではない。また日本ではインストラクショナルデザイン (ID)[9] 自体の認識が一般的でなく、システムの活用は個々の教師の自助努力に期待されるのが現状である [1]。メディア教育開発センターによる 2007 年度の教育機関への調査結果によると、情報コミュニケーション技術 (ICT) 活用教育を導入していない教育機関が 23.8%、うち導入の予定もない教育機関が 15.8% 存在しており、その理由として導入のノウハウや予算の不足、費用対効果への疑問などが挙げられている [1]。その結果、報告される実践事例は人的資源の確保が容易な大規模大学でのもものが大半であり、その他はコンピュータ操作に慣れた一部の教師が積極的に利用するにとどまっている事例が少なくない。

以上の問題に対し著者は、学習支援環境の活用を促進させるには、可能な限りシステムの利用に伴うコストを軽減することが肝要と考えている。ここでいう負担とは、単に教師のコンテンツ

作成に伴う時間的な負担だけではなく、学習者の活動における時間的あるいは心理的な負担、運用に伴う金銭的負担や人的コストなども含めたものである。単純に作業負担の軽減だけを追求すると、肝心の学習者に提供する教育の質が低下することにつながりかねない。しかし負担軽減と教育の質のトレードオフの妥協点を探るよりも、トレードオフの存在自体を緩和する工夫がより重要といえる。小規模な教育機関でも学習支援システムの運用を容易に行えるようにすることで、成功事例が蓄積され、そのことが現在費用対効果に不安を持つ教育機関でのシステム導入の動機付けにつながることを期待できる。

教育の負担軽減という視点で見ると、解決すべき課題は主に、(1) CMS などの学習支援プラットフォームの構築、カリキュラム構成、学習コンテンツ作成など科目を問わず学習、教育活動全体に共通する課題と、(2) 情報リテラシ演習やアルゴリズム学習など特定の学習科目や、PBL や協調学習、ピアアセスメントなど、特定の学習/教育スタイルがそれぞれ抱える特有の課題の2つに分類できる。(1)の解決へのアプローチは科目を問わず広い範囲の学習・教育活動に影響をもたらす。一方(1)がカバーしきれない(2)の課題については、個別に解決のアプローチを考えていく必要がある。逆に(1)の解決で(2)の課題の一部も同時に解決されたり、(2)の解決へのアプローチが(1)の課題解決への糸口となる場合もあり、つまり、いずれの課題も学習支援研究における重要な対象であると考えられる。

著者はこれまで、教育活動における負荷を軽減しつつ学習効果の高い環境を実現するための、計算機による支援手法の提案を研究テーマとして、科目共通の課題と科目特有の課題の両面で様々なアプローチを提案し、実践を含めた評価によってその効果を確認してきた。科目共通の課題へのアプローチとしては、学習履歴を利用した適応型テスト環境の構築 [10] に始まり、学内オンラインアンケートの実施環境構築 [11]、大量の講義映像コンテンツを蓄積・配信する仕組み [12]、また授業におけるコミュニケーション促進を効果的に行う手段の検討 [13] を行ってきた。また、科目特有の課題へのアプローチには、本論文で紹介するアルゴリズム学習用の演習問題自動生成 [14] の他、情報リテラシ演習におけるスタッフ間の効率的な情報共有手法 [15]、プログラミング学習におけるペアプログラミング支援環境の実現 [16]、協調学習における議論活動の教育的効果の促進の研究 [17] などに取り組んできた。

本論文ではそのうち、主に教育機関における授業の支援を軸とした、授業外の学習、授業内の活動、授業内演習の3つの視点において、運用負荷の軽減という観点からアプローチを行った研究について紹介する。各研究の詳細については次節で述べる。

1.2 本論文の概要

本論文は、教育機関における授業支援を軸とした3つの研究テーマについて、これまで発表した論文や発表の成果をまとめたものである。以下本論文の構成を、各研究テーマの概要とともに述べる。

1.2.1 運用コストを抑えた学科内全講義のオンデマンド配信への取り組み

まず2章で、授業外学習支援として、著者が中心となって2004-2005年に実現した学内講義配信のプロジェクトについて紹介する。

本プロジェクトは、教育機関で通常行われる対面授業を授業外でも利用可能なコンテンツとして活用することを目的とし、1学科内の全科目全講義をビデオ映像として蓄積し、学内オンデマンド配信を実現しようというものである。実施にあたって、複数年にわたって継続的に運用する必要性から、妥当な人的・金銭的成本での運用が求められた。そこで学生が主なスタッフとして参加する運営体制とし、講義映像の蓄積・配信作業を計算機を用いて支援する仕組みを構築し、運営にかかる人的・金銭的成本の削減を狙った。結果、2年間で1000を超える講義映像の蓄積と学内配信を実現した。

本章では、妥当なコストで運用を行うための工夫点を中心に講義配信プロジェクトの実践における運用システムの概要について紹介する。その上で、2年間の実践結果について分析し、本取り組みの妥当性について議論する。

1.2.2 授業内の学習者と教師のインタラクションを支援するシステムの構築と運用

次に3章で、授業内学習支援として、学習者と教師のやり取りを支援するシステムについて紹介する。

授業に対する学習者から教師へのレスポンスを活発化させること、また授業内の学習者と教師の活動を容易に記録する環境の実現をねらい、学習者と教師の授業中におけるやり取りを支援するシステム **inLec** を構築した。inLec は、対面授業での利用に耐えうる簡便なインタフェースの提供によって、コンピュータによる学習者と教師のやり取りの支援を可能とする Web アプリケーションであり、授業における学習者からのレスポンスの容易な把握と授業へのフィードバックを支援する。

本章では inLec の概要について紹介し、2001–2004 年の実際の授業で inLec を利用した実践結果について述べる。その上で実践結果から inLec の有用性を評価し、また inLec で記録・抽出可能な授業中の学習者と教師の活動について議論する。

1.2.3 アルゴリズム学習における間違い探し形式の演習問題を自動生成する手法の提案と評価

4 章では、アルゴリズム学習を対象とした演習問題の自動生成手法について述べる。具体的には、アルゴリズム学習のスタイルとして「間違い探し演習」を提案し、間違い探し演習問題の自動生成を行うための手法について紹介する。

間違い探し演習とは、あるアルゴリズムが実装されたソースコードを学習者に提示し、その中に含まれるアルゴリズム上の誤りを指摘する形式の演習である。本章で述べる提案手法は、アルゴリズム設計パラダイムを利用した誤り挿入対象の特定と、文法ベースによるコード置換操作を組み合わせ、正しい実装のソースコードに対して間違い探し演習問題として適した誤りを挿入するというものである。本提案手法で自動生成できる演習問題を複数人で評価した結果、本提案手法で生成される問題は手作業で生成するものと同等に演習問題として適していることを確認した。

本論文では間違い探し演習用の課題を自動生成する手法について紹介し、提案手法の評価によって明らかになった本手法の有用性について議論する。

最後に 5 章で全体のまとめと、今後の研究の展開について議論する。

第2章 運用コストを抑えた学科内全講義のオンデマンド配信への取り組み

本章では、著者らが鳥取環境大学環境情報学部情報システム学科にて行った、講義収録とオンデマンド配信への取り組みについて報告する。

本取り組みには3つの大きな特色がある。

1. 学科の全講義の収録を行っていること。
2. 活動における初期コストと運用コストが低く抑えられていること
3. 有志学生がスタッフとして参加し、専属スタッフなしの運営であること。

2004年度と2005年度の2年間で、1本90分の講義を1000本以上収録したが、機器メンテナンスや人件費を含めた2年間の運営経費の合計は200万円以下であった。

本章では上記3つの特色を実現するための著者らの工夫を説明し、運用コストについて分析を行った結果について議論する。

2.1 はじめに

大学などの講義（対面講義）を映像として記録し、オンデマンド配信するという取り組みが増加している。これにより、(1) 学生の予習や復習、講義欠席時の補講教材などに講義映像を活用できる、(2) 授業評価と改善の参考資料として活用できるなど、対面講義の有効活用が期待されている [18]。

鳥取環境大学（以下当大学）においても、情報システム学科（以下当学科）内で開講されるほぼ全ての専門講義を撮影・蓄積し、学内でオンデマンド配信する取り組みを、2004年度から2008年度後期に至るまで継続的に行っている。本取り組みでは、学生が主要スタッフとして参加する形態をとっている。本章ではそのうち、当学科で2004年度から2005年度にかけて、著者が中心となって行った学科内全講義のオンデマンド配信への取り組みについて紹介する。またその運用

結果をもとに、学生がスタッフとして参加することの影響を、自動撮影・配信システムを導入する場合と比較しながら評価する。

以下、2.2節では、背景として対面講義のオンデマンド配信に関する既存の実践について紹介し、2.3節で、当大学の状況に合わせた、スタッフも含めた講義配信システムのポリシーと概要、運用上の工夫について述べる。2.4節で2004–2005年度の運用結果について紹介し、2.5節で運用結果の評価を行う。最後に2.6節でまとめを述べる。

2.2 背景

対面講義のオンデマンド配信は、これまでも東京大学 [19] などいくつかの大学において実践報告がなされている。実際に講義配信を採り入れている教育機関は全体の13%程度とまだ少ない [20] が、一部の大学では一学科の全専門科目の講義 (20–30 科目) を継続的に撮影・配信する取り組みが始まっている [21, 22]。

講義の撮影から配信までの作業は、撮影対象の講義数が多くなるほどその作業量が増加していく。そこで上記の講義配信事例においては、大量の講義配信の作業を省力化することをねらって講義室に自動撮影機材を設置するという方法を取る場合が多い [22, 23]。講義を自動撮影するシステムには、教師をカメラで自動追尾したり [18]、複数のカメラを特定のアングルで固定したり [24, 25] といった手段をとり、また授業情報を補完するために、Powerpoint との連動、電子ホワイトボードによる板書の収録など映像以外の授業情報を同時収録する [26] 機能を持つものが多い。

しかし、大量の講義を撮影するという点から見ると、自動撮影機材の導入は必ずしも得策とは言えない。まず、自動撮影機材の導入は初期投資が高額である [27]。しかも、撮影機材が導入されていない講義室では自動撮影ができないため、手作業で撮影できる機材を別途用意するか、講義が行われる全ての部屋へ撮影機材を導入する必要がある。全科目撮影となると利用する講義室の数も多くなるため、設置コストも大きくなる。

さらに、カメラ自動追尾や Powerpoint 連携などコンテンツ作成支援の機能は、「教師は常に教壇にいる」「Powerpoint を使う」など特定の講義進行スタイルを前提としているが、実際の講義スタイルは教師によって、また科目内容によって様々であり、支援機能が前提としない授業スタイルが多ければ投資の無駄になる可能性もある。システムの機能に講義スタイルをあわせるよう教師に求めるという対策もあるが、講義スタイルの多様性を狭め教師の負担も増加することから、教師から講義撮影への協力を得られない可能性がある。

表 2.1: 当学科の開講科目数

	開講講義数	同時開講数 (最大)	1 日の合計 開講数 (最大)
2004 年度前期	17	2	5
2004 年度後期	23	3	6
2005 年度前期	20	3	7
2005 年度後期	30	3	9

2.3 鳥取環境大学での取り組み

2.3.1 全科目撮影・配信の背景

当大学では、全学生が入学時にノートパソコンを購入する。さらに学内のいたるところに情報コンセントを配置することで、学生が日常的に学内外のネットワークを利用し、学習ツールとしてノートパソコンを活用する環境が整っている。当学科ではこの環境を活用して、講義資料の配布やオンラインレポート提出など様々な教育支援を行っている [28, 29].

そして、学生が履修授業の復習や欠席時の補講資料として利用することを目的とし、2002 年度後期から履修講義のオンデマンド配信を行う取り組みをスタートした。当初は試験的に一部の講義に限って撮影を行っていたが、2004 年度から撮影対象科目を、原則当学科の全ての科目へと拡張することになった [30].

なお、以下当学科における学内講義の撮影・配信の取り組みを‘本取り組み’と表記する。

2.3.2 運用ポリシー

本取り組みでは、講義撮影とコンテンツ作成の作業を、当学科の学生有志（以下学生スタッフ）が主体となって行う形をとった。当学科の教員は 2005 年度時点で 15 人（うち教授 10 人、助教授 3 人、助手 2 人）であり、週 20 を超える講義 (表 2.1 参照) の撮影を行う人手を教員だけで確保する余裕はなかった。しかし当学科の学生は元々ビデオ撮影や編集作業の専門家ではないため、学生スタッフ間で技術力の個人差が大きい。さらに入学・進級・卒業によって頻繁にスタッフ構成が入れ替わるため、継続的な活動による技術の蓄積が期待できない。そのため作業内容をいかに単純化するか考慮する必要があった。

本取り組みでは講義の撮影方法として、自動撮影システムの導入はせず、講義の都度スタッフが撮影機材を講義室に持ち込み、手動で撮影を行うこととした。当学科では科目ごと、また教師ごとの様々な講義スタイルを制限しないこととしたため、講義スタイルに合わせて撮影方法を適宜調整する必要がある。そのため特定の講義スタイルを前提とする自動撮影システムを導入しても、当学科では有効活用できないと考えた。また、当学科では最大で3講義が同時に開講される(表2.1 参照)ことから、全科目を撮影するには自動撮影機材を最大3セット導入する必要があり、コスト面からも導入は困難であった。

2.3.3 作業の概要

一般的に講義を配信するまでには、(1) 撮影、(2) キャプチャ(計算機への撮影映像の取り込み)、(3) 編集(映像・音声の加工)、(4) レンダリング(配信に適したファイル形式への変換)、(5) 配信準備(配信サーバへのファイル追加、Web 上のリンク追加など)の手順で作業を行う必要がある。当学科でもこの手順により講義映像コンテンツを作成する。また、オフラインでの閲覧とバックアップ用途として、講義映像のDVDも同時に作成することとした。

全作業のうち撮影・キャプチャ・編集は学生スタッフによる手作業で、レンダリングと配信準備は自動処理で行うように実装した。またレンダリングについては、市販のシステムは極力使用せず、フリーソフトを組み合わせ作業を自動化するシステムを構築した。当学科での講義配信の流れを図2.1に示す。また各作業の所要時間を図2.2に、レンダリングとオーサリング(DVDイメージ作成)の詳細な手順を図2.3に示す。

2.3.4 運用上の工夫

対面講義の撮影は最も重要な作業である。キャプチャ以降の作業は失敗してもやり直しが効くが、講義は一度しか行われないため、撮影作業は失敗するとやり直しが不可能である。しかし撮影を手動で行う以上、人的ミスの発生を避けることは難しい。そのため、撮影作業を単純化して人的ミスの可能性を減らし、またトラブルが発生した場合に備えて代替手段をとれるような事前対策をとることが必要である。

撮影機材の設置は講義の合間に行う必要があるが、この時間は最大でも10分しかない上、担当の学生スタッフが直前に授業を履修している場合、撮影準備の時間確保が困難である。そのため、講義開始までに確実に撮影準備を完了させるため、講義室内でのスタッフの作業は「カメラを講義室に置く」「電源を入れる」だけで済むようにした。記録メディアの挿入やマイク接続、カメラ

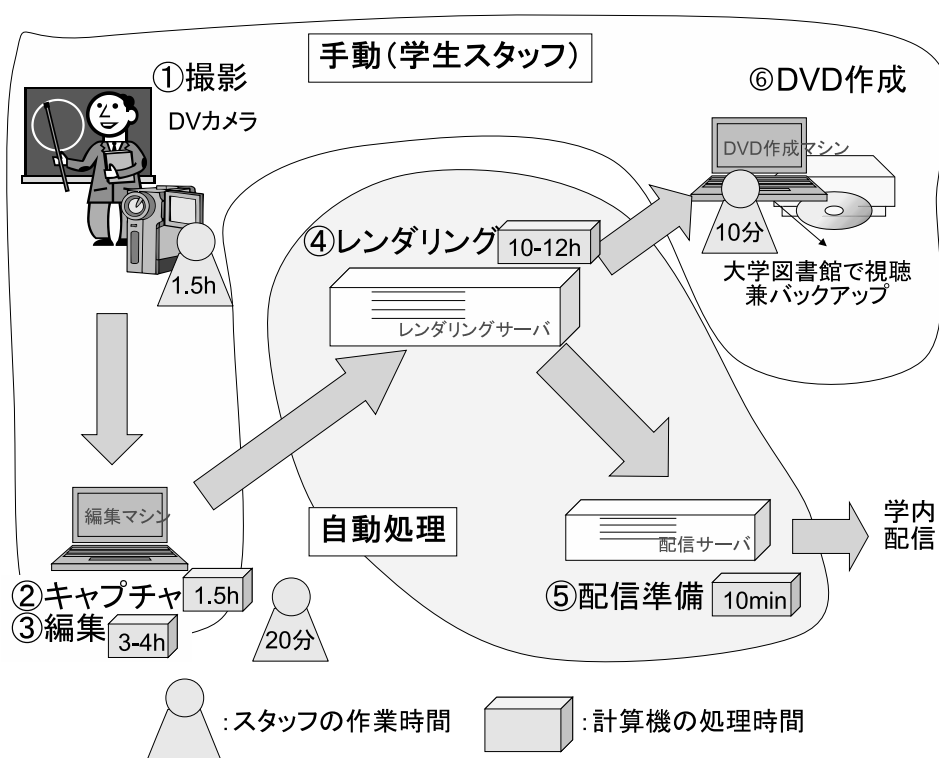


図 2.1: 講義配信までの流れ

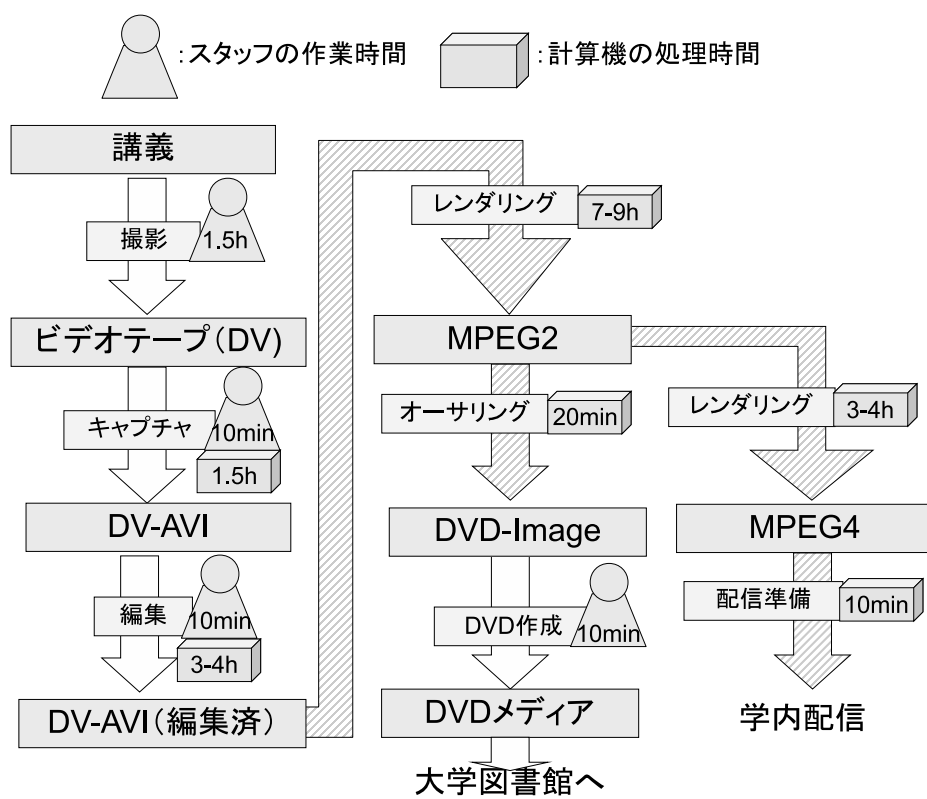


図 2.2: 各作業の所要時間

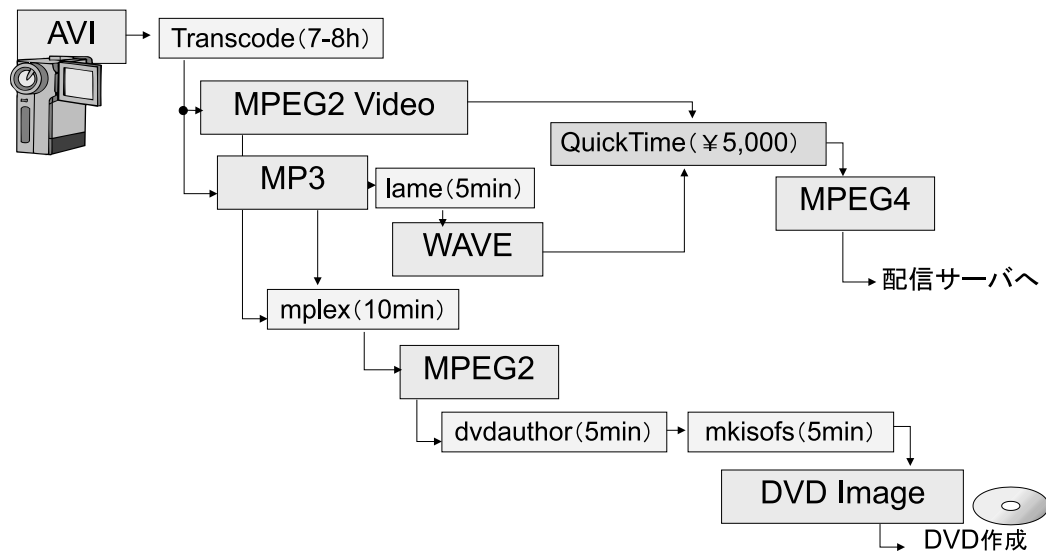


図 2.3: レンダリング・オーサリングの流れ（括弧内は所要時間）

の三脚への接続など、講義室以外でも行える準備作業は、前回そのカメラで撮影を行ったスタッフが撮影終了後すぐに行うようにした。

手作業による撮影では、適切な場面を収録するようにスタッフが適宜カメラのアングルを調整する必要があるが、この操作は撮影者の技術力に依存する。学生スタッフの技術力のばらつきが大きい本取り組みでは操作ミスの発生が避けられない。そのため、撮影アングルに依存せず収録が可能な講義中の音声を確実に収録できるように対策を取った。音声収録には指向性の高い外部接続マイクを用意し、また撮影中はビデオカメラに接続したイヤホンで常時収録中の音声を確認することとした。これにより収録音声の異常を感じたら迅速にカメラ内蔵マイクに切り替えることで、講義の音声収録ミスを極力防ぐことができる。さらにバッテリー切れによる撮影中断が起らないよう、ビデオカメラの電源はバッテリーではなく AC アダプタから直接供給するようにした。

編集作業についても、人的ミスの発生を防ぐため、大部分の作業を自動化するとともに手作業自体も極力単純化するようにした。具体的には手動で行う編集作業は、タイトル（図 2.4）やキャプション（図 2.5）の付加と全体的な音量調整のみで、録画した映像は基本的に無編集とした。映像のレンダリングや配信準備は自動化したため、講義映像 1 本あたりの編集作業時間は 10 分程度となり、大量の講義が蓄積される本取り組みの状況でも学生スタッフが十分対応可能な範囲内である。無編集の講義映像を効率よく閲覧するために、映像の任意の位置に閲覧者がインデックスを付加できるシステム [31] を閲覧インターフェースとした。



図 2.4: 講義映像タイトル例

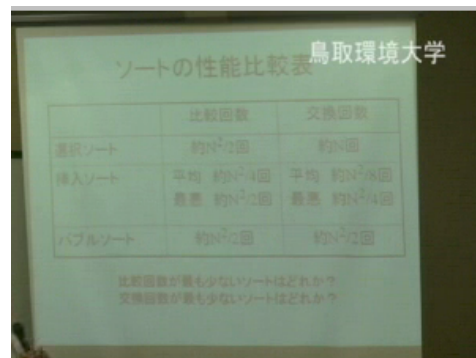


図 2.5: キャプション（画面右上）

本取り組みでは撮影講義数が多いため、撮影された講義がすべて確実に配信できているかどうかをスタッフ全員が把握する作業が困難になる。また、撮影したはずのある講義が配信されていない問題が見つかった場合、原因として(1)撮影ミス、(2)編集作業の遅れ、(3)レンダリングのエラー、(4)サーバ自体の故障、(5)ネットワークの不調など様々な可能性が考えられるため、問題に迅速に対処するためには作業状況を容易に把握する仕組みが必要であった。そこで、各作業の進捗状況を管理するツールを Web 上に作成した(図 2.6)。講義映像ごとに作業の進捗が○印で示されており、スタッフは Web ブラウザを通して学内どこからでも作業状況を閲覧可能である。また撮影時に気づいたことや編集時の留意事項などを、学生スタッフが自由記述できる項目も用意して、スタッフ間で情報共有をはかり、トラブル発生時には原因追及の参考資料として活用した。

科目名	日付	撮影	キャプ	編集	配信準備	DVD焼き	DVD印刷	メモ
情報ネットワークシステム演習1	2006/08/02	○	○	○	○	○	○	メモ
センシングシステム	2006/07/27	○	○	○	○	○	○	メモ
情報ネットワークシステム演習1	2006/07/26	○	○	○	○	○	○	メモ
コンピュータ入門	2006/07/26	○	○	○	○	○	○	メモ
ネットワークプログラミング論2	2006/07/24	○	○	○	○	○	○	メモ
データ構造とアルゴリズム	2006/07/24	○	○	○	○	○	○	メモ
オブジェクト指向設計	2006/07/24	○	○	○	○	○	○	メモ
情報システム基礎A	2006/07/24	○	○	○	○	○	○	メモ
データベース	2006/07/21	○	○	○	○	○	○	メモ
センシングシステム	2006/07/21	○	○	○	○	○	○	メモ
コンピュータ入門	2006/07/21	○	○	○	○	○	○	メモ

図 2.6: 作業記録表

表 2.2: 撮影・配信講義数

	撮影科目	撮影講義数
2004 年度前期	18	143
2004 年度後期	25	283
2005 年度前期	20	238
2005 年度後期	30	355
合計		1019

2.4 運用結果

2.4.1 配信講義数

本取り組みでは2004年度と2005年度を通して、合計1000を超える講義の撮影・配信を実現した(表2.2)。

当学科開講の講義のうち、外部講師による講義では撮影許可が得られない場合があった。また、機器故障などの原因で撮影に失敗する事例がいくつかあったが、それ以外の全ての講義について撮影することができた。

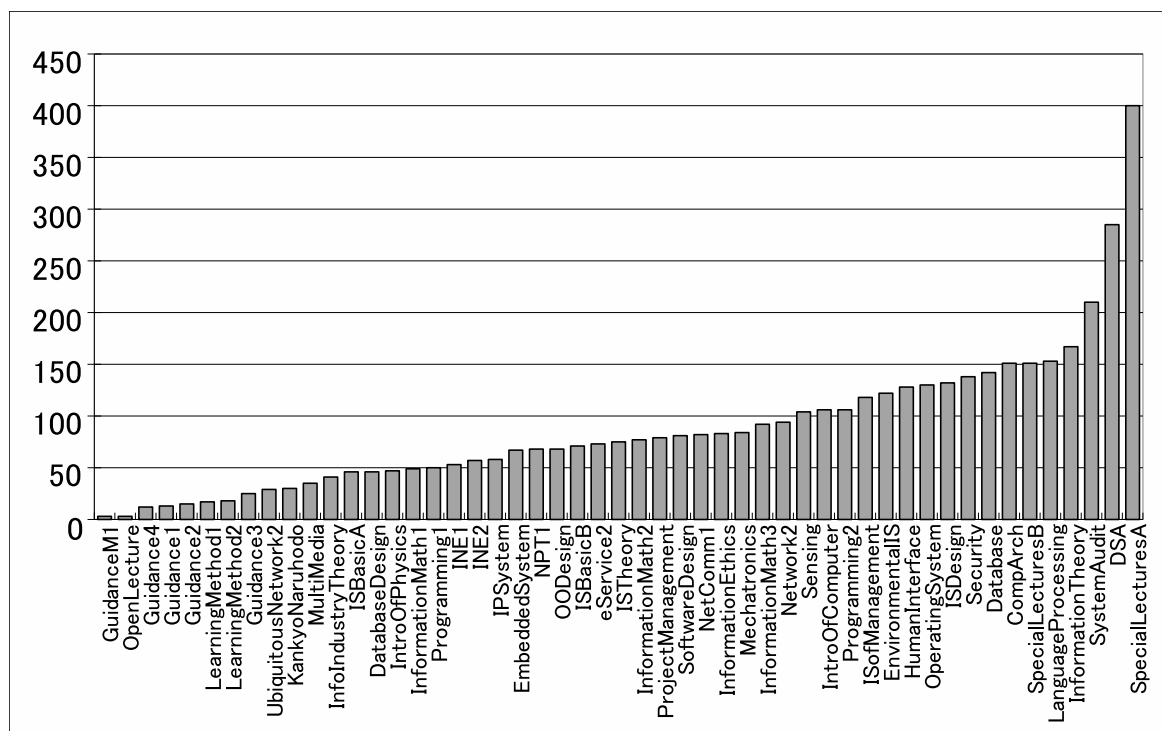


図 2.7: 科目あたり述べ閲覧数（昇順：横軸は科目名）

2.4.2 閲覧者数

2004年度は閲覧者の記録が取れていなかったため正確な人数を出せなかったが、接続履歴のIPアドレスから100人を超えるアクセスがあったと考えられる。2005年度は閲覧者の記録から、前期81人、後期115人が講義映像を閲覧した。2005年度において情報システム学科の学生総数は292人であったため、ほぼ1/4-1/3の学生が利用したことになる。

科目ごとに講義映像の延べ閲覧数を見ると、閲覧者数に大きな偏りが見られた(図2.7)。主に頻繁なレポート提出が必要な科目や必修科目について、閲覧数が増大する傾向にあった。最も閲覧数が多かった科目は4年生対象の前期必修科目で、就職活動で頻繁に講義を欠席しがちな時期であることから、補講目的での閲覧数が伸びたものと思われる。

2.4.3 学生スタッフ

2004年度、2005年度の各学期に参加した学生スタッフの人数を表2.3に示す。基本的には履修スケジュールに自由度のある学部3,4年生が参加スタッフの大半を占めた。また、前学期に参加し

表 2.3: 学生スタッフ人数（括弧内：新規参加スタッフの人数）

学年	2 年	3 年	4 年	M1
2004 年度前期	0	3	5	
2004 年度後期	0	3	8(3)	
2005 年度前期	3(3)	1(1)	4(2)	2(1)
2005 年度後期	5(2)	4(3)	3	2

た学生スタッフは次の学期以降も継続的に参加していたが、新規参加の学生スタッフも各学期に一定数存在した。特に 2004 年度学生スタッフのうち学部 4 年生が卒業（1 名は大学院に進学）した後の 2005 年度前期は、新規参加のスタッフが 10 人中 7 人を占めた。

各学期のスタッフの一人あたり平均撮影講義数は、2004 年度前期は 17 講義、2004 年度後期以降はいずれも 23–25 講義程度であった。スタッフの作業時間は、1 講義あたり撮影 1.5 時間、編集（DVD 作成含む）0.5 時間であった。これにより、各学期の学生スタッフ一人あたりの総作業時間は、2005 年度前期は 47.6 時間、後期は 50.7 時間となり、1 週間あたりの平均作業時間はそれぞれ 3.4 時間、3.6 時間と、毎週約 2 講義分の作業量となる。これは学部生でも十分作業可能な範囲の時間であると考えられる。

しかし実際は、各学生スタッフの履修スケジュールとの兼ね合いで、時間割に余裕のある一部の学生に作業量が偏りがちになった (図 2.8)。

スタッフへの謝金に関しては、時給 800 円を基準に支出した結果、2004 年度は合計 511,200 円、2005 年度は合計 711,600 円となった。結果、機器メンテナンスとメディア購入費用を合わせると、年間の経費は 2004 年度で約 80 万円、2005 年度は約 116 万円となった。

2.4.4 機材メンテナンス

本取り組みにおいて 2005 年度後期までに使用した機材一式を、表 2.4 に示す。機材の中で、レンダリングサーバについては高性能のサーバマシンを導入した。レンダリング処理は計算負荷が高く、しかも本取り組みでは 1 日にいくつもの講義映像を同時レンダリングしなければならないため、高負荷でも安定動作するサーバが必要であった。また配信サーバは大量の講義映像ファイルを蓄積するため、合計 400GB の HDD を持つサーバを用意した。サーバマシン以外の機材に関しては、家電量販店で購入可能な機材で運用上支障をきたすことはなかった。

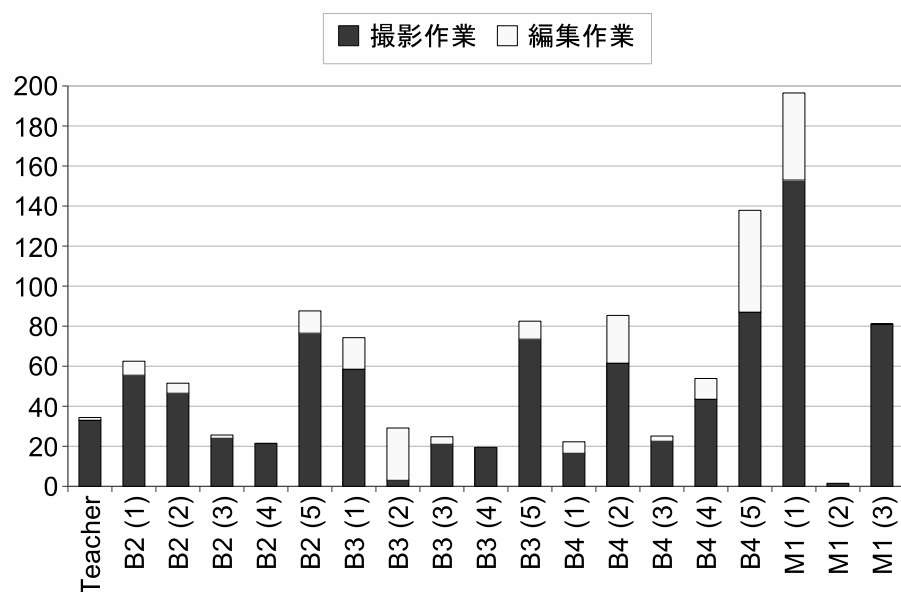


図 2.8: 2005 年度学生スタッフの活動時間 (単位:hour)

表 2.4: 購入機器 (2004-2005 年度総計)

機器名称	個数
DV カメラ	6
指向性マイク	3
三脚	3
編集用 PC	4
レンダリングサーバ	2
配信サーバ	1
DVD 作成・プリント用 PC	2
DVD ドライブ	1
プリンタ (DVD 印字用)	1

表 2.5: 機器の故障点数 (2004–2005 年度)

故障箇所	故障件数
カメラ (録画／再生異常)	4
三脚 (破損)	1
編集用 PC (動作不良)	1
DVD ドライブ (書込異常)	2
配信サーバ (電源故障)	1
配信サーバ (基盤故障)	1
レンダリングサーバ (HDD 故障)	4

次に機器の故障件数を表 2.5 に示す。使用した各機器は、ビデオカメラのテープ装置、PC・サーバマシンの HDD やファン、DVD ドライブなど、駆動部分を持つものが多いため、本質的に故障率が高いと言える。実際本取り組みでは、ビデオカメラは 1 年程度で録画・再生ができなくなり、DVD ドライブは 1 年程度で書込みエラーが頻発する状態になった。故障機器の修理・交換にかかった費用は、2004 年度が 10 万円、2005 年度は約 20 万円であった。

機器類の他に、映像記録用メディアとして DV テープと DVD-R の購入も随時行う必要があった。撮影済の DV テープはマスターデータとして全て保存するポリシーとしたため、講義 1 回に対して各メディアを 1 本ずつ使用した。メディア購入の総費用は、2004 年度は 20 万円、2005 年度は 26 万円となり、うち 3/4 は DV テープの購入費用であった。

2.4.5 教員の作業コスト

本取り組みにおいては、学生スタッフだけでなくスタッフへの指導や機器類の管理など、講義撮影・配信作業のとりまとめを行う管理者の存在が必要であった。当学科では管理者の作業を筆者ら教員が行った。

管理者の合計作業時間は、学期ごとで 132.5 時間、年間 265 時間となった。これは 2.4.3 節で求めた学生スタッフの平均作業量と比べて約 2.6–2.7 倍という負荷であり、平均すると毎週 10 時間は講義配信の管理作業に追われるという状況であった。作業内容の大半は、機器故障やソフトウェアエラーなどのトラブル発生における対処であった。

撮影機器のトラブルが発生した場合でも講義は継続的に行われるため、撮影や配信が中止され

表 2.6: 作業トラブル (括弧内は発生件数)

事 例	主な原因
撮影失敗	マイク不調 (1), カメラ故障 (4), 音声収録ミス (7), スタッフ遅刻 (1), 撮影済テープ上書き (5)
編集・レンダ リング失敗	講義情報登録ミス (23), タイトル編集ミス (5), レンダリングエラー (24), サーバ HDD 故障 (2), 編集マシン HDD 容量不足 (4)
DVD 作成失敗	DVD ドライブ故障 (2)
配信失敗	サーバ故障 (1)

ることがないよう、迅速に対策をとらなければならない。表 2.6 は本取り組みで発生したトラブルの内訳である。この結果から、平均して週に 2.8 件のペースでトラブルが発生しているが、学生スタッフは自身の履修授業などの関係で常時対応が困難であったため、管理者である教員が主にその対応を行わなければならなかった。

2.5 評価

2.5.1 機材投資

全科目の撮影を行う場合、撮影機材の故障時に迅速に代替機材を準備できるかどうか重要なポイントとなる。機器が故障しても講義は次々と予定通り行われるため、修理や代替機材の準備が遅れると、撮影できない講義の数がどんどん増加していく。たとえば当学科では、撮影機材が一日利用できないだけで最大 9 つもの講義が撮影できないことになる。

市販の安価な機材でシステムを構築できるということは、単に予算を抑えるだけでなく、容易にかつ豊富に代替機材を準備できるという利点がある。対して自動撮影システムで使用する機材は、そのシステム専用に機種が限定されている場合があり、代替機器の調達が可能としても時間や費用がかかる可能性がある。

また、機器故障発生の可能性は自動撮影・手動撮影に関わらず常に存在するため、機器に異常が発生していないかを常時監視するスタッフが必要になる。自動撮影システムの場合は担当者を別途用意しなければならないが、当学科では学生スタッフが、講義撮影や編集活動を通して機器の異常発生を常時監視している形になる。また、著者らが毎日学生スタッフと、メーリングリストだけでなく口頭でも情報交換を行うことで、取り組みの問題点や機器不良の兆候を早期に把握し対処できる運用が可能となった。

2.5.2 学生のスタッフ参加の意義

自動撮影システムの利点としては、人的コストがかからないという点が挙げられる。本取り組みでは学生スタッフへの謝金によって年間70万円を超える金銭コストが発生したが、自動撮影の場合はこの謝金コストが発生しない。また学生スタッフは学期ごとにメンバーが入れ替わり、特に年度ごとに学生スタッフ全体のスキルがほぼリセットされるため（表2.3参照）、学生スタッフ主体の運用でも作業内容の指導などで教員側の作業コストは決して少なくなる。

しかし、自動・手動に関わらず、出来上がる講義映像コンテンツの数は同じであり、映像の閲覧者にとって撮影手法は意識することではない。しかも学生スタッフにとっては、講義撮影や配信準備の活動に参加することは、映像配信の知識や技術を身に付けるだけでなく、撮影作業を通して講義の復習や予習を行うなど、さらなる学習機会を得られる場となる。実際に学生スタッフにインタビューを行った結果、複数の学生スタッフから、過去に履修した学習内容への理解がスタッフ活動を通してさらに深まった、という意見が得られており、その学習効果が期待できると言える。対して自動撮影システムの場合、人的作業量は節約できるが、機材は数年で故障や性能劣化が発生し、最終的に廃棄処分となる。

以上の点から、教育機関においては、学習機会の増加につながる本取り組みの形が、自動撮影システムの導入よりも妥当な選択であると考えられる。さらに謝金を除いた機器メンテナンス費用は年間20-30万円程度であることを考えると、継続的な運用における金銭的コストの負担も妥当な範囲内であると言える。

2.5.3 教員による映像の活用

閲覧者の記録から、学習者の復習の他に、教員が自身の講義や他の教員が行う講義を閲覧する目的で講義映像へアクセスするという利用形態も確認された。自身の講義を客観的に閲覧するという行為は講義映像を撮影しなければ不可能であり、また他の教員の講義を閲覧することで、講

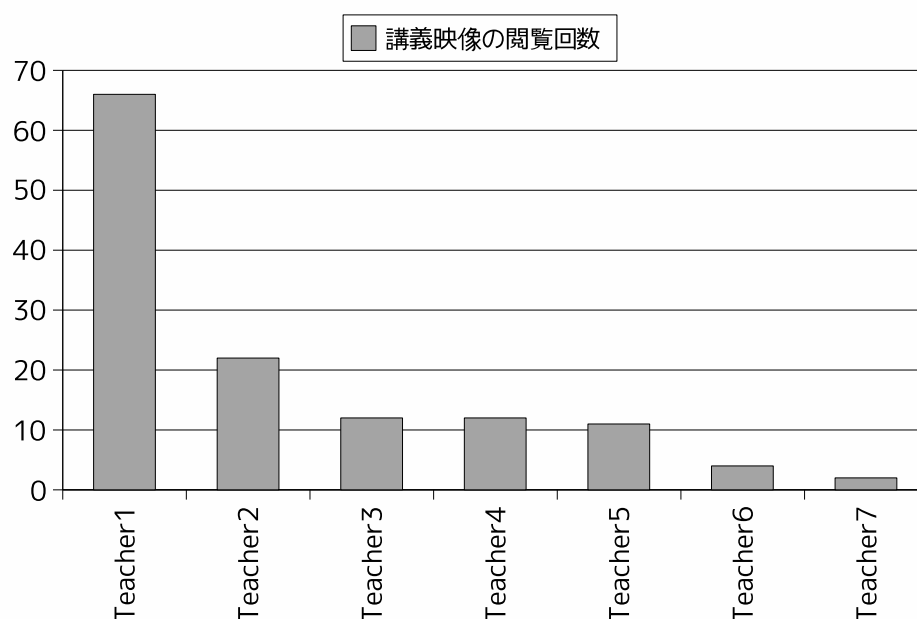


図 2.9: 教員の講義映像閲覧回数（ただし管理者を除く）

義方法を参考にしたり担当講義の進め方を調整したりといった目的に利用するだけでなく、近年多くの教育機関で求められている授業評価の参考資料として活用できることが期待できる。

ただし本取り組みでは、管理者を除いて講義映像を積極的に閲覧した教師は14人中7人に留まっている（図2.9）。蓄積した講義映像をさらに有効活用させるためには、将来的には教員の閲覧も考慮した仕組みの実現が必要であると考えられる。

2.6 おわりに

本章では、鳥取環境大学における学科内の全ての講義の撮影・配信への取り組みについて、実現手法として学生スタッフの活用、市販機材の利用、計算機による作業の省力化と自動化など、妥当なコストで複数年の継続運用を可能とするための運用手法について紹介した。2004–2005年度の実運用の結果、人件費と機器メンテナンスを含め年間100万円の運営経費で、2年間で1000を超える数の講義映像を蓄積し学内オンデマンド配信できたことを確認した。また運用結果を検証し、講義映像が学生に有効活用されていることを確認した。

講義映像をどのような目的で活用するかで、必要となる機能や労力は変化しうる。さらに人や資金など利用可能な資源によって、実現手法は変化しうる。本研究では資金やスタッフが豊富に

確保できない教育機関でも、専用のシステムを使わず妥当なコストで全科目の講義配信を実現することが可能であることを確認した。

なお当学科では、本取り組みを通して講義映像を活用する学習・教育スタイルが定着しており、著者が本取り組みを終了した 2006 年度以降も、システムの改良を加えながら講義映像配信の取り組みを継続している [32]。

第3章 授業内の学習者と教師のインタラクション を支援するシステムの構築と運用

本章では、授業内の学習支援として、授業中の学習者と教師のやりとりを支援するシステム **inLec** について紹介する。inLec は対面授業での使用を想定したシステムであり、本来の授業活動を妨げることなく、授業中の学習者と教師のやりとりをコンピュータの支援によって活発化することを目的とする。また inLec を利用することで、授業内の活動履歴を記録することを目的としている。

inLec を実際の授業で運用した結果、学習者からの積極的なレスポンスが得られることを確認した。また inLec の利用履歴から、授業の活動を抽出できることを確認した。

3.1 はじめに

コンピュータを活用した自主学習の支援環境が充実してきた現在でも、学習者と教師が直接接する対面授業は、学習のモチベーションを高める場として重要と考えられている [33]。対面授業の学習効果を高めるためには、学習者が積極的に授業へ参加し教師とコミュニケーションを図る状況が望ましい。しかし実際の授業では、学習者が授業に対する積極的な反応を見せないため、教師から学習者への一方的な情報伝達にとどまる事例が少なくない [34]。逆に受講者の多い授業では、学習者が積極的に反応してもそれを把握する労力が増すため、限定的なコミュニケーションにとどまりがちである。つまり、従来の授業形態は、対面授業による学習効果を十分に引き出せる環境とは言い難い。

そこで、学習者が授業に対する反応を出しやすく、かつ教師がその反応を把握しやすい環境を提供することが重要と考え、授業時間内における学習者と教師のやりとりを計算機で支援するためのシステム **inLec** を構築した。inLec は、学習者の積極的な授業参加を促し、授業内の学習者と教師の振る舞いを容易に確認できるようにすることを目的としたシステムである。

以下、3.2 節で inLec の概要について説明し、実際の講義での運用事例について、その結果と考察について 3.3 節で紹介する。

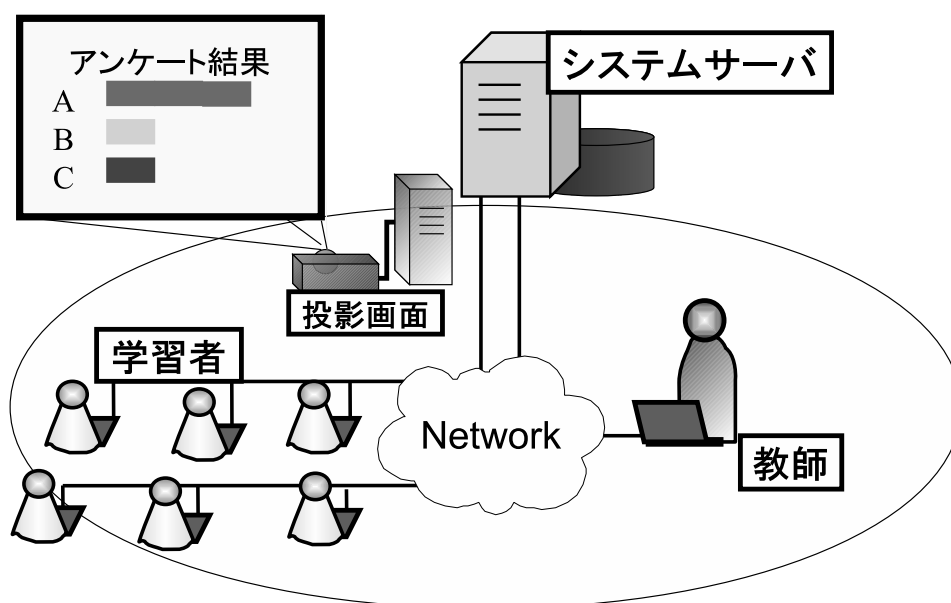


図 3.1: inLec 概略図

3.2 inLec の概要

inLec は、対面授業において、全学習者と教師（以下参加者）が授業中に利用する Web アプリケーションである。以下、inLec の運用環境と提供する機能の概要について説明する。

3.2.1 運用環境

inLec の概略図を図 3.1 に示す。inLec では全参加者が、ノートパソコンなど、Web ブラウザを利用できネットワークに接続可能な端末を講義室に持ち込む。参加者はそれぞれ、各自の端末からネットワークに接続し、Web ブラウザを通して inLec の各機能を利用しながら、授業に参加する。学習者と教師には、それぞれ学習者用、教師用のインタフェースが設けられている。

さらに、投影表示用の画面を出力する専用の計算機も用意する。これは、アンケート結果や出席状況など、inLec が取得した情報を、必要に応じて教師がプロジェクタを通して投影画面に表示するものである。投影画面も Web インタフェースによって実現しており、教師用インタフェースから制御可能である。

3.2.2 提供する機能

以下、inLec が提供する主要な機能について説明する。

簡易レスポンス機能

学習者用インタフェースには、**簡易レスポンスボタン**と**簡易レスポンス1行入力**が表示されている（図 3.2）。簡易レスポンスボタンには、「1」から「5」の数字と、「はい」「いいえ」の7つのボタンが用意されている。一方教師用インタフェースには、簡易レスポンス機能の質問入力欄が設けられている（図 3.3）。

教師は学習者に質問をする際に、手元の教師用インタフェースから質問文を入力する。入力した質問は投影画面に表示される。ただし質問文を空欄にすると、教師が口頭で直接学習者に質問をしたと仮定して「今言った質問に回答してください」がデフォルトの質問文として表示される。これにより、質問文の入力という操作を行わず、ワンクリックで学習者からの回答を募ることができる。

学習者は投影画面に表示された、あるいは口頭で問われた質問に対して、簡易レスポンスボタンのいずれかをクリックすることで回答する。どのボタンが何の回答にあたるかは、教師が口頭で伝えるものと仮定している。またその他の選択肢として、一行入力欄を利用することができる。回答の集計結果はリアルタイムで投影画面上に表示され、参加者全員が閲覧可能である（図 3.4）。また教師用インタフェースの“リセット”ボタンをクリックすると集計結果はクリアされ、次の質問を行うことができる。

操作を単純化することで、教師がその場で思いついた質問を頻繁に学習者に提示することができ、さらに集計結果をリアルタイムに確認できるため、教師が学習者全体の理解の傾向を素早く把握するツールとして利用できる。

いつでも返答機能

教師からの質問の有無に限らず、学習者から自発的に教師へ質問や意見を出せるインタフェースとして、**いつでも返答ボタン**を設けている。いつでも返答ボタンには、「わかったぞ!」「わからないぞ!」のボタンと一行入力欄を設けている（図 3.5）。学習者は授業中の任意のタイミングで、ボタンのクリックや文章入力により教師への意思表示が可能である。ボタンのクリックの集計結果は、リアルタイムに教師用インタフェースと投影画面に表示される（図 3.6, 3.7）。集計結果は過去1分以内に「わかったぞ!」「わからないぞ!」ボタンがクリックされた回数である。

簡易レスポンスボタン

1

2

3

4

5

はい

いいえ

簡易レスポンス1行入力

文章で回答

図 3.2: 簡易レスポンス機能インタフェース（学習者用）

超簡易レスポンスアナライザ

各生徒が一番最近に押した「簡易レスポンスボタン」を集計する機能です。

質問文

表示


自動更新:


投影表示:

選択式の回答を集計する

文章の回答を集計する

回答者はいません

リセット

図 3.3: 簡易レスポンス機能インタフェース（教師用）

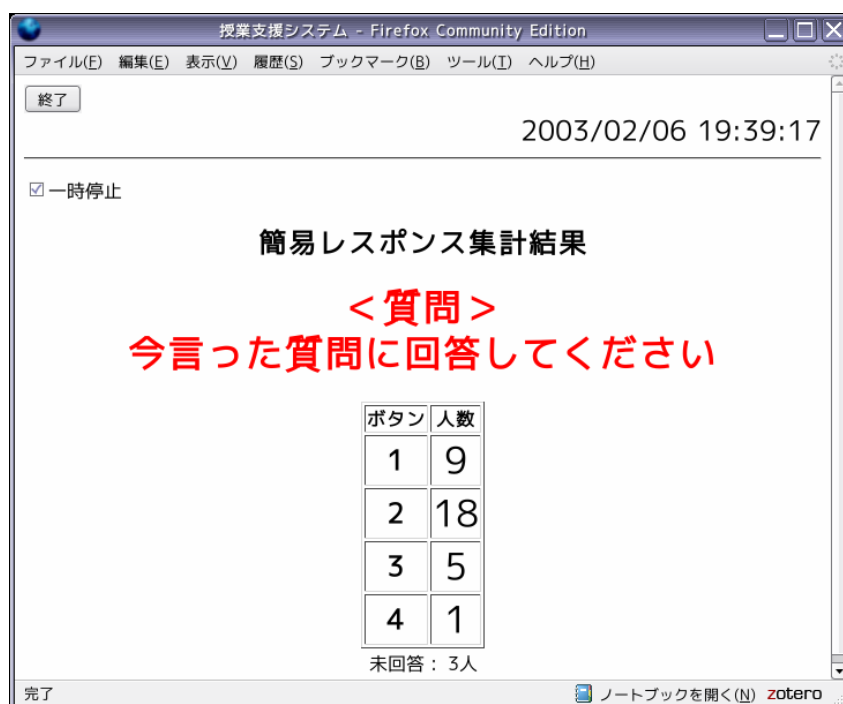


図 3.4: 投影画面：簡易レスポンス集計状況表示

本機能は学習者の意思表示を大まかに示すことが目的であり、厳密な人数のカウントが目的ではない。そのため、個人のボタンクリック回数にあえて制限は設けず、各学習者の意思表示の強さをクリック回数で表現できるようにした。また文章入力については、他人に知られたくない質問でも気軽にできるよう、教師用インタフェース上にのみ表示するようにした。さらに、いつでも返答ボタンのボタンクリックが10回以上になるか文章入力が行われた場合、教師用インタフェースから音が鳴る機能も実装し、教師が計算機から離れて講義を行っている時でも学習者からの意思表示に気づくようにした。

出欠管理機能

inLec はログインする際にユーザ認証を行うようになっている。また、学習者がログインできるのは、教師用インタフェースで「授業開始」操作を行ってから、「授業終了」操作を行うまでの間に限られる。inLec はユーザ認証できた学習者を出席と判断することで、当該授業に参加している学習者を特定する。

教師用インタフェースでは、学習者の出欠状況（ログイン状況）を閲覧する画面が設けられている（図 3.8）。出席状況はリアルタイムで更新され、教師用インタフェースで常時閲覧できるだ

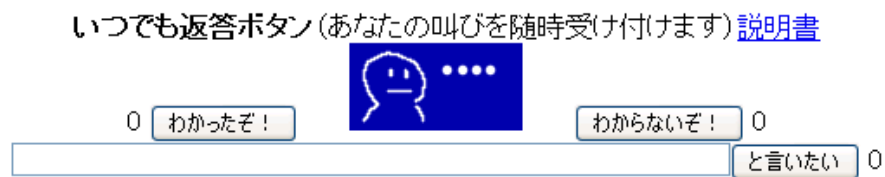


図 3.5: いつでも返答ボタン

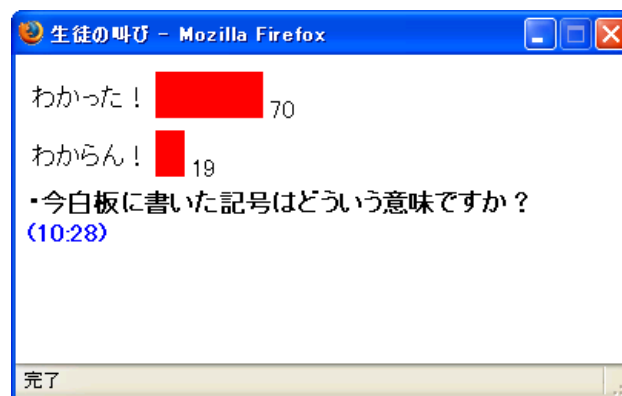


図 3.6: 教師用インタフェース：いつでも返答ボタン集計結果

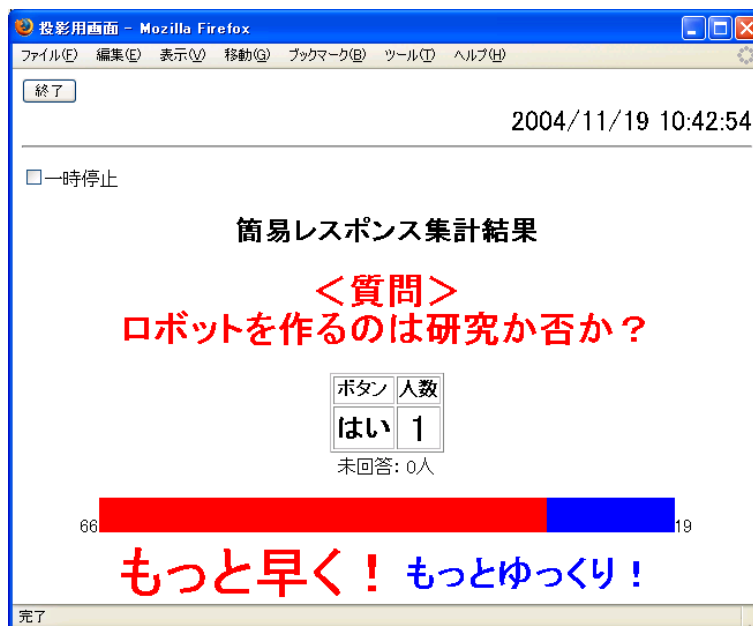


図 3.7: 投影画面：いつでも返答ボタン集計結果（画面下）

☐ 自動更新 (5秒ごと)

最新の出席状況を表示

出席者

てすとゆーざ 1 | てすとゆーざ 2 | てすとゆーざ 3 | テストユーザ 1 1

欠席者

テストユーザ 1 2 | テストユーザ 1 4

(11:57:30)

出席者と欠席者 の 出欠状況の投影画面表示

出欠情報をcsv形式で保存

図 3.8: 出欠確認画面

けでなく、投影画面に表示して出席状況を学習者に示すこともできる。

また、学習者のログイン画面には、座席番号を入力する欄も設けている。ログイン時に座席番号も入力することで、出席状況を単純なリストではなく、座席に対応した表にして表示することも可能である。ただし、授業を行う講義室の座席情報は、事前に管理者が登録しておく必要がある。

操作履歴解析機能

inLec は、利用者の全操作内容を、その操作が行われた時刻とともにログとして記録する。記録する操作内容は、inLec インタフェース上のボタンクリックと文章入力すべてが該当し、簡易レスポンス機能、いつでも返答機能でのクリックしたボタンや入力した文章の他、ログイン操作も含まれる。

この操作履歴を利用して、授業に関する様々な情報を提供することができる。操作履歴から抽出して提供できる情報としては以下のものがある。

- 1. 出欠表示** 授業開始から終了までの、出席者数の時間毎の推移を表示する。また、学習者ごとの各授業の出席時刻を一覧表示で確認でき、大幅に授業に遅刻した学習者のチェックも容易である。
- 2. 学習者の回答状況** 授業中の教師からの各質問について、質問を提出した時刻と、それに対する各学習者の回答とその回答時刻を一覧表示する。これを利用して、時刻推移による学習者の回答状況の変化をグラフ表示することもできる。さらに、いつでも返答機能を利用して学習者から質問が入力された時刻も一覧表示で確認できる。

3. 学習者の客観的情報と inLec 利用の関連 各学習者について、出席時刻や座席位置の情報と、各質問への回答時間やシステム操作回数を、表形式で表示する。これを利用して、出席時刻や座席位置とシステム利用状況の相関をチェックすることも可能である。

3.2.3 その他の機能

inLec ではその他の機能として、出席者の中からランダムに学習者の氏名を選択して投影画面に表示する**ルーレット機能**や、事前に登録した複数の質問のまとまりを提示し、学習者の回答結果を集計表示する**アンケート機能**も実装している。アンケート機能は主に授業前後の意見収集や、授業中に実施する小テストなどに用いることを想定している。質問内容は授業中に入力することも可能だが、基本的には授業前に準備することを想定した入力インターフェースとなっている。

ただし、3.3 節で述べる実践においては、いずれの機能もあまり利用しなかった。ルーレット機能については授業の流れ上利用する機会がなく、またアンケート機能は、事前準備の必要ない簡易レスポンス機能で十分代用できたことが理由である。なお、後に inLec からアンケート機能のみを抽出したシステムを別に構築し、学内アンケート用システムとして活用している [11]。

3.3 実践

著者はこれまで複数の授業科目において、inLec を利用した授業を実施した。

いずれの運用においても、授業後のアンケートにより、学習者からは「気軽に意見を出せる」「投影画面で出席や回答結果を確認できるのがよい」「従来の講義よりも退屈しない」などの評価を、また教師からは、「従来はあまり出てこなかったような学習者の意見を多く得ることができた」という評価を得た。inLec 運用の目標である、学習者の積極的な授業参加を促す効果が得られたものと考えられる。

inLec を利用した授業の諸情報を表 3.1 に示す。本節ではそのうち、同一科目で7回連続の inLec 利用を行った 2002 年度後期の“計算論”と、遠隔の学習者を含む授業を行った 2004 年後期の“特別講義”の実践結果について述べる。

3.3.1 実践事例：同一科目での inLec 継続利用

2002 年度後期の「計算論」では、同一の授業において7回連続で inLec を利用した学習を行った。当授業では7回とも同一の教員が講義を行い、著者は講義室内で inLec 運用のサポートを行っ

表 3.1: inLec 利用授業の諸情報

実施機関	鳥取環境大学 環境情報学部情報システム学科			鳥取県立青谷高校 総合学科
学期	2001 年度後期	2002 年度後期		2004 年度後期
科目名	計算機 アーキテクチャ	計算論	計算機 アーキテクチャ	特別講義 「研究とは何か」
対象学年	1 年生	2 年生	1 年生 (一部 2 年生含む)	2 年生
実施日	12/18	10/3,10,17,24,31 11/7,14	11/19,26	9/10,17 10/15

た。また、講義の様子はビデオカメラで撮影し、授業後の inLec 評価時の資料とした。

運用においては、学生は各自教室に持参した自分のノートパソコンを、教室内の各机にある LAN コネクタに接続した状態で、Web ブラウザに表示される inLec の学習者用インタフェースを利用しながら授業を受講した。使用する Web ブラウザは特に指定しなかったが、大半の学生は、当時大学推奨ブラウザであった Netscape を用いて接続した。一部、学校指定以外のノート PC を用いたり、Internet Explorer や Opera など Netscape 以外の Web ブラウザを利用する学生もいたが、それに起因する問題は特に発生しなかった。

以下、inLec 操作履歴をもとに、授業における inLec 運用結果について考察を行う。

利用者のレスポンス総数

まず、授業中における学習者からのレスポンスの評価指標として、inLec インタフェースへのクリック総数を確認する。学習者インタフェース上の操作可能なボタンは全て、教師あるいは授業内容に対するレスポンスとして用いられるため、クリック数を学習者のアクション回数と見ることが出来る。

表 3.2 に各授業の学習者と教師の総クリック数を示す。クリック総数は、授業の 1 回目と 2 回目以降の間で大きな差があることが確認できた。特に 1 回目の授業では、簡易レスポンスボタンやいつでも返答ボタンを数秒の間に何度もクリックするという履歴が非常に多く確認されたことから、恐らく学生にとって inLec 自体が珍しく、興味本位で必要以上に操作を試したということが推

表 3.2: inLec 総クリック数：鳥取環境大学（日付は全て 2002 年）

計算論				
日付	出席者	操作回数		
		学生	教師	学生/教師
10/03	65	3617	91	39.7
10/10	57	882	34	25.9
10/17	58	798	30	26.6
10/24	52	766	37	20.1
10/31	55	543	13	41.8
11/07	58	561	23	24.8
11/14	48	485	24	20.2

測される。ただし、授業ごとの教師の操作回数と比較すると、学生の操作回数は教師の操作回数にほぼ比例しており (単相関係数=0.97)、さらに簡易レスポンスボタンについて、授業毎に教師が提示した1質問あたりの学習者のボタン押下回数を計算すると、2回目以降の授業でも大きく減少していない (表 3.3) ことから、inLec によって、学習者と教師のやり取りは安定して行うことができたと考えられる。

簡易レスポンス機能の利用状況

簡易レスポンス機能による質問とその回答の推移のうち、回答の推移に顕著な変化がある場合についてビデオで授業の状況を確認した結果、多くの場合教師のヒントや指示など、教師の言葉が影響していることが確認できた。図 3.9 の例では、ある問題に対する学習者の解答が、最初は「はい」のみだったのが、1分後その状況を見ていた教師がヒントを出した後、「いいえ」の解答が増加し、その状況を見て「はい」から「いいえ」に乗り換える学習者も現れ、結果的に全員が「いいえ」と解答する結果となった。

簡易レスポンスの回答推移データを活用することで、どのような教師のヒントや指示が学生に回答ボタンを押させるなどの作用をもたらすか、さらには理解に役だったかなどを、授業後にチェックする手段として有用であると考えられる。ただし簡易レスポンス機能は質問内容を口頭で行う可能性があるため、回答推移データを有効活用するには、授業のビデオ映像と連動してデータを

表 3.3: 簡易レスポンスボタンのクリック数

日付	質問数	クリック数	クリック数/質問
10/3	15	1183	78.9
10/10	13	504	38.8
10/17	11	527	47.9
10/24	11	488	44.4
10/31	1	222	222.0
11/07	1	65	65.0
11/14	1	57	57.0

表 3.4: いつでも返答ボタン利用回数（一行入力欄は 10/24 から提供）

日付	ボタン	文章入力
10/03	127	-
10/10	86	-
10/17	7	-
10/24	33	1
10/30	100	8
11/07	25	5
11/14	3	3

確認できるような工夫が必要だと考えられる。

いつでも返答ボタンの利用状況

学習者によるいつでも返答機能の利用については、ボタンによる意思表示はある程度あったものの、文章入力による意思表示は、教師が利用を促した時以外はあまり使用されなかった（表 3.4）。アンケートによると一部の学生から、いつでも返答ボタンについて「利用に勇気がある」という意見もあった。ただし教師からは、いつでも返答機能による学生からの質問や意見には、教師にとって非常に有用なものが多かった、との意見を得た。

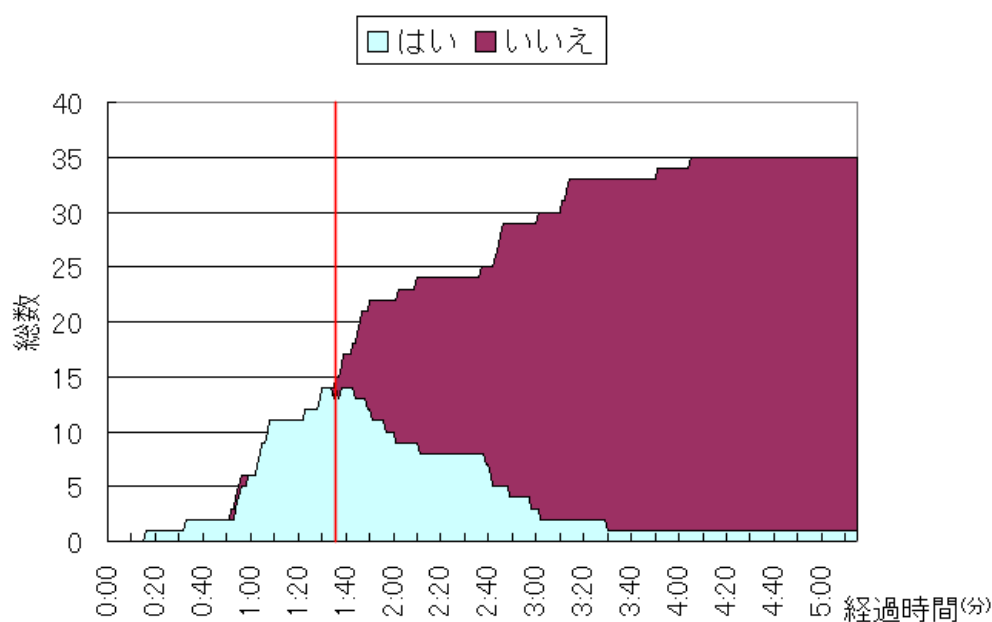


図 3.9: 回答の推移 (経過時間 1 分 35 秒あたりから「はい」と「いいえ」の回答数が逆転)

座席と利用回数の関係

授業参加に熱心な学生は前方に着席するだろうという仮説を立て、これを検証してみた。「計算論」の7回の講義のうち、6回以上出席している学生51人(10人:6回出席, 41人:7回出席)に対して、最も多い着席位置を調べ、座席の列ごとの学生の操作回数の平均をとった。その結果、前列の学生以上に後列の学生が意欲的にinLecを利用していることが確認できた(図3.10)。

この結果はあくまで当時の学生に見られた傾向を示すものであり、一般的な傾向かどうかは不明である。しかし学習者の着席位置とinLecによるレスポンスに相関が見られないという事実は、学習者の授業参加への意欲は教師-学習者間の物理的な距離とは本来無関係であり、inLecは物理的な距離を超えて学習者の授業への参加意欲をより正確に把握できるツールであることを示していると言える。

3.3.2 実践事例：遠隔学習者を含む環境でのinLec利用

鳥取県立青谷高等学校(以下青谷高校)では、2004年9月-10月に、「研究とは何か」をテーマとした全2回の特別授業を実施した。授業は2年生4クラスを対象とし、1,2組, 3,4組の2つの合同クラスに分かれてそれぞれ別日に授業を実施した。そのうち1,2組は1回目の授業のみ、3,4

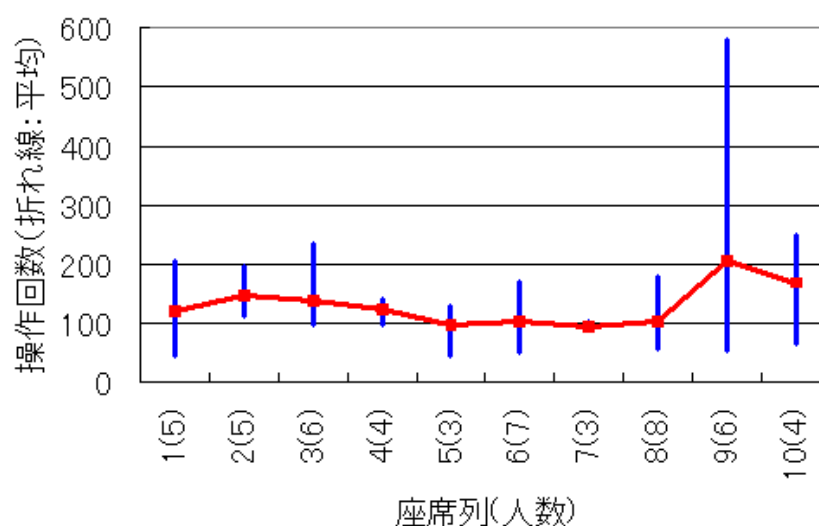


図 3.10: 座席列とクリック数の関係 (折れ線：平均, ひげ：分散)

表 3.5: 青谷高校の inLec 利用授業概要 (日付は全て 2004 年)

日付	クラス	講義	対面室	遠隔室
9/10	1,2 組	第 1 回	1 組	2 組
9/17	3,4 組	第 1 回	3 組	4 組
10/8	1,2 組	第 2 回	(inLec 不使用)	
10/15	3,4 組	第 2 回	4 組	3 組

組は 1,2 回両方の授業で inLec を利用した。授業の諸情報を表 3.5 に示す。

青谷高校では、生徒全員が計算機を操作できる環境として、固定端末が設置された演習室を利用することになった。しかし演習室の収容人数の関係から、2つの演習室に生徒が分かれざるを得ないという問題が発生した。そこで、教師は片方の演習室で講義を行い、もう一方の演習室とは学内 LAN と DVTS を利用した映像・音声のリアルタイム送受信を行うことにした。つまり、直接教師と生徒が対面する教室（対面室）と、教師と生徒が映像を介して対面する教室（遠隔室）に対して同時に授業を実施する形態になり、遠隔室の生徒にとっては、inLec が教師にレスポンスを投げる唯一の手段となる。ただし、元々講義形式の授業では、inLec を通さない生徒からのレスポンスは期待しにくいと考えていたため、映像・音声の送受信によって教師からの口頭伝達手段を確保することで、対面室と遠隔室で生徒-教師間のコミュニケーションの相違はそれほど大きくなりたらないのではないかと考えた。

表 3.6: inLec 総クリック数：青谷高校（日付は全て 2004 年，*:遠隔室）

日付	クラス	出席者	操作回数		
			生徒	教師	生徒/教師
9/10	1	38	6917	94	73.5
	2	38	4095*		43.6
9/17	3	39	5183	74	70.0
	4	39	4279*		57.8
10/15	3	37	14149*	63	224.6
	4	34	2695		42.8

授業の講師は著者が担当した。またアシスタントに鳥取環境大学の学生が 1 名、映像送受信用のカメラの操作と、技術上のトラブルへのサポート作業を行った。

以下 3.3.1 節と同様に、inLec 運用結果について考察を行う。

利用者のレスポンス総数

授業ごとの inLec インタフェースに対する総クリック数は、表 3.6 の通りとなった。いずれの組も、鳥取環境大学での実践結果と比べて参加人数が半分程度であるのに対し、総クリック数が非常に多い。また 3,4 組については、2 回目の授業でも鳥取環境大学の 1 回目の授業を上回る総クリック数を記録し、特に 3 組については、1 回目より 2 回目の授業の方が総クリック数が多いという傾向を示した。具体的には、第 1 回より第 2 回の方がクリック数が増加している生徒は、3 組が 37 人中 27 人で、4 組が 34 人中 9 人であった。

表 3.6 の結果からは、第 1 回の授業ではいずれも対面室より遠隔室の方がクリック数は少ないものの、第 2 回の授業で遠隔室である 3 組の総クリック数が、全クラスの中でも最高の 14000 回以上にのぼっており、対面室と遠隔室での顕著な差は確認できなかった。

簡易レスポンスボタンの利用状況

図 3.11 は、1 回目の授業において、教師が簡易レスポンス機能を用いて同一の質問した際の、各クラスの返答率を示す。いずれの質問でも、生徒からは高い回答率を得られた。2 組では、初めは回答者が少なかったが、授業が進むにつれて徐々に増えていった。また 3,4 組の生徒は、授業開始

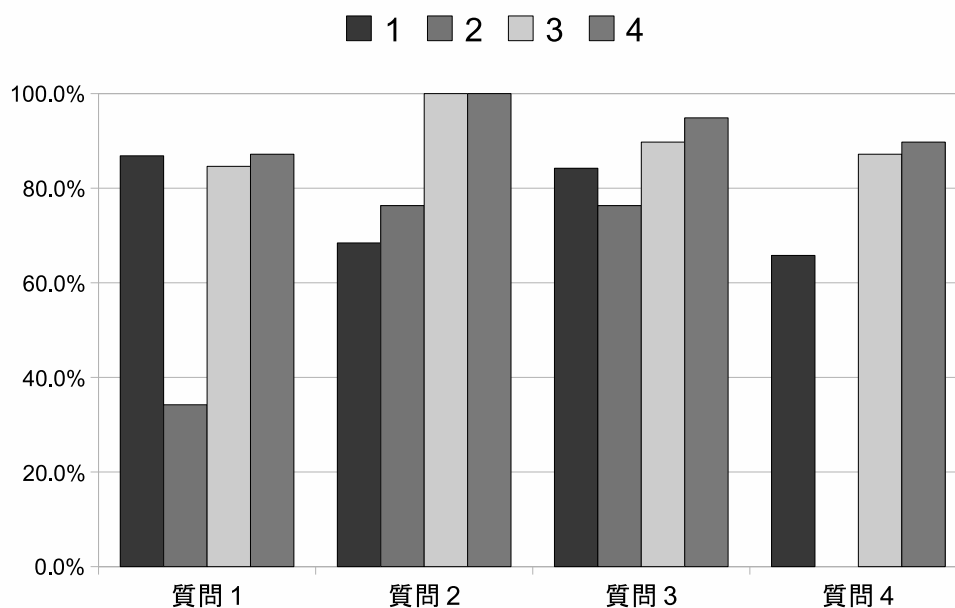


図 3.11: 簡易レスポンス機能による返答率：青谷高校（凡例の数字はクラス名）

から終了まで常に高い回答率を維持した。質問 4 に対する 2 組の回答率が 0% であるが、これは機器トラブルにより対面室の映像と音声が届いておらず、遠隔室の生徒が教師からの質問の存在を認識していなかったのが原因であった。

いつでも返答ボタンの利用状況

いつでも返答ボタンによる生徒からのレスポンスは、全体にわたって積極的に行われた（表 3.7）。

鳥取環境大学における実践結果との違いとしては、一行入力欄の積極的な活用が挙げられる。特に 3, 4 組では、1 回目の授業でも一行入力欄によるレスポンスが多かったが、2 回目の授業ではさらに増加している。実際には、生徒が入力した文章の大半は、教師へのプライベートな質問や雑談に相当するコメントなど授業と関係のない内容であったが、それらにも授業に支障が出ない範囲で積極的に応答したことで、書き込みに対する生徒の心理的な敷居が下がっていったのではないかと考えられる。

授業後アンケート

授業終了後、全クラスの生徒に対して inLec 利用に関するアンケートを実施した。

表 3.7: いつでも返答ボタンクリック数:青谷高校 (*:遠隔室)

	第1回		第2回	
	ボタン	文章入力	ボタン	文章入力
1組	5664	18	-	-
2組	2881*	7*	-	-
3組	3804	69	13338*	155*
4組	3320*	19*	2219	48

inLecでの意見の出しやすさについての5段階評価(図3.12)では、総じて、1,2組より3,4組の方が評価が高い結果が出た。3,4組いずれも、生徒の約8割がinLecでの意見の出しやすさに「強く」あるいは「少し」という肯定的な評価を示した。特に4組は、総クリック数で見ると3組に比べて少ないにもかかわらず、意見を出しやすいと「強く」考える生徒の割合が3組より多かった。操作履歴から、3組は元々活発に盛り上がりやすい性格を持つクラスだったと推測されるため、4組がその影響を受けてinLec活用を積極的に行うようになり、それが3組以上に好印象を持った原因ではないかと考えられる。

逆に、1回だけinLecを利用した1,2組については、1組は6割強の生徒から肯定的な評価を得られた一方、2組の肯定的評価は5割を切っている。また、1,2組に対する「inLecと通常講義のどちらがよかったか」という質問に対しては、1組は80%以上の生徒がinLec使用授業がよかったと回答しているのと比べ、2組のinLecに対する評価は明らかに低い(図3.13)。同じ授業を受けたにもかかわらず2組の生徒のみinLecへの印象が悪くなった原因として、対面室でinLecを利用する機会がなかったこと、さらに1回目のinLec利用時には映像の送受信で若干不具合が発生したことが影響していると考えられる。

対面室と遠隔室を両方体験した3,4組に対する「対面と遠隔のどちらがよかったか」という質問に対しては、遠隔室よりも対面室の方がよいと回答する生徒が目立った(図3.14)。生徒のコメントでは、対面室の方がよいとする主な理由として、遠隔室では映像を介することで教師の姿や声が把握しにくくなってしまうことが挙げられていた。一方、従来と異なる授業形態そのものへ関心が深かった生徒は遠隔室がよいと回答し、また映像を介する授業形式に特に違和感を覚えなかった生徒は「どちらでもよい」を選択する傾向にあることがコメントより確認された。遠隔講義において学習者の意欲を維持することの困難さはよく指摘される課題であるが[33, 35]、少なくともinLecによって学習者からのレスポンス手段を提供するだけでなく、教師から生徒への情

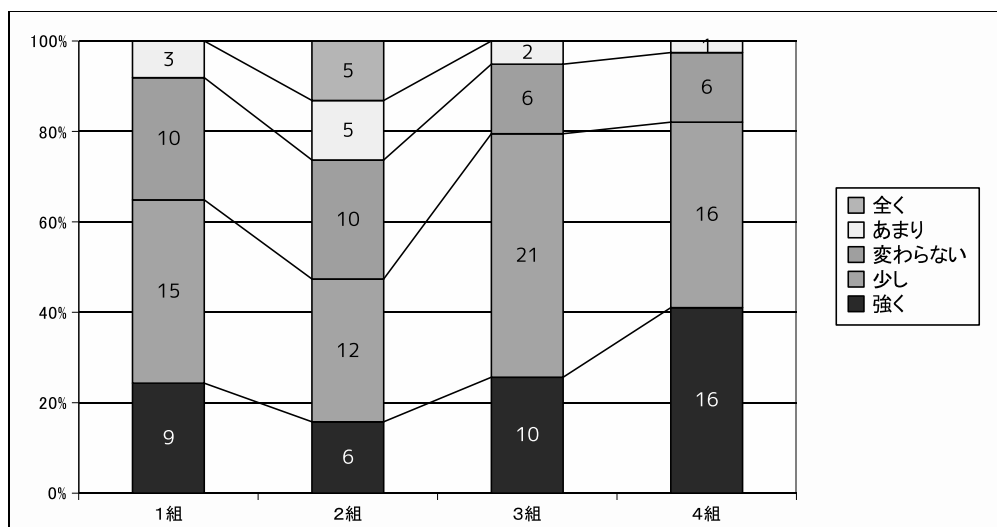


図 3.12: 質問「inLec では自分の意見を出しやすいか？」への回答

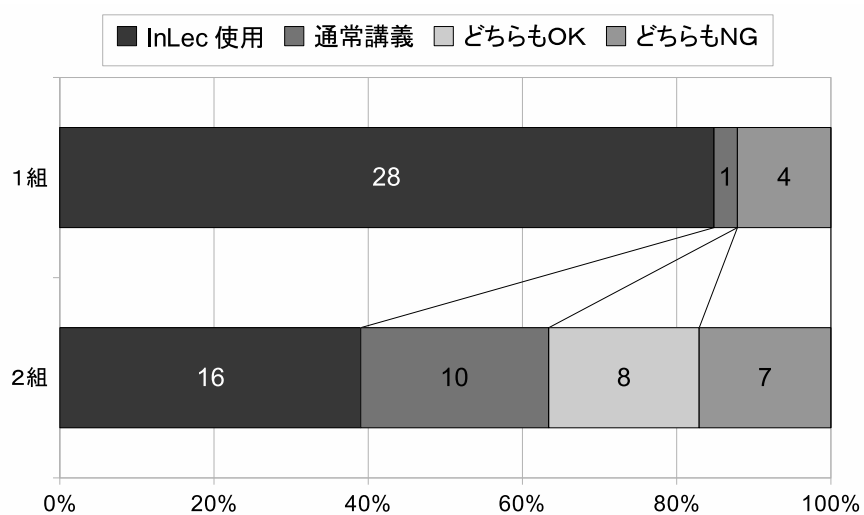


図 3.13: 質問「inLec と通常講義のどちらがよかったか？」への回答

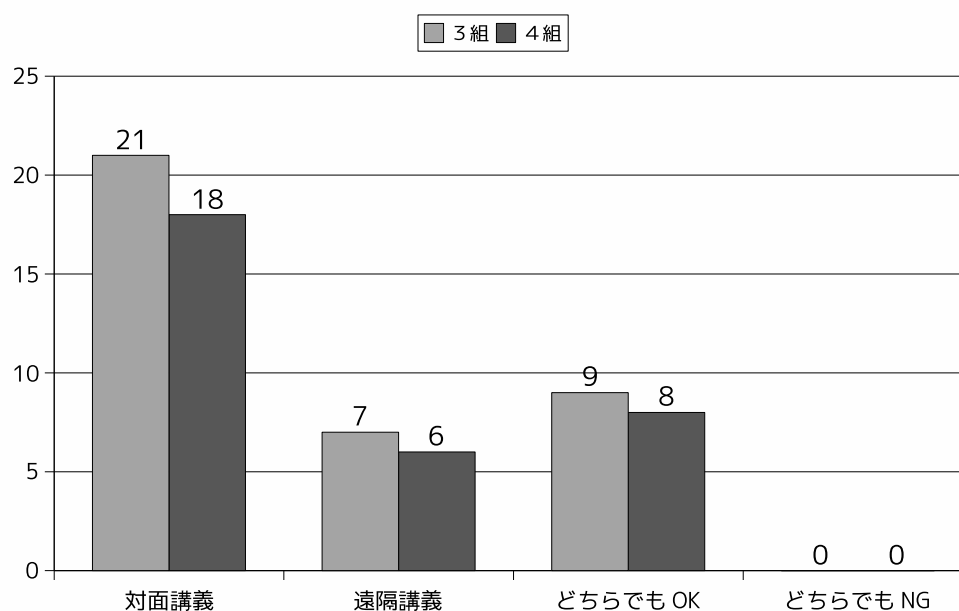


図 3.14: 質問「対面と遠隔のどちらの授業がよかった？」への回答

報伝達の違和感をどれだけ解消するかが重要であると言える。

3.4 おわりに

本章では、著者が構築した授業内の学習者と教師のやりとりを支援するシステム inLec について紹介し、実際の授業における inLec の運用結果とその評価について述べた。様々な学習者を対象とした運用において、総じて inLec による学習者からのレスポンスが多かったこと、さらに継続的な運用によっても学習者から一定数のレスポンスが持続されたことから、単純に新しいツールに対する興味からではなく、inLec によって授業中の教師とのやりとりが容易になるという効果が、学習者間で確実に認識されたものと考えられる。

今後の課題としては、遠隔講義での inLec 活用方法の検討がある。もともと inLec は対面講義での利用を想定して設計したものではあるが、今後ますます学習環境の多様化が期待される状況において、対面授業だけでなく遠隔授業においても、inLec が目指す“学習者と教師の授業内コミュニケーションの活発化”の実現は必須と言える。遠隔授業による臨場感の低下を補うため、学習者から教師へのレスポンス手段提供だけでなく、教師から学習者へのアクションを支援する手段を実現する必要がある、inLec の機能としてどう実現していくか、検討を進める必要がある。

第4章 アルゴリズム学習における間違い探し形式の演習課題を自動生成する手法の提案と評価

本章では、科目特有の課題についての研究として、アルゴリズム学習のための演習問題の自動生成手法を提案する。ここでの演習問題とは、あるアルゴリズムを実装したソースコードに誤りを含ませたものに対して、誤りの修正及び考察を行う形式を対象としている。演習問題の自動生成においては、演習を通じてアルゴリズムの理解を深められる誤りを生成できることが必須である。そこで著者は、正しいソースコードの適切な場所に適切な誤りを自動挿入するシステムを試作した。また、提案手法による誤り挿入の有用性について評価を行い、その結果をもとに提案手法を改善した結果、手作業による誤り挿入と同等程度の有用性を得られることを確認した。

4.1 はじめに

アルゴリズムの学習は、情報科学の学習において重要な分野の一つである [36]。アルゴリズムの講義では、学習者の理解度向上を狙った取り組みとして、講義で扱うアルゴリズムをプログラミング言語で実装する形式の演習が行われている。

コーディングを正しく行うためには、まず実装しようとするアルゴリズムの内容を十分に理解している必要がある。つまり、アルゴリズムの動作を再確認する契機となる点で、プログラミングはアルゴリズムの理解を深めるのに効果的な演習形式と考えられる。一方この演習形式は、実装言語の仕様やソースコードのデバッグなどに学習者の注意が向きやすく、演習本来の目的であるアルゴリズムの理解に学習者を集中させにくいという問題点がある [37, 38, 39]。

この問題点に対するアプローチとして、著者らはソースコードにおける**間違い探し演習**を提案する。本研究で提案する間違い探し演習の手順は以下の通りである。

1. 教師は、あるアルゴリズムを実装したソースコードを学習者に提示する。ただしソースコードには、アルゴリズムと矛盾する誤り箇所が1つ以上含まれている。

2. 学習者はソースコードに含まれている全ての誤りを発見し、実装すべきアルゴリズムと比較してどのような矛盾が生じているかを考察する.
3. 学習者は、アルゴリズムに合致するようソースコードの誤りを修正する.

間違い探し演習では、学習者はソースコード全体を自分で実装する必要はない. よって、構文エラーなどアルゴリズムの理解と関係のない事柄に学習者が注意を払う必要が減るので、教師はアルゴリズムの理解に焦点を絞った演習を行うことができる. また、学習者の解答は、問題として提示したソースコードの一部を書き換えただけなので、教師にとっては一から作成したソースコードをチェックする場合に比べて、採点作業の時間的負担が軽減する効果も期待できる. さらにこの効果は、一度に多数多種の演習問題を扱う演習を教師が実施しやすくなるという期待にもつながる.

本章では、間違い探し演習の問題を自動生成することを目標として、アルゴリズムの理解を深めるという演習目的に適した誤りをソースコード中に自動挿入する手法を提案する. 以下、4.2 節では本研究が想定する間違い探し演習の詳細及び問題作成の自動化について検討する. また、4.3 節では具体的な問題作成の自動化手法を紹介し、4.4 節では提案手法の実装について概要を述べる. 4.5 節では実際の授業で間違い探し演習を実施した結果を述べる. 4.6 節では提案手法の評価を行う.

4.2 演習形式の検討

4.2.1 演習問題の形式

間違い探し演習は、講義で扱ったアルゴリズムの理解度向上がその目的である. つまり学習者は演習を通して、学習したアルゴリズムの具体的な手順を振り返り、知識を再確認することが求められる.

そこで間違い探し演習では、ソースコードに含まれるアルゴリズムについて、学習者に“アルゴリズム名”だけを情報として提示する. ソースコードに含まれるアルゴリズムとの矛盾を発見、修正するために、学習者は提示されたアルゴリズムの具体的な手順を自身の知識から思い出し、そのアルゴリズムの手順がどのようにソースコード上に実現されているかを把握する必要がある. つまり学習者は演習問題の解答時は常にアルゴリズムの動作を意識することになり、結果アルゴリズムの知識を再確認する目的が達せられると考えられる.

演習問題としては、アルゴリズム上の誤りが存在するソースコード中の該当箇所、該当箇所の修正結果を解答として記述する問題を学習者に与える。コーディングではなくアルゴリズムの動作に学習者の意識を集中させるため、学習者は演習中にソースコードの実行環境を使えないこととする。また、修正前後のソースコードについてアルゴリズム上の差異を説明する問題を提示する。ソースコードの修正がアルゴリズムに及ぼした影響を考察させることで、学習者にアルゴリズムの動作へ意識を向けさせることを狙う。

4.2.2 演習に適した誤り

間違い探し演習においてソースコードに含まれる誤りは、その発見と修正によって学習者の正しいアルゴリズムの理解へつながるものでなければならない。そこで間違い探し演習では、以下の2つの条件を満たすソースコード中の誤りを**演習に適した誤り**と定義する。(1) その誤りを含むソースコードは、問題文で指定されたアルゴリズムの手順と矛盾した動作を行う。(2) 学習したアルゴリズムの手順を覚えていなければ、誤りを発見あるいは修正することは困難である。

構文エラーは、その発見と修正の作業がプログラミングの知識のみで行えるため、アルゴリズムの学習という目的に適さない。よって演習に適した誤りではないと定義する。また、 $i = i$ （代入文の左辺と右辺が同じ）など、コーディング自体に不自然さが含まれている誤りの場合、学習者がアルゴリズムとの矛盾を意識せずに誤りを発見、修正できてしまう可能性がある。この場合もアルゴリズムの演習目的に沿わないため、演習に適した誤りではないと判断する。

4.2.3 演習問題の自動作成

間違い探し演習の問題を手作業で作成する場合、教師はあらかじめ学習させたいアルゴリズムを実装したソースコードを用意し、そのソースコードの一部を修正して誤りを含んだ形にするという作業を行う。このとき、ソースコードに含ませる誤りが演習に適したものかどうかを検討しながら慎重に問題作成を行う必要があるため、教師にとって問題作成は負担が大きい作業となる。

そこで本研究は、間違い探し演習の問題作成を容易に行う手段を提供すること、つまり与えられたソースコードから、演習に適した誤りを含んだソースコードを自動的に生成する手法の構築を目標とする。これにより、教師の問題作成における負担は軽減し、間違い探し演習の導入が容易になることが期待できる。

さらに演習問題作成の自動化によって、一つのソースコードから複数の演習問題を生成したり、一つのアルゴリズムに対して実装の異なる多数のソースコードから演習問題を生成したりといった作業が容易となる。その結果、例えば学習者ごとに異なった演習問題を提示することで他人の解答を書き写す不正の防止を図ったり、学習者ごとの理解に合わせて難易度の異なる問題を提示したりなど、演習方式の多様化が期待できる。

4.2.4 関連研究

アルゴリズムの設計を通じた理解支援を目的とした研究には、アルゴリズム設計支援ツールである RAPTOR[39] や JPADet[37], SFC[40] などがある。学習者が構文エラーなどに気を取られずに学習を行う環境を実現するという目標は本研究と同じであるが、上記の研究はいずれも直感的なアルゴリズム設計を可能とする手段として、独自のグラフィカルなアルゴリズム設計環境を提供するというアプローチを取っている。一方本研究は、学習者が既に経験している既存のプログラミング言語をそのまま活用できるという点が異なる。

演習問題の自動生成に関する研究としては、ソースコードから空欄補充問題を自動生成する文献 [41] や、正しいソースコードを任意のブロックに分割してパズル化した問題を自動生成する文献 [42] がある。空欄補充問題は学習者にとって考えるべき箇所が空欄として明確にわかるのに対し、間違い探し演習の場合は注意深くソースコード全体を読んで修正すべき箇所がどこにあるかを探し出す必要がある。またパズル化されたソースコードを組み立てる問題は、間違い探し演習と異なり基本的にソースコード上の誤りは存在せず、ブロック間のつながりに学習者の注意を向ける。いずれも本研究とは、学習者に求める演習問題への取り組み方が大きく異なる。また難易度に応じて選択式/空欄補充/誤り訂正問題を自動生成する [43] は、問題作成者が予め入力した誤りの内容に置き換えることで誤り訂正問題を生成する手法を用いており、誤りの内容自体も自動生成する本研究とはアプローチが異なる。

4.3 誤り自動挿入の手法

本節では、本研究で提案する、演習に適した誤りが含まれたソースコードを自動生成する手法について述べる。提案手法では、与えられたソースコードをもとに、(1) ソースコード中の誤り挿入対象の決定、(2) 挿入対象に適する誤りへの置換という手順で、誤りを含んだソースコードの生成を実現する。以下、各手順について詳しく説明する。

なお本章では、C 言語のサブセットで記述されたソースコードを対象として説明および評価を行う。

4.3.1 誤り挿入対象の特定

どのような誤りが演習に適しているかはアルゴリズムごとに異なるため、ソースコードで利用されているアルゴリズムにそって誤り挿入対象を決定する必要がある。ここで誤り挿入対象とは誤り挿入によるソースコード中の変化する箇所のことであり、ソースコード中の変数、文や式だけでなく、制御構文など複数の式の集合も含む。しかし、様々なアルゴリズムのそれぞれに対して演習に適した誤りを検討するアプローチは汎用性に欠ける。

そこで本研究では、個々のアルゴリズムがベースとする**アルゴリズム設計パラダイム**（以下パラダイム）の情報をパラメータとして与え、これに対応した誤り挿入対象を決定するアプローチを考えた。パラダイムとは、アルゴリズムの設計時に用いられる枠組みであり、アルゴリズムの多くは何らかのパラダイムを利用して設計されている。つまり、多くのアルゴリズムに共通して使われるパラダイムについて誤り挿入対象の決定方法を定義しておけば、より汎用性の高い演習問題の自動生成が可能になると考えた。

提案手法では、アルゴリズムの講義において扱われることの多い [44]、分割統治法、再帰、動的計画法、貪欲法の 4 つのパラダイムについて、パラダイムの特徴を分析し、誤り挿入対象の決定法を検討した。また、いずれにも該当しないパラダイムについても誤り挿入対象の決定法を検討した。以下、それぞれのパラダイムにおける誤り挿入対象決定法の詳細について述べる。

パラダイム 1: 分割統治法

分割統治法は、問題を小問題に分割して、各小問題の解を再帰的に求め、それらを統合することで最終的に問題を解こうとするパラダイムであり、全体の枠組みは次のようになる。

- [計算] 問題の規模が小さい場合は直接解く
- 問題の規模が大きい場合は
 - － [分割] 問題をいくつかの小問題に分割する
 - － [再帰] 各小問題を再帰的に解く
 - － [統合] 各小問題の解を基に解を求める

```

1  #define N 10
2  int a[N] = { RANDOM N };
3  /* each element is a random integer from 0 to N */
4
5  void merge(int l[], int r[], int a[],
6            int llen, int rlen, int len){
7  /* snip */
8  /* merge two sorted arrays l[llen] and r[rlen] */
9  /* to make an array a[len] (len = llen+rlen) */
10 }
11
12 void mergeSort(int a[], int len){
13     int i;
14     int mid;
15     int left[len], right[len];
16     if(len > 1){
17         mid = len / 2;
18         for(i=0;i<mid;i++) left[i]=a[i];
19         for(i=0;i<len-mid;i++) right[i]=a[mid+i];
20         mergeSort(left, mid);
21         mergeSort(right, len-mid);
22         merge(left, right, a, mid, len-mid, len);
23     }
24 }
25
26 int main(){
27     int i;
28     mergeSort(a, N);
29 }

```

図 4.1: マージソートを実装したソースコードの例

以上の枠組みに対応する構文がアルゴリズム上重要な箇所であり、ソースコードにおける (a) 再帰呼出を含む文や式、(b) 再帰呼出の引数に使われている変数の値を更新する文、(c) 再帰呼出より後に実行される関数呼出文や return 文、(d)(a) の実行の有無を決定する分岐命令及びその条件式を誤り挿入対象の候補とする。(a) は「分割」に、(b) は「分割」「再帰」、(c) は「統合」、(d) は「計算」にそれぞれ該当する。

分割統治法に基づくアルゴリズムの実装例として、マージソートのソースコードを図 4.1 に示す。ここでは、再帰呼出を含む文・式 (図 4.1 の A)、再帰呼出の引数に用いられる変数 (left, right, mid, len) について、これらの変数を更新する文 (図 4.1 の B)、再帰呼出の後に実行される関数呼出文 (図 4.1 の C)、再帰呼出文の実行の有無を決定する分岐命令 (図 4.1 の D) を、誤り挿入対象の候補とする。

パラダイム 2: 再帰

再帰を用いたアルゴリズムでは、再帰呼出文及び再帰呼出文を実行するかを決める分岐命令がアルゴリズム上重要な場所であるので、ソースコードにおける (a) 再帰呼出を含む文や式、(b) 再帰呼出の引数に使われている変数の値を更新する文、(c)(a) の実行の有無を決定する分岐命令及びその条件式を誤り挿入対象の候補とする。

パラダイム 3: 動的計画法

動的計画法は分割統治法と同様に問題を小問題に分割して解くパラダイムである。ただし、分割統治法と異なり、小問題の解を記録する表を用意して、上位の問題を解くときにはその表を参照することで計算量を減らす。

動的計画法では、表の役割をする変数（以下表変数）が重要な役割を持つ。つまり、表変数を更新する文に影響する処理がアルゴリズム上重要な場所である。ソースコード中で表変数は通常配列で表現され、また反復処理を行うための初期化処理を行うことが多い。そこでソースコード中で最初に現れる、配列へ定数を代入する文を検出し、その配列を表変数と判断する。その上で、(a) 表変数への代入文、(b)(a) の文中に現れる変数の値を更新する文、(c)(a) の実行の有無を決定する分岐命令およびその条件式を誤り挿入対象の候補とする。

パラダイム 4: 貪欲法

貪欲法は局所的に最良の選択を繰り返すことによって最終的に大域的な解が得られるという発想に基づくパラダイムである。貪欲法の場合は、局所解の導出に該当する処理がアルゴリズム上重要な場所である。

そこで、ソースコードにおける繰り返し文（while 文、for 文）に着目し、(a) 繰り返し文中で、分岐命令によって実行の有無が決定される代入文、(b)(a) の右辺に現れる変数の値を更新する文、(c)(a) の実行の有無を決定する分岐命令及びその条件式を誤り挿入対象の候補とする。

パラダイム 5: その他

アルゴリズムの中には、特定のパラダイムに基づいて設計されていないものや、今回は考慮していないパラダイムに基づき設計されたものが存在する。その場合はアルゴリズムの全般に共通する動作決定の要素を、アルゴリズム上重要な場所であるとする。

表 4.1: 誤り挿入対象と誤りパターンの対応表

誤り挿入対象の構文	挿入可能な誤りパターン
定数 c	定数 $0, c + 1, c - 1$
純変数	+1 または -1 の追加; 別変数への置換
算術演算子 $+$, $-$	$+\rightarrow-$, $-\rightarrow+$
論理演算子	別の論理演算子へ変換
算術二項演算 (e.g., $A + B$)	一方の項を削除 (e.g., A)
論理二項演算 (e.g., $A \&\& B$)	一方の項を削除 (e.g., B)
関数呼出文	文を削除
制御構文 (e.g., <code>if (S) A else B</code>)	条件式を削除 (e.g., A)

具体的には、分岐、繰り返し、関数呼出の処理に注目し、ソースコードにおいてこれらに該当する条件式や分岐命令文を、誤り挿入対象の候補とする。ただし for 文は、4.2.2 節の性質 (2) を満たさない誤りを生成しやすいことが経験的に判明したため、誤り挿入対象の候補とはしない。

4.3.2 誤りパターン挿入

誤り挿入対象の候補を全て抽出した後は、予め与えられた個数の候補をランダムに選択し、その位置に実際に誤りを挿入する。具体的には、特定の構文パターンごとに対応する**誤りパターン**を予め定めておく。そして誤り挿入対象に存在する構文形式を確認し、その構文に対応する誤りパターンを適用する。構文と誤りパターンの対応表の主な項目を表 4.1 に示す。適用可能な誤りパターンが複数ある場合は、いずれか一つをランダムに選択する。

例として、誤り挿入対象が代入文 “ $i=j+1$ ” であった場合、含まれる構文は “変数 i ”, “変数 j ”, “算術演算 $j+1$ ”, “算術演算子 $+$ ”, “定数 1 ” の 5 つであり、そのうち 1 つが誤りパターン挿入の対象となる。ここで算術演算 $j+1$ が選択されたとすると、適用可能な誤りパターンは “一方の項を削除”, つまり “ j ”, “ 1 ” いずれかへの置換となる。

ただし、表 4.1 に従う変換を行うと、演習に適した誤りとならない場合が一部存在する。そこで、誤り適用後の構文をチェックするための制約ルールを設け、制約ルールを満たす誤りだけを実際に適用するという方法を取る。以下、制約ルールの概要を述べる。

構文エラー

誤りパターンとして“別変数への置換”を適用すると、未定義変数の参照や互換性のないデータ型の変数への代入など、結果的に構文エラーに相当する誤りを生成する場合がある。これは 4.2.2 節に示した演習に適した誤りの定義に反するので、任意の 2 変数 i, j において、(1) i と j が同一スコープ内に存在、(2) i, j のデータ型が同一（配列変数であれば次元数と要素数も同一）、かつ (3) 代入 $i = j$ または i と j の比較演算が存在、の 3 つの条件が全て満たされる場合に限り、 $i \rightarrow j$ の変数置換を認める。これにより、構文エラーの可能性のある誤りは自動生成しないようにする。

不正又は不自然な式

誤り挿入対象の文法が“定数”の場合の誤りパターンは“定数値の変換”となるが、分母の定数の場合、 $i/2 \rightarrow i/0$ のように式の分母が 0 となる誤りを生成する場合がある。しかしこれは単なる数学上の誤りに過ぎず、アルゴリズムとの矛盾とは無関係であるため、変換候補から除外する。同様に、 $i*1$ 、 $i+0$ などの無意味な演算式はアルゴリズムに関係なく誤り箇所の発見が容易となるため、これらも変換候補から除外する。

4.4 実装

4.3 節で提案した手法を実現するシステムの試作版を、図 4.2 の構成で実装した。本試作システムは Windows XP (SP2) 上で動作する。字句・構文解析は、flex 2.5.4, bison 2.1 を用いて実装した。C コンパイラとして gcc 3.4.4, 誤り挿入対象検索及び誤りパターン適用は Scheme で実装し、処理系には SCM 5e2 を利用した。プログラムの規模は字句・構文解析部が約 1,000 行、誤り挿入対象検索・誤りパターン適用部が約 3,600 行である。

以下、実装システムの動作手順について説明する。

利用者は、1) ソースコード、2) 実装アルゴリズムのパラダイム、3) 1 つのソースコードに挿入する誤りの個数、4) 生成する演習問題の数をパラメータとしてシステムに与える。

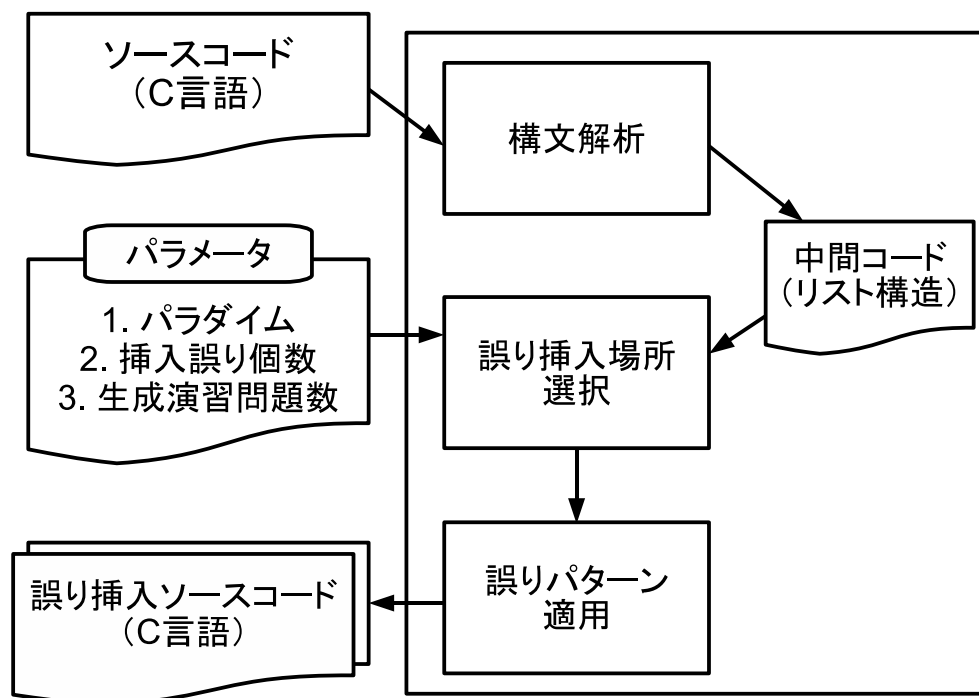


図 4.2: 試作システムの構成

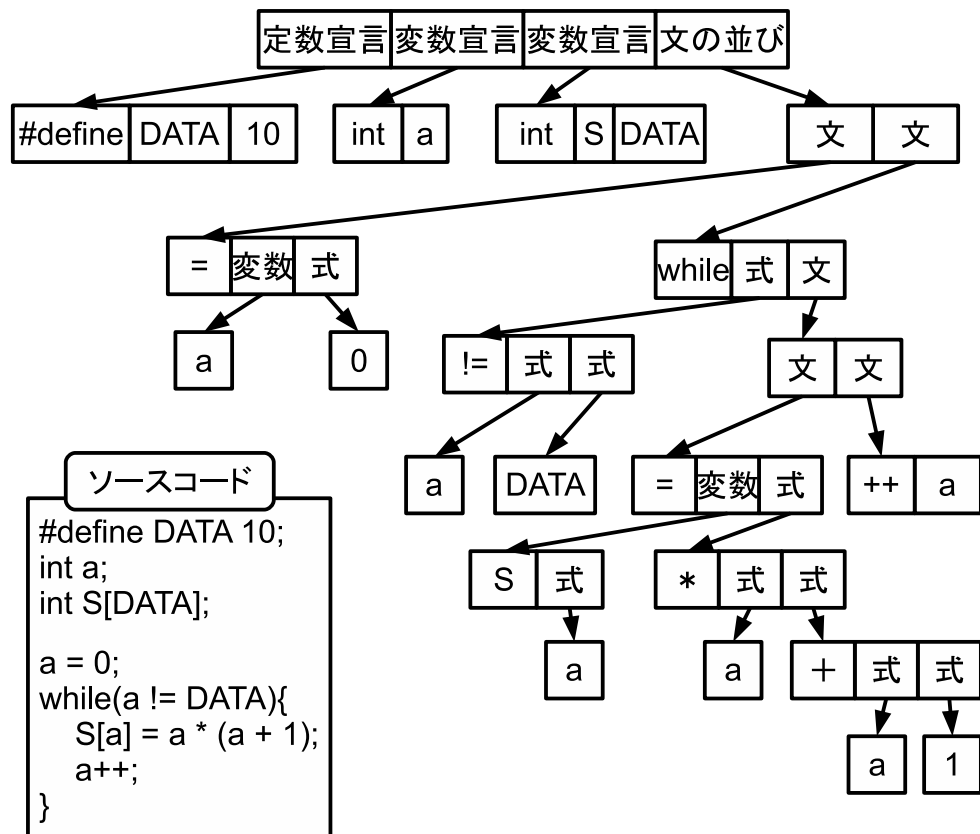


図 4.3: 中間ノード（構文木）の例

システムはまず入力されたソースコードを字句・構文解析し、出力として構文木の構造をとる中間コード（図 4.3）を生成する。実装における中間コードは、Scheme で解析可能なリスト構造（S 式）で表現する。

次に生成した中間コードを走査し、誤り挿入対象を検出する。一回目の走査で、パラダイムに対応する構文、誤り挿入対象決定に関連する変数や周辺構文を記録しておく。その後二回目の走査で、変数の依存関係や構文の位置関係をチェックし、誤り挿入可能な位置を特定する。誤り挿入可能と特定された構文には、そのノードの親ノードとして、誤り挿入対象を示すリスト（MIS リスト）を挿入する（図 4.4 右上）。

二回目の走査が完了したら、挿入した MIS リスト群から、指定された挿入誤り個数だけ MIS リストをランダムに選択する。選択された MIS リストの子孫ノードが、実際に誤りを挿入する構文となる。

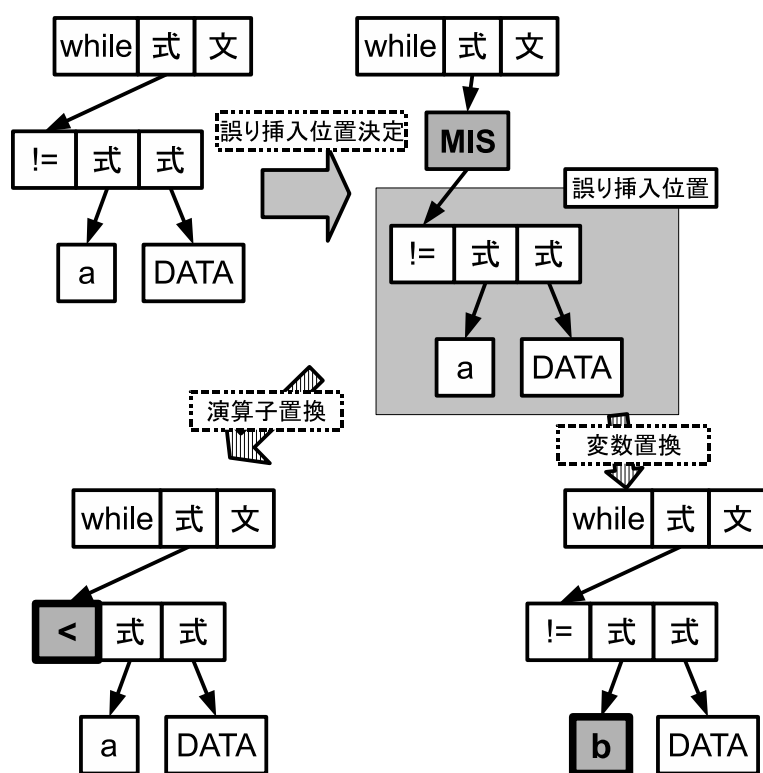


図 4.4: MIS ノード挿入と誤りパターン適用

選択した各 MIS リストの子孫ノードについてそれぞれ構文をチェックし、表 4.1 に従い対応する誤りパターンでリストを置換する (図 4.4 下). 1つの誤り挿入対象に対して複数の誤りパターンが適用可能な時は、ランダムに一つ誤りパターンを選択する. また置換後の構文については制約ルールのチェックを行い、ルールを満たさない場合は構文を置換前の状態に戻し、別の誤りパターン適用を試みる.

誤り挿入が完了したら、最後に中間ノードをソースコードに変換して出力する. 以上の誤り挿入対象決定と誤りパターン適用の処理を、生成する演習問題数分繰り返す.

4.5 間違い探し演習の実践

提案手法の評価に先立ち、間違い探し演習に対する学習者の反応を確認するため、大阪大学基礎工学部情報科学科の2年次後期必修科目「データ構造とアルゴリズム」において間違い探し演習の評価実験を実施した. 受講者は1年次及び2年次前期に Pascal 及び C 言語のプログラミングの経験がある. 演習時間は1回15分程度で、その時間中に演習問題 (Q1-Q3) 及び演習に関するアンケート (Q4,Q5) への回答を実施するという形式で計4回 (Exp.1-4) の評価実験を実施した. また、各回とも挿入誤りの異なる問題を複数用意し、各学生にはランダムに選んだ1問を与えた. 取り上げたアルゴリズムや回答者数などの各回の実施状況を表 4.2 に示す.

なお、第1,2,4回は試作システムで演習問題の生成を行ったが、第3回のみ手作業で演習問題を作成した. これは、第3回で扱ったアルゴリズムが4.3.1節で述べたどのパラダイムも利用しておらず、for文を処理の中心とするものであったため“その他”のパラダイムで適用可能な誤りパターンが存在しなかったことが原因である. このようなソースコードへの対応は今後の検討課題である.

演習では、与えられたソースコードに対して (Q1) 誤りを発見し修正せよ、(Q2) 元になっているアルゴリズムとの相違点について考察せよ、という問題を提示する. 第2回以降は、提示されたアルゴリズムの要旨を理解しているかを把握する為に (Q3) そのアルゴリズムについて簡潔に説明せよ、という問題を追加した. 図 4.5 に選択ソートを題材にした演習問題の例を示す.

またアンケートでは、演習がアルゴリズムの理解に役立ったかを問う選択肢式の質問 (Q4) と、間違い探し演習について自由記述での回答を求める質問 (Q5) を用意した. Q4 では以下に列举する選択肢を設けた. 選択肢 1,2 は演習によって理解が深まった、選択肢 3,4 は演習によって理解が深まらなかったと解釈できる.

選択肢 1: より深く理解するのに役立った

表 4.2: 講義での間違い探し演習実施状況

	Exp.1	Exp.2	Exp.3	Exp.4
	Binary	Selection	Shell	Quick
	search	sort	sort	sort
Q1,Q2,Q3 解答者数	72	73	69	67
Q4,Q5 回答者数	58	66	49	44
問題数	6	3	3	3

学籍番号	名前
次に示すソースコードは「選択ソート(昇順)」を実装したものであるが、1ヶ所だけ間違いが含まれています。	
<pre> 1 for(k = n-1; k > 0; k--){ 2 i = 0; 3 m=k; 4 while (i < k){ 5 if (data[i] < data[m]) 6 m = i; 7 i = i + 1; 8 } 9 t = data[m]; 10 data[m] = data[k]; 11 data[k] = t; 12 }</pre>	(1) 間違いを修正して下さい。 (2) 間違いが含まれていることによって、正しいアルゴリズムと、結果や動作がどのように変わってしまうかについて考察してください。 (3) 選択ソートというアルゴリズムについて簡潔に説明してください。

図 4.5: 演習問題の例

表 4.3: Q4 の回答の人数分布

	Exp.1	Exp.2	Exp.3	Exp.4
選択肢 1	45	48	30	32
選択肢 2	8	10	7	6
選択肢 3	2	1	1	2
選択肢 4	3	2	4	3
選択肢 5	-	5	7	1

選択肢 2: 演習前は誤解をしていたが解消できた

選択肢 3: 演習をしたために理解が曖昧になった

選択肢 4: 演習を通じて内容が理解できなかった

選択肢 5: その他（自由回答，第 2 回以降で追加）

Q4 の集計結果を表 4.3 に示す．各回とも 75%以上の学生が選択肢 1,2 のいずれかを回答しており，大半の学習者にとって間違い探し演習はアルゴリズムへの理解に効果があると考えていることがわかった．

また，理解が深まったと考えた（選択肢 1,2 の回答を行った）学習者と理解が深まったとは考えなかった（選択肢 3,4 の回答を行った）学習者の 2 つのグループに分けて，Q1-Q3 の正答率を集計したものを表 4.4 に示す．選択肢 1,2 を選んだ学習者に対しては，第 1-4 回各々の場合について示しているが，選択肢 3,4 を選んだ学習者は数が少ないため，第 1-4 回の合計のみ記載している．Q2,Q3 では，完全に誤答ではないが説明不足である解答を‘部分正答’として集計した．

表 4.4 より，Q4 で選択肢 1,2 を回答した学習者について，Q2 の正答率が第 4 回を除いて 7 割を下回っており，逆に選択肢 3,4 の回答を行った学習者について，Q2 の正答率が 2 割以上という結果が得られた．つまり一部の学習者は，理解できた/できないという主観的な感覚と，自身の実際の理解度が合致していないことを意味している．本演習を実施する場合は，採点結果のフィードバックを行うことで，学習者の理解度に対する認識を確実にするプロセスが重要だと言える．

Q5 については，4 回の実施で計 67 件の意見が得られた．分類すると次のようになった．

1. アルゴリズムを理解しないと解けない（14 件）
2. ソースコードを精読する契機となる（9 件）

表 4.4: Q4 の回答結果ごとの正答率

	Q4 回答				
	1 or 2				3 or 4
	Exp.1	Exp.2	Exp.3	Exp.4	Exp.1-4
回答者数	53	58	37	38	18
Q1	69.8%	93.1%	75.7%	80.0%	55.6%
Q2	67.9%	69.0%	54.1%	84.2%	27.8%
(部分正答含む)	84.9%	79.3%	64.9%	73.7%	44.4%
Q3	-	75.9%	67.6%	68.4%	53.8%
(部分正答含む)	-	96.6%	97.3%	100%	100%

3. ゲーム性が高く面白い (10 件)
4. 「良いと思う」などの肯定的意見 (8 件)
5. 「よくわからない」などの否定的意見 (1 件)
6. 演習に対する学習者側からの助言 (17 件)
7. その他 (8 件)

1. 及び 2. の意見が多いことから、アルゴリズムに対する理解を深めるという本演習の趣旨が学習者にも意識されていたことがわかる。また、3. の意見は、間違い探しという演習形式が学習者の興味を引きつけたことを示していると言える。

以上の結果から、間違い探し演習は学習者には概ね好評であり、かつ演習の目的を十分に満たしていることが確認できた。

4.6 提案手法の評価

本節では提案手法の評価方法とその結果を述べる。

評価基準として、手作業で挿入する誤り（以下**手作業誤り**）と、提案手法で自動生成可能な誤り（以下**自動誤り**）を比較し、自動誤りに対する評価が手作業誤りに対する評価と同等以上であれば、提案手法は有効であると判断する。

4.6.1 評価方法

評価に用いるソースコードは、文献 [44] を参考にアルゴリズムの学習で扱う典型的なアルゴリズムを選択した。それぞれ、クイックソート（分割統治法）、二分探索（再帰）、ナップザック問題（動的計画法）、部分和问题（動的計画法）、ダイクストラの最短経路問題（貪欲法）、再帰を使わない二分探索（その他）、選択ソート（その他）の 7 種類のソースコードを用意した。

用意した各ソースコードに対し、まず試作システムを利用して誤りを一つ含む演習問題を生成した。提案手法で生成可能な全パターンの誤りを適用し、7 種類のソースコードから合計 280 個の自動誤りのサンプルを得た。次に同じソースコードに対して、誤りを 1 つ挿入した演習問題を 3 つずつ手作業で作成した。本評価では、計算機科学を専門とする教員 3 名と博士後期課程の学生 3 名（以下**問題作成者**）が手作業誤りの作成を行い、126 個、複数人で重複したサンプルを除くと 95 個の手作業誤りのサンプルを得た。結果、自動誤りと手作業誤りを合わせて、重複するサンプルを除く合計 328 個の誤りサンプルが得られた。

その上で、328 個の全誤りサンプルについて、間違い探し演習の問題として有用か否かの評価を行った。評価には、著者らの所属研究室の学生 12 名（情報科学研究科博士前期課程 2 年 4 名、1 年 4 名、情報科学科 4 年 4 名。以下**評価者**）の協力を得た。評価の際、各サンプルは評価者ごとにランダムな順番で提示し、また各サンプルが提案手法と手作業のいずれで作成したものであるかは評価者には知らせなかった。

評価者による有用性判定の完了後、各サンプルの演習問題としての有用性を、有用と判断した人数の全 12 人に対する割合（以下**支持率**）として表した。なお、演習問題としての有用性は 4.2.2 節で定義した基準に拠ることとしたが、実際には‘有用/有用でない’の境界の基準は評価者ごとの個人差が大きい。そのため本評価では、過半数の評価者から支持される誤りサンプルを、演習問題として有用であるとする。

4.6.2 評価結果

以下、提案手法が演習問題として有用な誤りをどの程度生成できているかについて評価する。表 4.5 は自動誤りの各サンプルについて、有用性の支持率別に 4 グループに分けてカウントした結果である。全自動誤りサンプルの 73.2%（ $=35.7\% + 37.5\%$ ）が 50%を超える評価者から有用性を評価された。

一方、手作業誤りにおける同様の支持率分布では、サンプルの 82.1%（ $=57.9\% + 24.2\%$ ）が 50%を超える評価者から有用性を評価されており、自動誤りは手作業誤りに比べると若干支持率

表 4.5: 自動誤りサンプルの有用性支持率

	有用性の支持率 (%)			
	75% 超過	75%以下 50%超過	50%以下 25%超過	25% 以下
クイックソート	20	21	10	4
二分探索 (再帰)	15	23	15	8
二分探索 (反復)	13	26	13	3
ナップザック問題	14	10	1	0
部分和問題	10	3	3	0
最短経路問題	20	12	10	0
選択ソート	8	10	7	1
合計	100 (35.7%)	105 (37.5%)	59 (21.1%)	16 (5.7%)
手作業誤り	55 (57.9%)	23 (24.2%)	15 (15.8%)	2 (2.1%)

が低い結果となっている。有用性の低い誤りの自動生成を抑えるために、提案手法を改良する必要がある、次節で議論する。

4.6.3 提案手法の改善

4.6.2 節の結果を踏まえ、提案手法の改善方法について検討する。

支持率が50%以下の自動誤りサンプル75個について、適用した誤りパターンの種類別にサンプル個数を集計した結果を表4.6に示す。総数が2個以下の誤りパターンは「その他」でまとめている。表4.6から、特定の誤りパターンに支持率の低い誤りが固まっていることが確認できる。

また各サンプルを検証した結果、生成を回避できる誤りパターンが31個存在することがわかった。以下、対応可能な誤りパターンについて説明する。

“変数 → 変数+1”，“変数 → 変数-1”の誤りパターンのうち、二分探索において、検索値を格納する変数に対して加算，減算するパターンが合計12個存在した。しかし検索値はソース全体を通して不変であることから、不変の値を持つ変数の増減は“不自然な式”と判断された結果となった。そこで、ソースコード中に表れる変数のうち、(1) その変数に値を代入する代入文が1つだけ存在、

表 4.6: 有用評価が 50%以下の誤りパターン

誤りパターン	総数
変数 → 変数+1	17
変数 → 別の変数	12
変数 → 変数-1	12
定数 1→2	9
論理演算子置換	6
配列添字の変数置換	4
算術演算子置換	3
その他	12

(2) その代入文は、一度しか実行されないことが保証されている、つまり変数が宣言されたスコープ内（グローバル変数の場合はソース全体）の制御構文（if,for,while）外で現れる、の2つを満たす変数は“格納値が不変の変数”と判断し、変数+1, 変数-1 の誤りパターンを適用しないという制約ルールを追加することとした。これにより二分探索に限らず様々なアルゴリズムで、不変値を格納する変数を増減する誤りを生成しない対応が可能となる。

また変数置換を適用した誤りサンプルでは、連続する2つの代入文の左辺が同一になるサンプルが含まれていた（8個）。結果、前の代入文が無意味な実行文となることから、有用性評価が総じて低い結果となった。この誤りパターンに対しては、「代入文の左辺の変数置換は、直前あるいは直後の代入文の左辺の構文と異なる形にする」の制約ルールを追加することで、誤りパターンの適用を避けることとした。

定数値に適用する誤りパターンのうち、1→2, 2→3 の置換は、自動誤り 11 個と手作業誤り 1 個全てのサンプルで、過半数の評価者の支持を得られなかった。評価者のコメントから、定数増減の誤りパターンはソースコード作成者の単純な入力間違いを想定させ、アルゴリズムを意識せずに誤りを修正できてしまう可能性が指摘された。よって、1→2, 2→3 の誤りパターンは対応表から削除するのが適当であると判断した。

以上のアプローチを提案手法に適用した結果、支持率 50%超のサンプル数には変化がなかったが、支持率 50%以下のサンプル数が 31 個減少し、支持率分布は表 4.7 に示す形に変化した。50%を超える評価者から有用性を評価された誤りが 82.3%(=40.16%+42.17%) となり、手作業誤りの支持率とほぼ同等となった。よって、本提案手法は手作業で演習問題を作成する場合と同程度に有

表 4.7: 自動誤りサンプルの有用性支持率（改善版）

	有用性の支持率 (%)			
	75% 超過	75%以下 50%超過	50%以下 25%超過	25% 以下
クイックソート	20	21	7	0
二分探索（再帰）	15	23	9	3
二分探索（反復）	13	26	5	1
ナップザック問題	14	10	1	0
部分和問題	10	3	3	0
最短経路問題	20	12	10	0
選択ソート	8	10	5	0
合計	100 (40.16%)	105 (42.17%)	40 (16.06%)	4 (1.61%)
手作業誤り	55 (57.9%)	23 (24.2%)	15 (15.8%)	2 (2.1%)

用となったと考えられる。

4.6.4 提案手法の再現率評価

表 4.7 では、支持率が 75% を超えるサンプルの割合には、自動誤りと手作業誤りで若干の開きがある。そこで、現在の提案手法が、問題作成者が作成した手作業誤りをどの程度再現可能かを確認し、提案手法が現在カバーしきれていない誤り挿入手法を検討する。

まず手作業誤りについて、提案手法が同一の誤り挿入対象を決定できた割合（誤り対象再現率）と、挿入対象が同一でかつ挿入する誤りパターンも同一であった割合（挿入誤り再現率）を求めた（表 4.8）。その結果、誤り対象再現率がクイックソートを除いていずれも 70% 以上であるのに対し、適用する誤りパターンも含めた挿入誤り再現率は、二分探索（反復法）の 71.4% を除いて 36.4–58.3% の範囲であった。

次に手作業誤りのサンプルを支持率別にグループ分けし、再現率の平均を求めた結果を表 4.9 に示す。表 4.9 では支持率の高いサンプルほど再現率が高くなる傾向が確認できる。つまり提案手法は、有用性が低い手作業誤りを回避できていると考えられる。ただし、誤り対象は特定できなが

表 4.8: 提案手法の再現率（アルゴリズム別）

	誤り対象 再現率	挿入誤り 再現率	総数
クイックソート	8(50.0%)	6(37.5%)	16
二分探索（再帰）	9(75.0%)	7(58.3%)	12
二分探索（反復）	13(92.9%)	10(71.4%)	14
ナップサック問題	11(78.6%)	6(42.9%)	14
部分和問題	8(72.7%)	4(36.4%)	11
最短経路問題	9(75.0%)	7(58.3%)	12
選択ソート	13(81.3%)	7(43.8%)	16
全体	71(74.7%)	47(49.5%)	95

表 4.9: 支持率別の再現率

	誤り対象再現率	挿入誤り再現率	総数
$0 < x \leq 25\%$	0(0%)	0(0%)	2
$25 < x \leq 50\%$	5(33.3%)	3(20.0%)	15
$50 < x \leq 75\%$	15(65.2%)	13(56.5%)	23
$75 < x \leq 100\%$	51(92.7%)	31(56.4%)	55

ら挿入誤りを再現できなかった手作業誤りのサンプルは合計で 24 個にのぼっていることから、有用な誤りパターンの一部を提案手法は十分に網羅しきれていないと考えられる。

4.6.5 再現率向上への検討

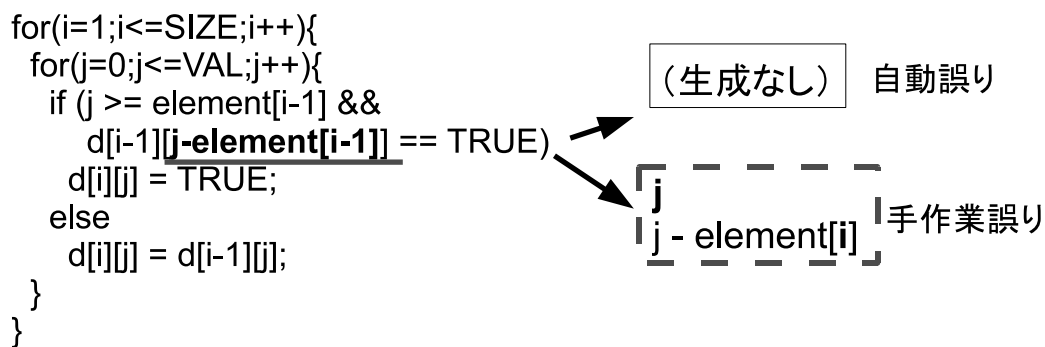
4.6.4 節の結果を踏まえ、提案手法で再現できなかった支持率 50%超過の手作業誤り 34 個について、誤りパターンの傾向を調べた。該当サンプルが 2 個以上存在した誤りパターンについて、その個数を表 4.10 に示す。

表 4.10 に列挙した誤りパターンについてサンプルを 1 つずつ検証した結果、構文ベースの誤り挿入が適用できるパターンが一部存在することがわかった。

例えば配列の添字への誤り挿入は、配列の要素数の範囲外へアクセス可能な誤りの生成を避け

表 4.10: 支持率 50%以上の再現不可手作業誤りの誤りパターン

誤りパターン	個数
配列の添字式 $[A-B] \rightarrow [A]$	5
変数置換 (4.3.2 節定義外)	4
二項演算 $\rightarrow$ 変数置換	3
二項演算子 $< \rightarrow \leq$	2
配列 $\rightarrow$ 定数	2

図 4.6: 手作業誤りの例：配列引数 $[A-B] \rightarrow [A]$

るため、提案手法では行わないこととしていた。しかし配列の添字が二項演算式 ' $A-B$ ' の場合、片方の項を削除する誤り挿入 ($[A-B] \rightarrow [A]$) を適用した手作業誤り 5 個が高い支持率を得ていた。さらにそのうち 4 個は、置換後の添字表現 A がソース内の別の文で配列添字に使用されていることがわかった (図 4.6)。よって、配列の添字への誤り挿入は「ソース中に現れる同じ配列の添字式表現への置換」に限り可能とすることで、要素数の範囲外へのアクセスを起こさない構文ベースの置換が可能となる。ただしこの誤りパターンを適用すると、手作業誤りの再現とは別の新しい自動誤りサンプルも生成されうるため、追加で評価実験を行い、安定して有用な誤りを生成できるか確認する必要がある。

また、その他の誤りパターンについては、ソースコード中の変数がアルゴリズムにおいて果たす意味的な役割 (例: 変数 mid は中間点という意味を持つ、など) を理解しないと挿入できない誤りが多数含まれていた (図 4.7)。提案手法はパラダイムを基に構文ベースで誤りパターンを決定する手法のため、変数の意味理解も含めた誤り挿入への対応は別のアプローチが必要となると考

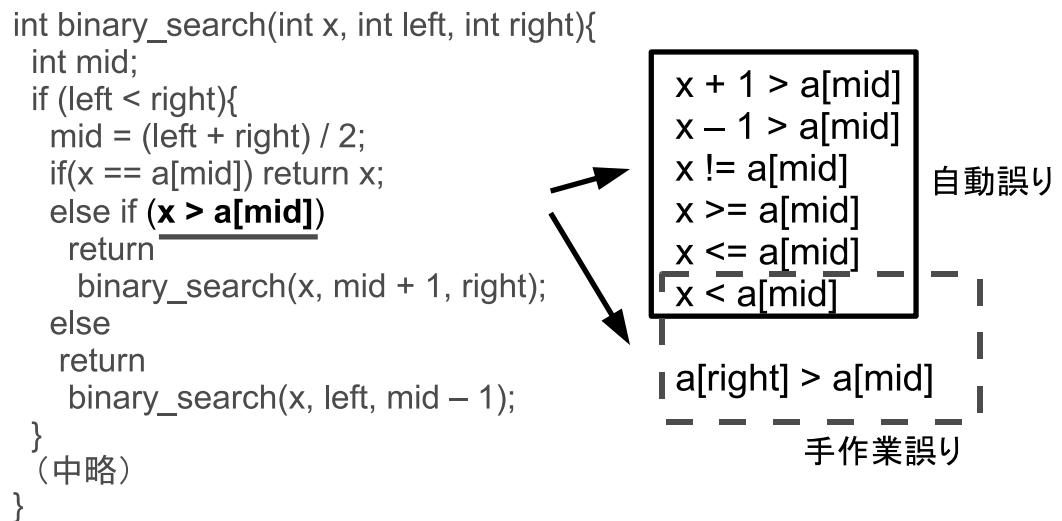


図 4.7: 手作業誤りの例：純変数 → 配列変数

えられる。これについては今後の課題である。

4.7 おわりに

本研究ではアルゴリズム学習を対象とし、ソースコードの誤り修正を行う演習を想定した演習課題の自動生成法を提案した。具体的には、アルゴリズム設計パラダイムを利用した誤り挿入対象の抽出手法、構文主導型のソースコード置換ルールによる誤り挿入方法を示した。これにより、入力されたソースコードに対して演習に適した誤りの自動挿入を実現した。

また提案手法について評価を行い、その結果を踏まえ誤り挿入手法を改善することで、提案手法による演習問題作成が手作業で生成する場合と同程度に有用であることを確認した。

今後の課題としては、ソースコードの意味解析を含めた誤り挿入手法の改良が挙げられる。さらに発展的な目標として、自動生成した演習課題の難易度付けや解答の判定処理も自動化した、アルゴリズムの自主学習環境の構築を目指している。

第5章 おわりに

本論文では、教育機関における授業を対象とした学習支援環境の構築を主軸とし、授業外学習、授業内学習、授業内演習それぞれにおいて、実用的なコストで学習効果の高い環境を実現するための研究について紹介した。

2章では、学科内の全科目、全講義の撮影と学内配信の取り組みについて、安価で継続運用可能な運用体制とするためのシステム構築という視点から紹介した。学生がスタッフとして参加する運用体制とし、同時に計算機による編集作業の支援システムの構築によって、人的作業に起因するミス発生を軽減することにより、年間100万円程度の運用コストで、2年間で1000を超える講義映像を蓄積、学内配信することができた。また実践を通して、講義映像を復習用教材や教員の自己評価用素材として利用する学習環境が定着し、学生スタッフ自身の学習効果も期待されることも確認した。

3章では、計算機による授業内支援環境として、学習者と教師のやりとりを支援するシステム inLec について紹介した。計算機を活用して学習者からのレスポンスを促すため、授業中の使用に耐えうる簡易なレスポンス用インタフェースを提供する Web アプリケーションを作成した。実際の授業で複数回運用した結果、inLec を用いることで学習者からの積極的なレスポンスが得られること、また inLec 操作履歴から授業中の活動を分析できる可能性を確認した。

4章では、アルゴリズム学習を対象とした、間違い探し演習問題の自動作成手法について紹介した。演習問題の自動作成手法として、アルゴリズム設計パラダイムを利用したソースコードへの自動的な誤り挿入手法を構築した。また本手法で生成可能な誤りパターンを評価した結果、手作業で作成するものとはほぼ同等の品質の間違い探し演習問題を、本手法で生成できることを確認した。

今後、科目非依存の課題へのアプローチについては、上記の研究成果をさらに発展させ、大量に蓄積された授業映像コンテンツの再利用性を高める研究、特に授業内活動の履歴を利用して、映像教材として利用する際の効率を高める手法、さらに詳細な授業評価に応用する手法について検討を行う予定である。

また科目特有の課題へのアプローチについては、今後もアルゴリズム学習に焦点を当て、学習者や教師の負担を増加させずに効果的な学習環境を実現するためのシステム構築について研究を

進めていく予定である。本論文で紹介した間違い探し演習が対象とする基本的なアルゴリズムだけでなく、並列アルゴリズムや分散アルゴリズムなどより多様なアルゴリズムの学習についても支援手法を検討していく。

謝辞

本研究の全過程にわたって懇切なるご指導とご支援を賜りました，大阪大学大学院情報科学研究科 増澤利光教授に謹んで深謝の意を表します。また，本論文作成にあたり有益なご教示ご指導を賜りました，大阪大学大学院情報科学研究科 井上克郎教授，大阪大学産業科学研究所 八木康史教授に深謝いたします。

本論文作成におけるご指導のみならず，著者の研究活動において多大なご指導とご助言をいただきました，大阪大学大学院情報科学研究科 角川裕次准教授に深謝いたします。

これまでの著者の研究活動全般にわたり懇切なるご指導とご支援を賜りました，大阪電気通信大学 都倉信樹学長に謹んで深謝の意を表します。

また，鳥取環境大学環境情報学部情報システム学科 齊藤明紀教授，熊本大学総合情報基盤センター 永井孝幸准教授には，特に著者の鳥取環境大学における研究活動において，多岐にわたる有益なご指導ご助言をいただきました。ここに深謝いたします。

本研究における有益なご助言やご協力をいただいた，大阪大学大学院情報科学研究科 大下福仁助教に深く感謝いたします。

アルゴリズム学習における間違い探し演習問題自動生成の研究において，多大なるご支援ご協力を頂いた，大阪大学大学院情報科学研究科修士 伊藤亮太氏に深く感謝いたします。

本研究における有益な議論，また評価活動へのご協力をいただきました，鳥取環境大学環境情報学部情報システム学科の皆様と，大阪大学大学院情報科学研究科増澤研究室の皆様に感謝いたします。

最後に，本論文作成に至るまで著者の研究活動を後押ししてくれた，妻 由里子と両親に，心から感謝します。

参考文献

- [1] メディア教育開発センター. e ラーニング等の ICT を活用した教育に関する調査報告書 (2007 年版). <http://www.nime.ac.jp/reports/001/main/eLearning07-jp.pdf>, 2007.
- [2] Danish Technological Institute. Study of the e-learning suppliers ' "market" in Europe. http://ec.europa.eu/education/programmes/elearning/doc/studies/market_study_en.pdf, Jan 2005.
- [3] 青木恭介. 複線型評価制度と高等教育機関の負担. 工学教育, Vol. 53, No. 3, pp. 14–19, May 2005.
- [4] 国宗永佳, 新村正明, 和崎克己, 不破泰, 師玉康成, 中村八束. 信州大学インターネット大学院の現状と評価. 教育システム情報学会誌, Vol. 22, No. 4, pp. 264–271, Oct 2005.
- [5] 井上博樹, 奥村晴彦, 中田平. Moodle 入門 - オープンソースで構築する e ラーニングシステム. 海文堂, Sep 2006.
- [6] Advanced Distributed Learning. SCORM(R). <http://www.adlnet.gov/scorm/>, (accessed: Nov 24, 2008).
- [7] 鈴木克明. e-Learning 実践のためのインストラクショナル・デザイン. 日本教育工学会論文誌, Vol. 29, No. 3, pp. 197–205, Feb 2006.
- [8] 宮川繁. 1. オープン・コース・ウェアの現状と展望 (<特集> e ラーニングの広がりと連携). 情報処理, Vol. 49, No. 9, pp. 1029–1038, Sep 2008.
- [9] Robert M. Gagné, Walter W. Wager, Katharine C. Golas, and John M. Keller. *Principles of Instructional Design (fifth edition)*. Wadsworth Publishing, Jun 2004.
- [10] 長瀧寛之, 都倉信樹. 履歴を利用した問題選択を行う演習・試験システムの実現. 情報処理学会第 61 回全国大会 講演論文集, pp. 4–367–368. 情報処理学会, Oct 2000.

- [11] 長瀧寛之, 永井孝幸, 都倉信樹. 大学内でのアンケートシステムの運用事例と考察. 平成 16 年度工学・工業教育研究講演会講演論文集, pp. 191–192. 日本工学教育協会, Jul 2004.
- [12] 長瀧寛之, 永井孝幸, 都倉信樹. 学生が作業スタッフとして参加する学科内全科目の講義撮影・配信への取り組み. 工学教育, Vol. 55, No. 5, pp. 8–14, Sep 2007.
- [13] 長瀧寛之, 永井孝幸, 都倉信樹. 講義中の学生と教師のやりとりを支援するシステムの構築と運用. 情報技術レターズ (情報科学技術フォーラム (FIT 2003)), pp. 387–388, Sep 2003.
- [14] 長瀧寛之, 伊藤亮太, 大下福仁, 角川裕次, 増澤利光. アルゴリズム学習における間違い探し形式の演習課題を自動生成する手法の提案と評価. 情報処理学会論文誌, Vol. 49, No. 10, pp. 3366–3376, Oct 2008.
- [15] 濱田正人, 長瀧寛之, 都倉信樹. 演習内に複数の講義室で情報を共有し蓄積できる教員用システムの開発と運用. 電子情報通信学会技術研究報告, Vol. 104, No. 643, pp. 13–18, Jan 2005.
- [16] 長瀧寛之, 北村英純, 永井孝幸, 都倉信樹. 遠隔ペアプログラミング支援システム SATORI の開発とプログラミング教育への適用. 平成 16 年度工学・工業教育研究講演会講演論文集, pp. 445–446. 日本工学教育協会, Jul 2004.
- [17] 林昌弘, 長瀧寛之, 大下福仁, 角川裕次, 増澤利光. 議論活動における調査資料の活用を支援するシステム HAKASE の構築. 情報処理学会研究報告, Vol. 2008, No. 13, pp. 119–126, Feb 2008.
- [18] 村上正行, 西口敏司, 亀田能成, 角所考, 美濃導彦. 京都大学での実践に基づく講義アーカイブの調査分析. 日本教育工学会論文誌, Vol. 28, No. 3, pp. 253–262, 12 2004.
- [19] 坂元昂, 中原淳, 西森年寿. e ラーニング・マネジメント 大学の挑戦. オーム社, Jul 2003.
- [20] 特定非営利活動法人日本イーラーニングコンソーシアム (編). e ラーニング白書 2008/2009 年版: 2.2.3 インターネット等を用いた遠隔教育, pp. 71–75. 東京電機大学出版局, Aug 2008.
- [21] on Campus: 奈良先端科学技術大学院大学電子図書館, UNIX magazine 2006 Autumn, pp. 78–85. 株式会社アスキー, Oct 2006.
- [22] 牛島和夫. 講義記録システムを含む教育環境の構築と教育実践. 情報処理学会研究報告, Vol. 2004, No. 11, pp. 23–30, Jan 2004.

- [23] 光原弘幸, 能瀬高明, 三好康夫, 緒方広明, 矢野米雄, 松浦健二, 金西計英, 森川富明. 徳島大学における e-Learning のシステム開発・運用・実践. 日本教育工学会論文誌, Vol. 29, No. 3, pp. 425–434, Feb 2006.
- [24] Paul E. Dickson, W. Richards Adrion, and Allen R. Hanson. Automatic creation of indexed presentations from classroom lectures. In *ITiCSE '08: Proceedings of the 13th annual conference on Innovation and technology in computer science education*, pp. 12–16, Jun 2008.
- [25] Fleming Lampi, Stephan Kopf, Manuel Benz, and Wolfgang Effelsberg. An automatic cameraman in a lecture recording system. In *Emme '07: Proceedings of the international workshop on Educational multimedia and multimedia education*, pp. 11–18, Sep 2007.
- [26] 日本視聴覚教育協会. 社会教育を活性化する e ラーニングシステムの構築に関する調査. 視聴覚教育, No. 682, pp. 20–30, Aug 2004.
- [27] Surendar Chandra. Lecture video capture for the masses. In *ITiCSE '07: Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*, pp. 276–280. ACM, Jun 2007.
- [28] 永井孝幸, 長瀧寛之. 大規模分散ネットワーク環境における教育用計算機システム 2. 教育用計算機環境の事例 6. 必携パソコン化編. 情報処理, Vol. 45, No. 3, pp. 255–258, Mar 2004.
- [29] 齊藤明紀, 長瀧寛之, 永井孝幸. 個人所有ノート PC を用いた計算機環境. 情報処理学会研究報告, Vol. 2005, No. 83, pp. 29–34, Aug 2005.
- [30] 長瀧寛之, 永井孝幸, 都倉信樹. 学科内全講義のビデオ撮影・蓄積・配信への取り組み. 平成 17 年度工学・工業教育研究講演会講演論文集, pp. 104–105, Sep 2005.
- [31] 長瀧寛之, 永井孝幸, 都倉信樹. 学生による講義ビデオのしおり付け実験の報告. 情報処理学会研究報告, Vol. 2003, No. 103, pp. 27–34, Oct 2003.
- [32] Takayuki Nagai. Laborsaving Video Archive System for Daily Manual Recording. In *8th International Conference on Information Technology Based Higher Education and Training (ITHET 2007)*, pp. 216–221, Jul 2007.

- [33] Juan Manuel Dodero, Camino Fernández, and Daniel Sanz. An experience on students' participation in blended vs. online styles of learning. *SIGCSE Bull.*, Vol. 35, No. 4, pp. 39–42, Dec 2003.
- [34] Quintin I. Cutts and Gregor E. Kennedy. Connecting learning environments using electronic voting systems. In *ACE '05: Proceedings of the 7th Australasian conference on Computing education*, pp. 181–186. Australian Computer Society, Inc., 2005.
- [35] 辻義人, 田島貴裕, 西岡将晴, 奥田和重. 異なる背景を持つ受講者の遠隔教育に対する評価観点の検討—遠隔サイエンス・コミュニケーションの実現に向けて—. *コンピュータ&エデュケーション*, Vol. 25, pp. 82–87, Dec 2008.
- [36] The Joint Task Force for Computing Curricula. Computing curricula 2005 - The Overview Report. ACM (online), available from http://www.acm.org/education/curric_vols/CC2005-March06Final.pdf. (accessed: Nov 16, 2007).
- [37] 斐品正照, 徳岡健一, 河村一樹. 構造化チャートを用いたアルゴリズム学習支援システム. *情報処理学会論文誌*, Vol. 45, No. 10, pp. 2454–2467, Oct 2004.
- [38] 新村晃示, 鈴木浩之, 稲垣文雄, 伊藤大輔, 小西達裕, 伊東幸宏. プログラミング言語学習を意識させないアルゴリズム作成・テスト環境の構築. *教育システム情報学会研究報告*, Vol. 19, No. 5, pp. 63–70, Jan 2005.
- [39] M.C. Carlisle, T.A. Wilson, J.W. Humphries, and S.M.Hadfield. RAPTOR: a visual programming environment for teaching algorithmic problem solving. In *Proc. the 36th SIGCSE technical symposium on Computer science education*, pp. 176–180, Feb 2005.
- [40] T. Watts. The SFC editor a graphical tool for algorithm development. *Journal of Computing Sciences in Colleges*, Vol. 20, No. 2, pp. 73–85, Dec 2004.
- [41] 柏原昭博, 久米井邦貴, 梅野浩司, 豊田順一. プログラム空欄補充問題の作成とその評価. *人工知能学会論文誌*, Vol. 16, No. 4, pp. 384–391, 2001.
- [42] 布広永示, 松下孝太郎, Kenneth J. Mackin, 大城正典, 山崎和子. 遺伝的アルゴリズムを用いたパズル型プログラミング学習支援システムの設計と開発. *教育システム情報学会誌*, Vol. 25, No. 2, pp. 204–213, Sep 2008.

- [43] 菅沼明, 峯恒憲, 正代隆義. 学生の理解度と問題の難易度を動的に評価する練習問題自動生成システム. 情報処理学会論文誌, Vol. 46, No. 7, pp. 1810–1818, Jul 2005.
- [44] 浅野哲夫, 和田幸一, 増澤利光. アルゴリズム論. オーム社, Jul 2003.