

研究室規模の蔵書管理と文献共有による学習支援 —コードレビューを取り入れた共時学習としての輪講の支援—

川鯉光起[†] 中矢誠[†] 富永浩之[†]

書籍関連サイトと連携した研究室規模の蔵書管理システムを構築し、その上に文献共有による学習支援を提案する。蔵書管理の対象は情報系の専門書であり、目的は知識や技能の向上のための精読である。そこで、貸出履歴や既読者のコメントを活用し、書籍利用を活性化させるため、記録部と分析部を追加した。また、文献共有による精読の形態を、個読・引継・輪講の3通りに分類し、支援に必要な機能を検討する。本論では、特に、プログラミングの解説書を題材とする演習付の輪講に着目し、コードレビューを取り入れた支援システムを検討する。レビュー者が書いているコードの編集状況をリアルタイムに中継し、レビュー者がその場でコメントを付与していく。その過程を、スナップショットとして保存し、後から再生して追体験することができる。試作版として実装した機能を議論し、ユーザによるシステムの試用結果について報告する。

Learning Support using Common Use Books Based on Library Management in a Laboratory Scale - Lectures and Exercises in Turn with Code Review -

KOKI KAWAGOI[†] MAKOTO NAKAYA[†] HIROYUKI TOMINAGA[†]

We have proposed a library management system in a laboratory scale for common use books about technology and science. The system cooperates with several book search and seller sites in order to acquire book information. We also consider learning support functions using common use books, especially about information engineering. We added record and analysis sub-modules into the system. We regard careful reading as three types; self-pace, inheriting and synchronic. We focus lectures and exercises in turn for a handbook of programming as the third aspect. We designed a support tool with code review functions for peer review. It offers streaming view of a coding program by a reviewee and real-time chat for comments of reviewers, which are recorded as snapshot series. We report results of experiments of a prototype tool.

1. はじめに

本研究室では、大学の図書館とは別に、教員の蔵書および情報系サークルで購入した図書を共同で管理している。学生が卒業時に教科書などを寄贈することもある。現在、蔵書数は3000冊程度である。分野は、情報系や数理系を中心として、就職関連や娯楽的な本も含まれる。プログラミングやLinuxの解説書などは、セミナーや演習などで同時に活用するため、同じ書籍を複数所有している。ゼミ生やサークル生だけではなく、教員の授業を受講する学生も図書を借りに来ることがある。このような研究室規模の蔵書管理をオンラインで行うには、個人向けのツールでは、不十分である。また、大量の蔵書の登録作業がネックになる。専門の司書がいるわけではなく、利用者が分担して蔵書の管理も担う必要がある。

一方、書籍関連サイトの中には、サイト内での検索サービスを利用できるだけでなく、他のサイトからも使えるAPIを公開しているものもある。対応したプログラム言語で、簡単なスクリプトを作成すれば、提供元の書籍情報にアクセスできるWebページが構築できる。これらを利用すれば、蔵書の書籍情報の取得や管理が容易になる。本研究では、まず、これらの書籍関連サイトと連携した研究室規

模の蔵書管理システムの開発を行ってきた[1][2]。

本システムが主な対象とするのは、学生が学習や研究に用いる書籍であり、何らかの知識や技能を習得することを目的とする。そのため、全体または必要な個所を選んで精読することになる。その際、関連する書籍も調べたり、他人の助言を必要とすることが多い。そこで、蔵書の貸出状況や既読者のコメントを活用し、書籍利用を活性化することが求められる。また、書籍の内容を理解するため、興味や目的が近い利用者が、相互に協力し合う機会を提供することも必要である。本研究では、蔵書管理をベースに、文献共有による学習支援の機能について検討する。本論では、特に、プログラミングの解説書に着目し、演習を伴う輪講を支援する機能について議論する。

2. 研究室規模の蔵書管理システム

2.1. システムの概要

本研究が提案している蔵書管理システムのモジュール構成と主な機能は、図1の通りである[3]。表紙部は、ユーザ登録とユーザ認証を行う。登録部と検索部は、一般的な書籍情報および研究室の蔵書情報を扱う。外部の検索サイトや販売サイトと連携する。司書部は、ユーザ区分に応じた貸出と返却を行い、利用履歴を保存する。現在、主要なDBと基本機能を実装中である。総称的な著作物としての書籍と、個々の具体物としての蔵書の情報を整理し、検索/

[†]1 香川大学
Kagawa University

登録/貸出/返却/購入の作業を効率化する。実際の本棚の収納位置に合わせ、表紙や背表紙の画像を並べ、書籍名が似た蔵書や関連する蔵書を探しやすくする。蔵書の有無を忘れて重複購入してしまうことを防ぐため、書店からのモバイル端末での利用にも対応する。電子書籍も扱えるようにし、輪講の補助を行う。ユーザの区分を設け、グループ毎に権限を設定して、複雑な貸出のルールに対応する。既存のシステムとの差異として、既読者から未読者への情報伝達に基づく学習支援も提供する。既存のサイトを参考にしながら、ユーザ評価を行って、使いやすいインタフェースを提供する。

2.2. 書籍サイトとの連携

本システムは、外部の書籍関連サイトと連携し、登録と検索の作業を効率的に実現する。書籍情報を検索する API を公開している書籍販売サイトには、Amazon や楽天 Books などがある[4][5]。これらのサイトでは、表紙の画像や目次の一覧、本文の一部が閲覧できたりする。また、売上ランキングや関連本の紹介、既読者のレビューや評価などを知ることができる。さらに、廃刊本の中古情報や、近刊本の最新情報なども取得することができる。しかし、購入の促進を前提としており、他のサイトとの比較の制限など、利用目的や頻度の点で制約がある。

一方、公的機関の書籍検索サイトとして、2010 年より国立国会図書館(NDL)がオンラインサーチ機能を提供している[6]。非営利であれば、クレジットの明記のみで利用できる。ただし、特定のサーバから過剰なアクセスがあると、制限されることがある。NDL が提供する API には、複数の種類がある。基本的な利用方法としては、検索キーワードを UTF-8 エンコードし、リクエスト URL に付加してリクエストする。検索結果は、XML や HTML で返却される。本システムでは、書籍情報の登録や検索には、主に NDL API を利用し、詳細情報への外部リンクや購入支援には、Amazon の API である AWS などを利用する。

2.3. 登録部および書籍 DB と蔵書 DB

登録部では、書籍 DB と蔵書 DB への登録を行なう。書籍情報の取得は、バーコードリーダーで ISBN を読み取り、NDL API を利用する。書籍 DB には、これを加工して利用する。ISBN のない書籍も扱うため、独自の書籍 ID を発行する。複数の著者、翻訳書の英名と和名、監修や翻訳などの著者の役割についても対応する。他の書籍関連サイトへのリンクとして URL も記述する。蔵書 DB では、同一の書籍については、冊数番号を発行し、セマフォとして用いる。ただし、所有者が異なる、一方が傷んだり書込みがある、などの場合は個々を区別する。また、所有者や収納位置を格納する。貸出中や持出禁止など、利用状況も保持する。本棚 DB では、本棚の位置と書籍の分類を格納する。蔵書は情報系分野がほとんどであるため、UNIX, DB, C 言語、

ネットワークプログラミング、資格試験など、独自の書籍分類を設定する(図 2)。本棚の位置と収納する蔵書の分類が対応するよう、適宜、分類を再編する。

2.4. 検索部

検索部では、蔵書のみ検索と、書籍全般の検索を行う。前者では、本システムの蔵書 DB を用い、収納位置を背表紙で視覚的に示すバーチャル本棚として提供する。後者では、NDL API で検索をかけ、表紙画像で一覧表示する。ここで、蔵書を優先的に表示する。蔵書については、利用状態、所有者、利用履歴などの情報も表示する。蔵書にない書籍については、AWS なども利用し、在庫状況、絶版なら中古販売情報など、購入検討の参考になる情報を表示する。

2.5. ユーザ区分と表紙部

ユーザの権限に応じて、教員/ゼミ生/サークル生/TA/一般教員/一般学生/学外者の区分を設ける(表 1)。学内者は、学内サーバのアカウントを認証 ID とする。教職員証と学生証には IC チップが搭載されているため、カードリーダーで認証を行うことができる。教員は、購入した個人蔵書を提供したり、関連予算での書籍購入を承諾する。そのため、全ての権限が与えられる。ゼミ生は、研究室に常駐していて、書籍を研究に利用するため、自分の机での長期の保管(移動貸出)が可能である。サークル生は、部費で購入した書籍を共有し、部室を利用しているため、連絡が取りやすい。そのため、中期の貸出と購入依頼が可能である。TA と一般教員は、事前に登録できるため、中期の貸出を認める。一般学生は、授業で指定された参考書を借りる際のみ一時的なユーザとなる。個別にユーザ登録を行い、短期の貸出のみとする。学外者は、頻繁に大学に来ることができないため、長期の貸出を認める。ユーザ ID には、申請したメールアドレスを認証に用いる。

2.6. 貸出 DB と司書部

貸出 DB では、貸出返却に必要な情報を格納する。個々の貸出ごとに、貸出 ID を発行する。借り手のユーザと蔵書の ID、貸出の開始と終了の日付を保存する。行為コードは、通常貸出、移動貸出、予約など、ユーザ権限に基づく貸出方法を表す。貸出ステータスは、貸出中や返却済のような貸出状態を表す。司書部では、ユーザの権限に応じた利用を管理する。貸出するときは、ISBN コードをバーコードリーダーで読み取り、蔵書を特定する。ISBN コードのない蔵書は、書籍名の一部などで、インクリメンタルサーチできるようにする。貸出の種類としては、通常の貸出の他に、指定日での予約貸出や、長期での移動貸出を認める。また、持出の禁止にも対応する。返却に際しては、期限内かどうかを記録し、延滞状況によっては、該当するユーザへの貸出制限を行う。

2.7. 購入部

購入部では、書籍購入の希望申請と注文実施を行う。前

者では、希望理由の選択肢(研究/授業/勉強/就職/課外/その他)を選んだり、個別の要望を備考に書く。新規の申請と、既にある申請への賛同が行える。書籍販売サイトと連携し、在庫状況なども反映される。後者では、教員やサークルの会計係が申請状況を考慮して、実際に注文する。Amazon API を利用して、カートに入れる購入直前の動作まで代行できる。発注が完了した書籍は、蔵書 DB に加えられ、注文済や入荷待などの利用状況が更新される。購入の希望から発注までの手順は、図 3 のようになる。



図 1 蔵書管理システムのモジュール構成と機能

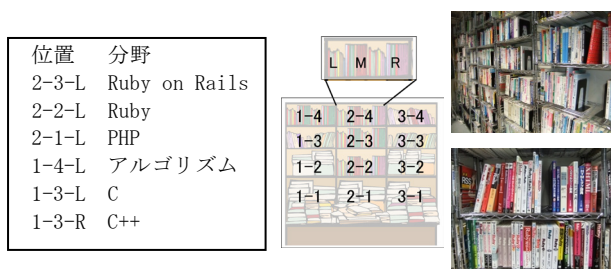


図 2 本棚構成と蔵書分類

表 1 ユーザの権限と特性

		期間	購入	登録	特性
Staff	研究室教員	長期	◎	○	個人蔵書を提供
Semi	ゼミ生	長期	○	○	自分の机に常駐
Club	サークル生	中期	○	○	部室を利用
Assist	TA	中期	×	×	業務に必要で貸出
Teacher	一般教員	短期	×	×	連絡が容易
Student	一般学生	短期	×	×	授業の指示で貸出
Guest	学外者	長期	×	×	普段は学外

3. 蔵書管理システムの機能の拡張

3.1. 貸出履歴や付加情報の活用

研究室規模の蔵書管理においては、その利用履歴や付加情報を学習支援に役立てることが考えられる(図 4)。研究室での図書の利用は、単に読むだけでなく、メモを残したり、内容を実践したり、精読が必要である。輪講では、レジュメを作成し、本文を参照したり引用したりする。数学や物理の教科書では、演習問題を解いて理解を深める。プログラミングの解説書では、例題プログラムを実行して確認したり、自分で類似のプログラムを作成する。内容のミスを指摘したり、状況の変化により修正や読替えも必要である。効率的に学習を進めるには、ゼミ生やサークル生で、これ

らの情報を共有することが有効である。また、先輩の先行研究を引き継ぐとともに、関連知識の学習過程を伝承することも重要である。ただし、安易に頼らず、段階に応じた閲覧を制限することも必要となる。以上を踏まえ、蔵書管理システムに分析部と記録部を追加した[7]。

3.2. 蔵書管理システムの分析部

分析部では、システムの履歴 DB を用意し、ユーザの利用を分析する(図 5)。それを基にして、貸出回数や貸出期間から人気の書籍を探すことができる。また、分野毎に時系列に対する分析も行なう。理由としては、情報系の書籍では他の分野の書籍に比べて、新しいことへの価値が大きいなど、分野によって新しさの価値が異なるからである。他に、個人の履歴を確認することができる。利用者と分野での絞り込みを行う。これを利用し、先行研究に携わった卒業生が利用した書籍を参考にすることができる。

3.3. 蔵書管理システムの記録部

記録部では、ユーザによる書籍への付加情報を扱う(図 6)。これには、書籍の評価、書評、紹介など、読書メモに相当する情報が含まれる。さらに、演習問題の解答やプログラムのソースコードなど、学習ノートに相当する情報も含まれる。学習ノートは、サークル生以上のユーザのみ閲覧可能にする。本研究室では、学習に利用した書籍の小レポートのページを作成することがある。そこで、書籍の情報を得る際に、その内容を自動的に取得できるようにする。それにより、本システムのために新規に要約を記入する必要がない。また、プログラミング関係の書籍では、演習問題が含まれていることがある。そこで、演習問題のソースコードも、システムを利用することで共有できるようにする。また、公開時の条件も設定可能にし、最初から全ての解答を閲覧可能にしない。テキストデータ以外の情報も付加できるようにすることで、UML など図解中心の学習にも対応する。他に、ユーザや書籍のキーワードに、先輩からの学習アドバイスを付加することもできる。例えば、書籍のキーワードに対して、推薦する利用順序などを示す。

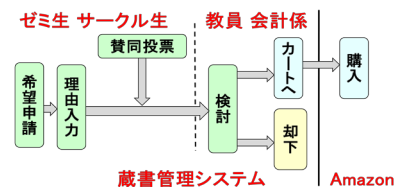


図 3 書籍販売サイトと連携した購入部

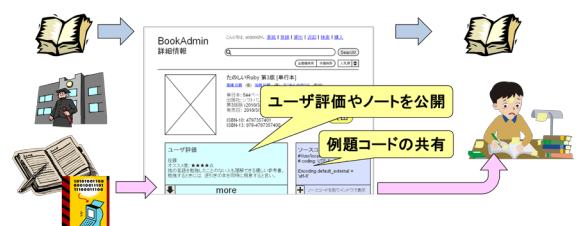


図 4 利用履歴の活用と注釈文書の閲覧

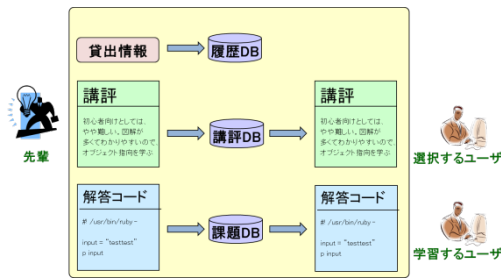


図5 蔵書管理システムの記録部

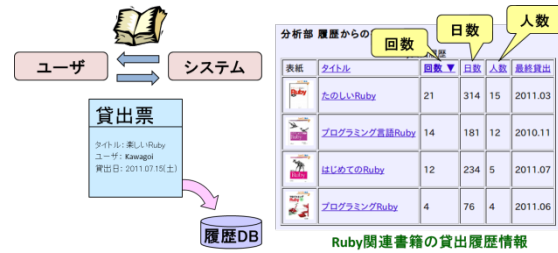


図6 蔵書管理システムの分析部

表2 情報系書籍の文献共有による精読の3つの型

	単独学習としての個読	通時学習としての引継	共時学習としての輪講
目的	知識や技能の習得	研究の引継 先輩の追体験	複数人で同時に 技術や知識をつける
書籍選択	人に勧められた本 評価の高い本	先輩の読んだ本	合議で決める
対象	初歩的な本 特殊な本	先輩の卒論・修論 必要技術の本	一般的な本 名著
ペース	自分で決める	先輩の記録を参考	計画的・定期的
残す文書	私的なメモ	積み上げ式のノート	相互閲覧のレジュメ 発表資料
要望	個人スケジュール管理 モチベーション維持	学会の論文の管理 システム文書	同時に複数人の管理 輪講の時間の管理

3.5.3 ポインタと配列はどう違う？

```
int arr[3] = {1, 2, 3};
int *arr_p;

arr_p = arr;

printf("%d\n", arr[1]); // arr[1]
printf("%d\n", *(arr_p+1)); // *(arr_p+1)
printf("%d\n", arr_p[1]); // arr_p[1]
```

ソースコード

図解

・ data[0] / *data でアクセスできるデータの位置の違い
 - 配列 : メモリ上に固定された位置
 - ポインタ : dataの指している位置

メモリのどこか a | b | c | d | e | W

str[] a | b | c | d | e | W

str* []

【練習問題3-5】
 文字列 "Hello World" をポインタで宣言し、出力時に配列として扱うプログラムを作成せよ。

練習問題

図7 情報系の書籍の構成

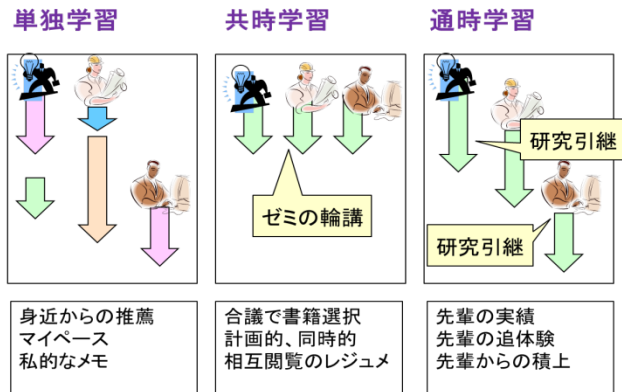


図8 情報系の書籍の文献共有による精読の3つの型

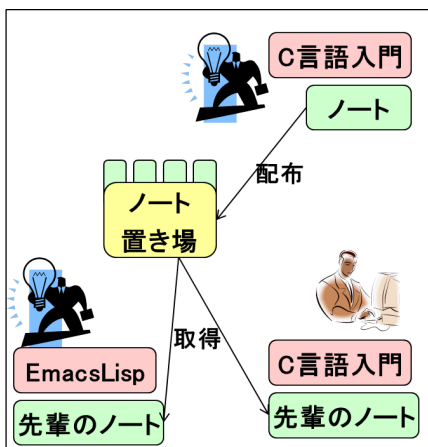


図9 単独学習による個読

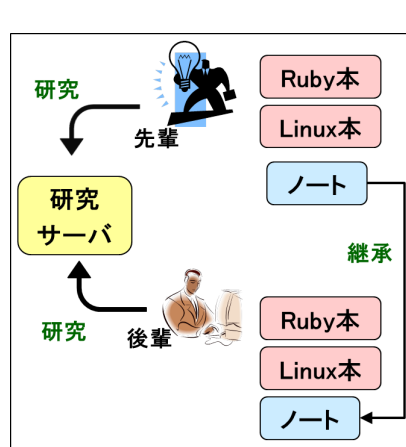


図10 通時学習による引継

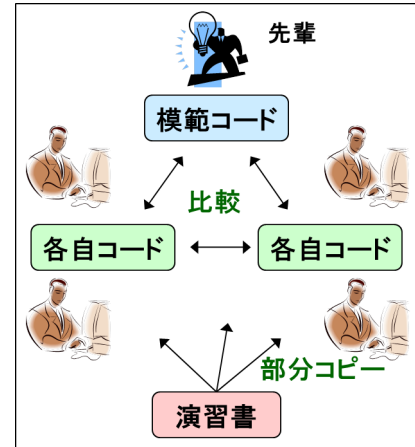


図11 共時学習による輪講

4. 蔵書管理と連携する学習支援

4.1. 情報系分野の文献の特徴

本システムで提案する学習支援では、特に情報系の分野の文献を対象とする(図7)。システムやプログラム言語の解説書は、バージョンに依存する機能や手順の説明が含まれることがあり、バージョンが異なると利用できないことがある。また、技術の変化に対応するためにも、新しい書籍が有用になる場合が多い。他にも、読者に一定の知識があることが前提とされている。そのため、自分のレベルと異なる文献を選択すると、簡単すぎて得るものが少なかったり、難しすぎて分らなかつたりする場合がある。

内容は構造的になっており、章単位での取捨選択も可能である。そのため、目次や索引が非常に重要になってくる。最近、オンラインの書籍販売サイトでは、目次などで全体の内容を公開しているところもある。また、引用なども多く、だれが書いても同様になるパブリックなものも多い。しかし、本の間違いに関する影響が大きく、記述が間違っていた場合は、システムやプログラムが正常に動作しない。

添付資料の種類も多く、例題や練習問題を含むプログラムのソースコード、プログラムの設定、プログラムの実行結果、概念の図解、UMLのようなフォーマットの決まった図、PCの画面画像などがある。特に、プログラムのソースコードや設定ファイルのように、文字データとして存在することに意味のあるデータは、オンライン上で配布されていることが多い。

4.2. 情報系分野の文献精読

情報系分野の文献の精読の特徴を整理する。文献に含まれる例題や練習問題を、一部入力して試したり、問題を解きながら進める。専門書の問題の場合には、数式を解いたりプログラムを実行しながら進めることもよくある。

読み方としては、章ごとに行ったり来たりしながら、理解を深めていくことがある。そして、同分野の文献と比べながら読むこともある。難しい場合には、初級の文献も参考にしながら精読を進める。一度理解したものでも、思い出すために読むことがある。

他の人と協力しながら精読することも多い。例えば、分からない部分を人に教えてもらいながら読むことがある。また、何人かで分担して読むこともある。さらに、他人のノートを参考に、問題を解いたり、解決したりすることも多い。理解した内容を他の人に話すこともよくある。加えて、指導をする者が参考になる文献を選択し、指導を受ける者に読ませたり、説明のときに利用することもある。

4.3. 情報系分野の文献共有による学習の分類

研究室規模の蔵書管理においては、その利用履歴や付加情報を学習支援に役立てることが考えられる。研究室での図書の利用は、単に読むだけでなく、メモを残したり、内容を実践したり、精読が必要である。輪講では、レジュメ

を作成し、本文を参照したり引用したりする。数学や物理の教科書では、演習問題を解いて理解を深める。プログラミングの解説書では、例題プログラムを実行して確認したり、自分で類似のプログラムを作成する。内容のミス指摘したり、状況の変化により修正や読替えも必要である。効率的に学習を進めるには、ゼミ生やサークル生で、これらの情報を共有することが有効である。また、先輩の先行研究を引き継ぐとともに、関連知識の学習過程を伝承することも重要である。ただし、安易に頼らせず、段階に応じて閲覧を制限することも必要となる。

以上を整理し、情報系の文献共有による学習のための精読を、表2のように3つの型に分類した(図8)。単独学習の個読は、身近からの推薦で書籍を選択し、個人がマイペースに学習を進めるものである。通時学習としての引継は、先行研究の引継などで、関連する書籍を、先輩の実績を基に、追体験のように精読していくものである。共時学習としての輪講は、ゼミなどで、合議で学習する書籍を選択し、計画的、同時に学習を進めるものである。

4.4. 単独学習としての個読

単独学習では、知識や技能の習得を目的に、1人で学習するものを想定している(図9)。他人の影響を大きく受ける通時学習・共時学習と異なり、他人からの影響は少ない。書籍の選択時には、自分のレベルに合っているかを確認する。加えて、他人の読み終えた評価を選択時の指針にしたり、身近からの推薦を参考にする。学習するペースは決まっておらず、マイペースに進める。対象は、誰もが学習するような初歩的な文献や、他の人が学習しないような特殊な文献である。ドキュメントとしては、私的な学習メモを残し、先輩のメモを参考に学習を進める場合もある。

4.5. 通時学習としての引継

通時学習では、研究の引継や、先輩の追体験を目的に、1人または複数人で学習するものを想定している(図10)。単独学習や共時学習と違い、先輩が築き上げてきた過去の実績からの影響を大きく受ける。書籍の選択時には、先輩が読んだものや、研究に必要なものを選ぶ。学習するペースは、先輩の体験を参考にする。対象の書籍は、初心者向けよりは、やや上の本である。

ドキュメントとしては、先輩から伝承したノートを参考にし、古い部分を修正したり、新しい内容を追記するなど、積上げ式で作成する。具体的には、書籍の参考になる箇所や、研究用の環境で異なる部分の注意、前提となる知識など開発・運用するために必要になる情報である。既になく先輩の作成したものを参照したり、今後来る予定の後輩に向けて残す。

4.6. 共時学習としての輪講

共時学習では、輪講として、複数人で同時に技術や知識を身に付けるため、複数人で学習するものを想定している

(図 11). 単独学習や通時学習と異なり, 常に複数人で同時に学習するため, 他人からの影響が最も大きい. 書籍の選択時には, 合議によって適したものを選ぶ. 学習するペースは, 事前に計画を決め, 定期的に行う. 対象の書籍は, 多くの参加者が勉強すべき一般的な書籍や, 名著と呼ばれる特定分野における定番の書籍であることが多い.

ドキュメントとしては, 発表担当者が作成する配布レジュメ, 提示スライド, 練習問題として出題する課題がある. また, 輪講の参加者が作成した課題の解答なども残す. 章単位で分担して作成することになる. 本論では, 文献共有による精読のうち, 共時学習としての輪講に焦点を当てる.

5. 共時学習としての輪講とコードレビュー

5.1. 現況の輪講による問題点

説明中心の輪講においても, 現状では多くの問題点がある. オンラインでの輪講の管理は, 簡単な Web ページや, Wiki を用いているものが多い. 事前準備の段階では, 進捗状況の不透明さに関する問題がある. そこでは, 輪講のスケジュールやレジュメの管理はできているが, 発表者の準備の進捗状況などは管理されておらず, 急遽中止となることもある. 加えて, 発表者が準備に苦戦していても, 他の参加者が気付いていないため, 助けることもできない.

輪講開催中の段階では, 発表へのフィードバックや, 問題演習をスムーズに行いにくい問題がある. 発表者は, 発表中にフィードバックが無いと, 発表時に説明を調整したり, その場のフォローが難しい. また, 次回の発表の調整をするため, 発表後にもフィードバックが求められる. 逆に, 聴講者には, 発表を遮っての意見を言いつらい場面もある. そのため, 聴講者同士で, 分かりにくいことを, 解決することも難しい. 問題演習では, 解答を回収したり, ユーザに応じた閲覧権限を付けることが難しく, 円滑な問題演習を行いつらい.

発表後の段階では, 復習する場合や欠席者へのフォローができていない. 例えば, 演習があった場合には, 記録が残っていないため, 欠席者は後から演習分の理解をすることが難しい. それに加えて, 参加者も復習したい場合に, 指摘箇所を覚えていないことも多い. 座学の場合には, 個人で作成したメモなども紛失することがある.

5.2. 輪講のモデリング

本研究が提案する輪講支援では, 輪講を以下のようにモデリングする(図 12). 最初に, 輪講の一連の流れを考える. 1 つのテーマを決め, 計画的・定期的に輪講会を開催する複数回のまとまりを輪講プロジェクトとする. 決められたテーマに沿って行なわれる, 1 回の輪講を輪講セッションとする. 大きな流れとしては, 輪講プロジェクトの提案, 輪講セッションの事前準備, 輪講セッションの開催中, 輪講セッションの事後処理となる.

輪講プロジェクトの提案時には, 参加者のカテゴリ, 対象文献の分野, 期間や方法などを仮決めする. 参加者のカテゴリでは, ゼミ生やサークル生といった対象とする参加者の範囲を選択する. 対象文献の分野の選択では, Ruby の解説書, 教育支援システムの論文のように, 文献まで決めずに, 分野を決める. 期間は, 開始・終了予定の時期を考える. 方法は, 講義のみや, 演習付きなど, 形態を選択する. そして, 仮の案を, 参加希望者と議論し, 参加者や文献, 期間, 方法を確定する. 輪講プロジェクトの情報が確定したら, 輪講セッションごとの, 分担, 日時, 担当者の一覧を作成する. このとき, 全ての輪講セッションの詳細を決める必要はなく, 今後数回分を決めるだけでもよい. 全ての輪講セッションが終了したら, 輪講プロジェクトを終了する.

輪講セッションのおおまかな流れは, 輪講セッションの事前準備, 輪講セッションの開催中, 輪講セッションの事後処理である. 輪講セッションの事前準備は, 発表者が, 文献の精読, 精読ノートの作成, 提示資料・課題資料の作成をし, 進捗状況の報告と作成物の提出を行う(図 13). 精読ノートは, 文献を精読した際に作成するメモである. 提示資料は, 講義時に用いるプレゼンテーション用のスライドである. 課題資料は, 参加者に解答させるための練習問題である. 追加で, 発表練習と, 他者への確認を行なう場合がある.

輪講セッションの開催中に, 発表者は提示資料を用いて発表をする. 発表中に課題資料から演習を行う場合もある. 開催中に参加者は, 発表を聞くだけではなく, 発表に理解度を示したり, 発表に意見をしたり, 課題資料を解くこともある. 他者へ影響のある行動ではないが, 参加者はそれぞれ発表中に, 理解を深めるため個別のノートをとる. 課題資料の問題が, プログラミング問題の場合には, 提出後にコードレビューをすることもある.

輪講セッションの事後処理では, 発表に対しての講評を記入する. 聴講者は, 発表者の作成資料やプレゼンテーションに対して, 意見やアドバイス, 理解度をフィードバックする. 発表者は, 聴講者からの講評をもとに, 反省する.

5.3. コードレビュー

本研究で扱うコードレビューとは, チーム内のメンバが作成したコードに対して, 本人を交えて, チーム内・外のメンバで閲覧し議論するものとする(図 13). 各自が作成したソースコードに対して, 問題点や改善点を指摘したり, 本人に助言や指導を行う. ユーザの役割は, コードを作成してレビューされる者をレビューイとし, コードを閲覧してレビューするものをレビューアとする.

対象とするソースコードを検討するために, ソースコードを模範コード, 例題コード, 解答コード, 自作コードの 4 つに分類した. 模範コードは, Linux のカーネルのように, 輪講の対象となる教科書的なコードである. 例題コードは,

ソースのプログラムのように、概念や算法の説明となるコードである。解答コードは、各自が作成した練習問題のコードで、問題には共通理解があるが必ずしも適切とは限らない。自作コードは、各自が自由に作成したコードで、設計が自明とは限らず、設計と実装の対応が不明である。ここから、コードレビューに適しているものは、解答コードと自作コードであると考えられる。理由としては、模範コードや例題コードは修正が必要な場合が少なく、理解して参考にするコードリーディングの範囲だと考えたからである。

コードレビューの人間関係としては、3つに分類した。1つは、下級者のコードを主に上級者が指摘するもの、同程度のメンバで作成者と一緒に検討し合うもの、上級者のコードを下級者も確認するものの3つである。1つめ、2つめは、コードレビューとして適しているが、3つめのはコードリーディングに近いので、本研究では1つめと2つめのレビューを想定する。

一般に、コードレビューで行われる行為やメリットには、教育的なもの、成果的なものがある。教育的な側面では、作成者のコードをメンバ全員で共有する、作成者による意図・目的や設計との対応の説明、レビュアーから作成者へ不明点などの質疑応答、メンバ相互による問題点や改善点の指摘、レビュアーへの助言や指導などがある。

成果的な側面では、暗黙知の明示化と意識化、コーディング規約や開発ツールの標準化、各自のコーディング技能や特性の相互理解、プロジェクトのメンバとしての相互のコミュニケーション、システム開発の目標や理念の確認がある。本研究では、前者の教育的側面に特化して支援する。

5.4. 関連するツールや研究

関連する既存の支援ツールは、コードレビューシステムとライブコーディングシステムに分けられる。前者は、主に非同期式で、レビュアーが事前に書いておいたコードを、後からレビュアーが閲覧し、コメントするものである。ソースコードのバージョンごとに、その差分にレビューする Review Board[8]や Rietveld[9]がある。また、複数のバージョンに対して設計や実装の方針などを指導するものとして、Gitlab[10]、RhodeCode[11]がある。後者は、同期式で、レビュアーが書いているコードをライブ中継としてレビュアーに公開し、リアルタイムにチャットなどでコメントしていくものである。代表的なものに、Live coder[12]がある。これらは、上級者から初心者への指導の側面が強く、本研究が目的としているピアレビューの要素が弱い。

オンラインでのプレゼン支援システムとしては、次のようなものがある。平井らは、ビデオ学習を支援する協調学習システム[13]を挙げている。これは、グループ化と同期・非同期を使い分けることで、ビデオ教材の理解を支援している、ビデオ教材を見ながら、チャットやビデオへの理解度確認バーで相互学習を行える。そして、Yiyuan らの Web 型遠隔教育システムのオンラインプレゼンテーション[14]

では、スライドを使い、オンラインでプレゼンテーションすることを支援している。発表者の写っているビデオとスライドの画面を共有し、プレゼンテーションを Web を通じて行うことを可能にしている。本システムでは、スライドのみの配信としているが、輪講に焦点を当て、同じページから問題の出題などを統合的に扱えるところが異なる。

6. 輪講支援ツールの設計と機能

6.1. 輪講支援ツールの設計

蔵書管理システムをベースに、文献共有による学習支援のうち、共時学習としての輪講を支援するツールを提案する。対象とする分野は、主に情報系の専門書であり、恒偽だけでなく、演習を伴う輪講を想定する。特に、プログラミングの解説書を題材とする場合は、解説書に掲載された例題プログラムや、輪講の参加者が作成した練習問題の解答プログラムをコードレビューする。ツールは、Ruby on Rails で実装する。ツールの利用形態について、図 14 のユースケース図で整理した。

6.2. 輪講支援におけるユーザのロール

輪講支援システムでは、利用者は、フェーズごとに権限や役割が異なるものもあり、全部で 15 種に分類した。フェーズごとに、システム全体、プロジェクト、セッション、コードレビューでロールを分ける。理由は、複数のプロジェクト参加者が、交代に講師を担当する性質上、毎回のセッションでは、プロジェクトの責任者だけが権限を持つわけではなく、それぞれのセッションの責任者が必要になるからである。

輪講プロジェクトに関わるロールは、主催者、支援者、参加者、監修者である。主催者は、輪講プロジェクトを提案する人である。開催している輪講プロジェクトの責任者となり、プロジェクト管理の多く権限を持つ。参加者の上位権限に当たるので、参加者の有する権限を与えられる。支援者は、輪講プロジェクトに賛同し、提案会議で発言する人である。輪講プロジェクトが提案段階から開催段階に変更されたときに、参加者へ変わる。参加者は、輪講プロジェクトに実際に参加する人である。監修者は、輪講プロジェクトを全般的に指導する人で、教師や上級生が担当する。文献や場所などを提供したり、許可を与える。

輪講セッションに関わるユーザは、ロールとしての分類したものと、出欠ステータスで分類したものがある。ロールとして分類したものには、発表者、聴講者、傍聴者、解説者、司会者、記録者がある。ロールには、輪講セッションの内容に影響するものと、進行に関係するものがある。発表者は、輪講セッションで発表を担当する人で、内容と進行に関わる。輪講セッションごとに異なる。複数人で担当することも可能である。発表用の提示資料や、練習問題としての課題資料の作成も行う。聴講者は、輪講プロジェ

クトの参加者で、その輪講セッションで議論に加わる人である。内容に関わる。事前に課題資料を作成・提出することもある。傍聴者は、輪講プロジェクトの参加者ではないが、輪講セッションを傍聴する人である。内容に関わる。解説者は、輪講セッションで助言や解説をする人で、内容と進行に関わる。司会者は、輪講セッションの進行を管理する人で、進行に関わる。発表者が牽引することも多い。記録者は、輪講セッションでの発言や行為を整理して編集する人で、進行に関わる。議事録の提出なども担当する。

出欠ステータスで分類したものは、分類した主席者、欠席者、臨席者がある。出席者は、輪講プロジェクトの参加者のうち、その輪講セッションに出席した人である。欠席者は、輪講プロジェクトの参加者のうち、その輪講セッションに欠席した人である。後に、システムを用いて復習する。臨席者は、輪講プロジェクトの参加者ではないが、その輪講セッションに臨席した人である。

コードレビュー時に、輪講セッションのルールに追加されるルールとして、レビュイーとレビューアがある。レビュイーは、コードレビューを受ける人である。レビューアの意見を聞きながら、プログラムを修正する。レビューアは、コードレビューをする人である。レビュイーのコードや、行動を見ながら、意見する。

6.3. 輪講支援の講義ページの機能と GUI

講義ページでは、提示資料としてパワーポイントや PDF などのスライドを用いての発表時の利用を想定する。図 16 に提示資料を用いた講義ページの GUI を示す。画面の構成として、提示資料の 1 ページと、参加者のコメント、聴講者の個別ノートがある。図 15(a)は、輪講エントリの TOP ページである。

輪講エントリのトップページでは、直近の輪講の内容や、各機能へのリンク、輪講に用いる文献情報、参加者一覧など、輪講プロジェクトの参加者に必要な機能を統合的に扱えるようになっている。

図 15(b)の講義ページでは、発表者が表示している提示資料のページが、聴講者の画面にも同期される。そして、全員がスライドの内容に対して、コメントとして、指摘や意見を投稿できる。その投稿は、リアルタイムで発表者、聴講者の画面に反映される。提示資料の同期を止めることで、聴講者の意志で発表者の表示しているページとは別のページを閲覧したり、意見することが可能である。また、参加者それぞれが、個別ノートを作成することができ、各自がシステム上で要点を整理することができる。

セッションに参加できなかった欠席者も、後に復習ページを利用することで、輪講の流れを追うことができる。復習時には、提示資料の同期がないため、手動でスライド送りをすることになる。復習時でも、コメントがスライドに関連付けられているため、講義中に出た議論を追体験することが可能である。

6.4. 輪講支援のコードレビューページの機能と GUI

図 15(c)に、レビューする解答コードの一覧ページ、図 15(d)に、コードレビューページの GUI を示す。解答コードの一覧ページでは、提出したユーザの一覧が表示される。入出力サンプルのある問題に関しては、解答の正誤によって、ユーザを区別して表示する。解答者のリンクの上にカーソルを合わせることで、解答コードの一部をポップアップで表示することができる。解答者の名前をクリックすることで、対象をレビュイーとしたコードレビューページへ移動する。

コードレビューページでは、コードを査読されるレビュイー側のコーディングページと、査読するレビューア側のレビューページを扱う。レビュイーの画面では、エディタ部分でソースコードを作成・変更することができる。Ajax を用いて同期処理を行なっているため、変更結果は常にシステムに送信される。その結果を受け、レビューアはリアルタイムにレビュイーのコードの変更を確認することが可能である。レビュイーがコンパイル・実行した結果も同様にリアルタイムにレビューアの画面上の結果表示部分に反映される。コードレビューの参加者は、コードに対してコメントを投稿することができる。コメントは、投稿時に解答コードの位置を選択することができる。コメントは、解答コードの右側に表示され、クリックすることでコメントをつけた時のコードが表示される。

レビューア側のページでは、エディタ内のコードの編集機能は停止になっている。復習時には、解答コードとコメントの履歴を閲覧しながら、コードレビューを追体験することができる。

6.5. 輪講支援システムの試用実験とユーザ評価

試作版の輪講支援システムにおいて、特にコードレビューに関する機能について、試用実験を行った。まず、予備実験として、解説書「R」を用いて、数回 30 分の輪講プロジェクトを実施した。参加者は 10 名、期間は 4 ヶ月程度で計 19 回の輪講セッションを実施した。書籍の全 17 章のうち、14 章分で約 200 ページを対象とした。参加者の負担や日程も考慮し、各自の分量を抑え、1 日で複数の輪講セッションを開催するときもあった。コードレビューの対象としては、R 言語の文法事項、簡単な数学や統計の計算などである。各自のノート PC に R 言語の実行環境を導入してもらい、解説書の実行例や例題プログラムを提示しながら、各自で確認するという形式である。予備実験では、実際にシステムを利用しながら、参加者の意見を聞き、機能への要望を収集した。

次に、試用実験では、輪講の演習におけるコードレビューの機能に絞って実施した。題材は、C 言語やシェル・コマンドの簡単なプログラムを Ruby 言語の初心者へ Ruby スクリプトとして書き直してもらい、それを皆でレビューす

るという形式である。実施中に意見を聞くとともに、紙面で自由記述のアンケートを実施した。

システムを利用した感想としては、次のようになった。紙のコードレビューや GoogleDocs を利用してコードレビューをしていたユーザからは、「以前のコードレビューよりも、本システムを使う方がはるかに便利だ」などの強い肯定的な意見を得ることができた。他には、「他人のコードを見ることによって、不備や意見をコメントするのはいい経験であった。」「質問を受けることで、無意識に書いていた部分を明確に考えることができた。」といった学習に対する効果を感じている意見も多く得ることができた。また、「レビュー時のコメントが残るので後からでも思い出しやすいと感じた」といった復習に対しても肯定的な意見もあった。

加えて、システムに対する要望も多く得ることができた。機能の要望では、「レビューもコンパイル・実行を自由に行えるようにしてほしい」といった意見が多かった。「コードレビューを含む、輪講セッションの進行を伝えるタイムラインなどが欲しい」といった意見も多く見られたので完全オンライン化するために、機能として追加する。

加えて、ユーザビリティでの意見としては、コメント表示の最適化を求める意見も複数あった。「コメントの種別として、質問、意見、助言などのステータスを付けて色分けが欲しい」や「同意見の場合に「いいね」ボタンが欲しい」などの要望もあった。コメントを見やすくする工夫が今後の課題である。

7. おわりに

書籍関連サイトと連携した研究室規模の蔵書管理システムを提案している。個人所有やサークル共有の書籍を利用者が共同で管理する。外部の書籍関連サイトと連携し、諸作業を効率化する。蔵書の登録には、書籍検索サイトから書籍情報を取得し、蔵書にない書籍の検索も可能にする。利用者にユーザ区分を設け、それに応じた権限を与え、サービスの内容も変更する。貸出と返却だけでなく、購入の希望を受け付け、他のユーザの賛同と予算に応じて、書籍販売サイトからの購入を支援する。

本システムの管理対象は情報系の書籍であり、利用目的が知識や技能の習得のための精読である。そのために、貸出履歴や利用者からの付加情報を活用し、書籍利用の活性化を図る。そこで、蔵書管理に基づく文献共有による学習支援を新たに提案した。まず、現行のシステムに記録部と分析部を導入した。

さらに、情報系書籍の特徴を整理し、文献共有による精読を、単独学習としての個読、通時学習としての引継、共時学習としての輪講の3つの型に分類した。特に、共時学習に着目し、輪講をプロジェクトと捉え、セッションの進行としてモデリングした。これに基づき、輪講支援システムの機能を検討し、試作版を実装した。プログラミングの

解説書に多く見られる例題プログラムの理解や、練習問題としてのプログラミングを支援するため、同期式でピア型のコードレビューを取り入れた。試作版による試用実験では、ユーザ評価として多くの意見が収集できた。

今後の課題を述べる。蔵書管理システムでは、文献共有による学習支援を強化するため、記録部と分析部の機能を充実させる。学术论文や各種資料の管理も行い、研究室の他のグループウェアとも連携する。電子書籍への対応も検討する。将来的には、他の研究室にも導入してもらい、蔵書の一時的なレンタルなど、システムの相互運用も目指す。

文献共有による学習支援としては、輪講支援システムの本格的運用を目指す。ユーザ評価による意見を分析し、機能とユーザビリティを向上させる。また、蔵書管理システムの記録部と分析部を活用し、単独学習としての個読、通時学習としての引継についても、より具体的な支援を検討し、機能を実現する。

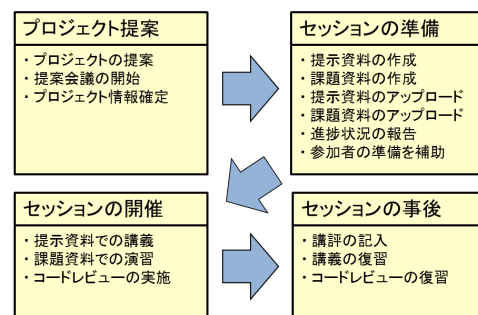


図 12 輪講プロジェクトの進行モデル

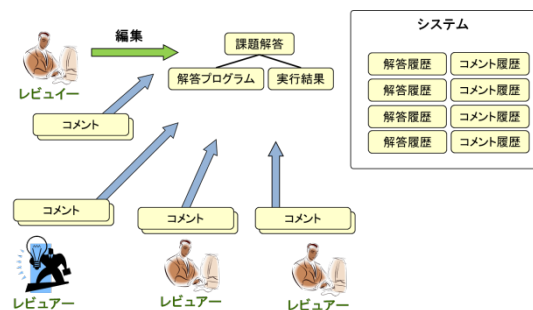


図 13 コードレビューの概念モデル

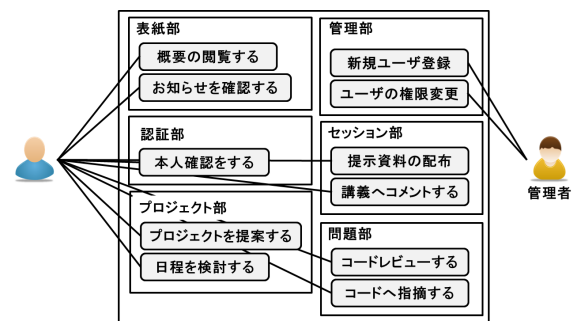
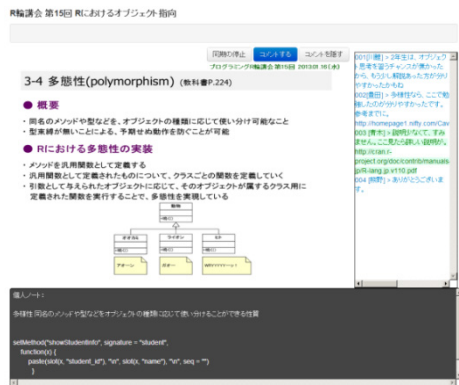


図 14 輪講支援システムに関するユースケース

プログラミングR 輪講TOPページ

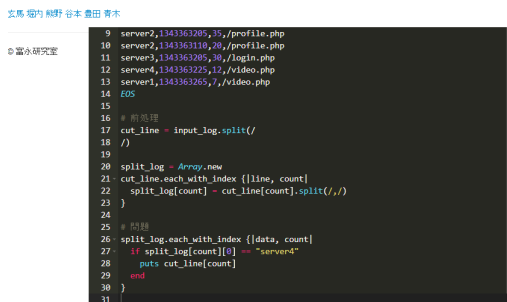


(a) 輪講エントリ画面

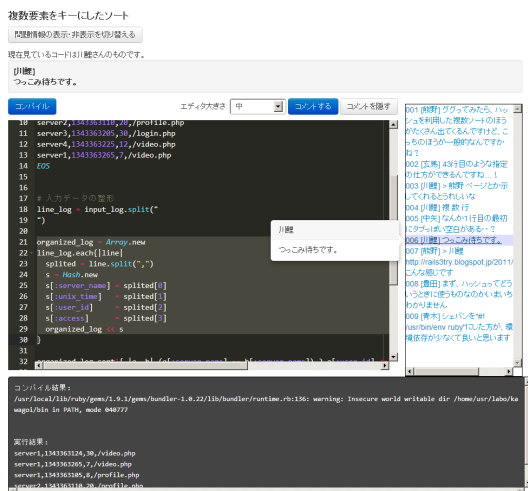


(b) 講義画面 (提示資料の閲覧画面)

解答一覧



(c) 解答一覧画面



(d) レビュー画面

図 16 輪講支援システムの GUI

参考文献

- 1) 川鯉光起, 中矢誠, 富永浩之: 書籍販売サイトと連携した研究室規模の蔵書管理システムの試作, 教育システム情報学会 第36回全国大会 講演論文集, pp.218-219 (2011).
- 2) 川鯉光起, 富永浩之: 書籍販売サイトと連携した研究室規模の蔵書管理システム -DB 設計と基本機能-, 情処研報, Vol.2011-CE-108, No.2, pp.1-10 (2011).
- 3) 川鯉光起, 富永浩之: 書籍販売サイトと連携した研究室規模の蔵書管理システム -試作とユーザ評価-, 信学技報, Vol.141, No.335, pp.41-46 (2011).
- 4) Amazon.co.jp, <http://www.amazon.co.jp/>.
- 5) 楽天 Books, <http://books.rakuten.co.jp/>.
- 6) 国立国会図書館サーチ, <http://iss.ndl.go.jp/>.
- 7) 川鯉光起, 富永浩之: 研究室規模の蔵書管理と文献共有による学習支援の検討, 教育システム情報学会 第37回全国大会 講演論文集, pp.356-357 (2012).
- 8) Review Board, <http://www.reviewboard.org/>.
- 9) Rietveld, <http://code.google.com/p/rietveld/>.
- 10) Gitlab, <http://gitlab.org/>.
- 11) RhodeCode, <http://pythonhosted.org/RhodeCode/>.
- 12) Live coder, <http://kagamin.net/hole/lc/>.
- 13) 平井和人, 小山達也, 米澤宣義, 宮寺庸造: ビデオ学習を支援する協調学習システム, 信学技法, Vol.106, No.437, pp.43-48 (2006).
- 14) Yiyuan Gong, 佐藤 博, Rahman Mahfuzur, 上野 晴樹: Web 型遠隔教育システムのオンラインプレゼンテーションツール, 信学技報, Vol.106, No.383, pp.27-31 (2006).