

## 情報系分野の解説書の輪講支援システムにおける

## プログラム問題のコードレビュー機能の検討

Code Review Functions of Programming Problems  
in a Supporting System for Lectures and Exercises in Turn豊田 竜也† 川鯉 光起† 富永 浩之†  
Tatsuya Toyoda Koki Kawagoi Hiroyuki Tominaga

## 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情報系・数理系を中心とした図書を共同で管理している。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。一方、書籍関連サイトでは、各種のAPIを公開し、サイトの書籍情報を外部から利用できるサービスが行われている。先行研究では、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]。システムのモジュール構成は、図1の通りである。登録部では、バーコードリーダーでISBNを読み取り、国立国会図書館のNDL APIを利用する。検索部では、仮想本棚を表示する。司書部では、ICチップ搭載の身分証で認証し、ユーザの権限に応じたサービスを提供する。購入部では、書籍販売サイトへの発注のリクエストを管理する。

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、全体または必要な箇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、情報系分野の文献を題材とし、演習を伴う輪講を支援するシステム BoocLeas を開発している。特に、プログラム問題の解答を皆で検証するコードレビューの機能を検討する。



図1 蔵書管理システムのモジュール構成

## 2. 蔵書管理と情報系分野の文献共有

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、思い出すために読むことがある。

また、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。

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。そのため、履歴部と注釈部の2つのモジュールも追加している。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。

## 3. 情報系分野での輪講

輪講は、何人かのグループで、学術書や技術書を読み進めていくものである。質や量の点で、独習するには困難な文献を何人か持ち回りで分担し、相互に教え合うことで、効果的かつ効率的な学習に繋げる。輪講では、担当者が文献の内容を要約し、配布資料や提示資料を用いて、他のメンバに説明していく。他のメンバは、不明な点を質問したり、問題提起に対して意見を述べたりする。さらに、演習として、文献に掲載された練習問題や担当者による出題を皆で解いて論評する。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、

† 香川大学, Kagawa University

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。

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る、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。

#### 4. 輪講の進行

本研究が提案する輪講支援では、輪講の進行を図 2 のように捉える。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。

輪講セッションの大まか流れは、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。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。課題資料から演習を行う場合もある。参加者は、発表を聞くだけでなく、発表への理解度を表明したり、発表内容に意見したり、課題資料を解くこともある。参加者は、個別の行動として、理解を深めるため、各自のノートをとる。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基に、反省する。

#### 5. 輪講支援ツールの設計

本研究では、輪講の支援システム BoocLeas を開発中である。対象とする分野は、主に情報系の専門書であり、講義だけでなく、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。利用者は、システム全体、プロジェクト、セッションのそれぞれにおいて、適切なロールを持つ。複数のプロジェクト参加者が、交代で講師を担当するため、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のものには、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。発表者が兼任することも多い。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、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。

## 6. 輪講支援ツールの機能

輪講セッションにおける講義中は、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。画面の構成は、提示資料のウィンドウ、参加者のコメント、聴講者の個別ノートがある。

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。

講義ページ(図 3)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、個別の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ようになる。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。

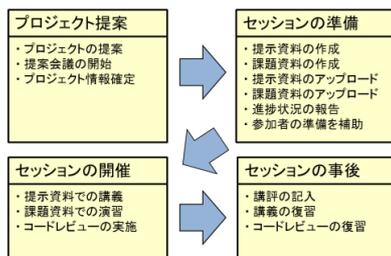


図 2 輪講プロジェクトの進行モデル



図 3 輪講支援の講義ページ

## 7. プログラムのコードレビュー

一般に、情報系分野の輪講においては、プログラムのソースコードを読む機会が幾つかある。対象となるソースコードには、模範コード、例題コード、解答コード、自作コードの 4 つが考えられる。模範コードは、Linux のカーネルのように、コード自身が教科書として読まれるようなコードである。例題コードは、ソートのプログラムのように、概念や算法の説明として挙げられたコードである。解答コードは、練習問題の解答として各自が作成したコードである。題意は明確であり、共通理解の下にあるが、必ずしも解答として適切とは限らない。自作コードは、各自が自由に作成したコードである。本人以外には、仕様や設計が自明とは限らず、実装との対応も不明である。

模範コードと例題コードを読むことは、特に修正の必要な場合が少なく、ひたすら精読して理解するコードリーディングにあたる。解答コードと自作コードを読むことは、より批判的な精査が必要であり、そのフィードバックが重要となるコードレビューにあたる。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プログラムのソースコードに対して、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 4)。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イー、コードを閲覧してレビューするものをレビューアーと呼ぶ。コードレビューにおける対象者の関係を、3 つに分類した。1 つ目は、下級者のコードを主に上級者が指摘するもの、2 つ目は、同程度のメンバで作成者と一緒に検討し合うもの、3 つ目は、上級者のコードを下級者も確認するものである。最初の 2 つは、コードレビューとして適しているが、3 つ目はコードリーディングに近い。本研究では 1 つ目と 2 つ目のレビューを主に想定する。

一般に、コードレビューで行われる行為には、教育的なもの、成果的なものとの 2 つの側面がある。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、レビュー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、レビュイ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、前者の教育的な側面に特化して支援する。

## 8. 輪講支援ツールのコードレビュー機能

BoocLeas のコードレビュー機能および利用手順について紹介する[3]。まず、輪講において、担当者がプログラム問題を提示する。それに対して、参加者全員が解答としてプログラムを作成し、システムにソースコードをアップロードする。担当者は、ソースコードの一覧から適当なものを選び、選ばれたメンバがレビュイーとなり、残りの参加者がレビューアーとなる。

図 5 に、レビューする解答コードの一覧ページ、図 6 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

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、対象をレビュア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アー側のコーディングページと、査読するレビュアー側のレビューページを扱う。レビュアーの画面では、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、レビュアーはリアルタイムにレビュアーのコードの変更を確認することが可能である。レビュアーがコンパイル・実行した結果も同様にリアルタイムにレビュアー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、解答コードの具体的な箇所を指定してコメントすることができる。コメントのログは、解答コードの右側に表示される。これをクリックすることで、コメントされた時点の以前の解答コードを参照することができる。

レビュアー側のページでは、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、コードレビューを体験することができる。

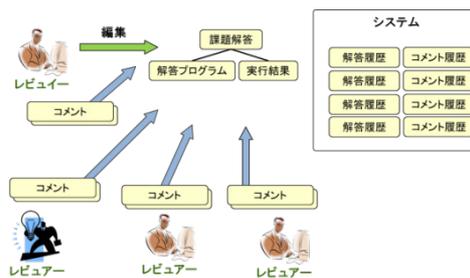


図4 コードレビューの概念モデル

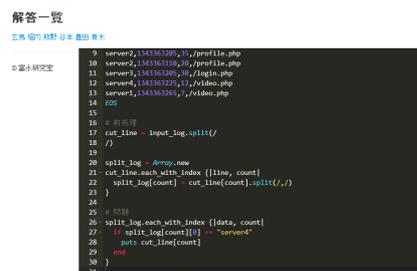


図5 コードレビュー機能の解答一覧画面

## 9. 試用実験によるユーザ評価

システムのコードレビューの機能に絞って試用実験を行った。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。参加者は7名で、C言語は既習であるがRuby言語の初心者である。期間は2ヶ月で計6回、毎回30分程度の輪講を実施した。利用者には、自由記述のアンケートに回答してもらった。

システムを利用した感想としては、「紙面やGoogleDocsよりも便利である」との好評価が得られた。「他人のコードを見て、不備や意見をコメントするのは良い経験だった」「質問を受けることで、自分が書いたコードを明確に意識した」「レビュー時のコメントが残る、後から思い出しやすい」といった、学習効果への肯定的な意見も多かった。

システムの機能については、「レビュアーもコンパイル・実行を自由に行えるようにしてほしい」「コードレビューを含む、輪講全体の進行を伝えるタイムラインが欲しい」などの要望があった。ユーザビリティについては、「コメントの種別(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」「同意見の場合に、いいねボタンが欲しい」などの要望があった。

## 10. まとめ

本研究では、情報系分野の文献を対象とし、演習を伴う輪講を支援するツールBoocLeasを開発している。本論では、特に、プログラム言語やアルゴリズムの解説書を対象とし、例題コードの解説や演習問題の解答コードをレビューする同期型のコードレビュー支援の機能を議論した。今後の課題として、試用実験の意見を基に、ツールの機能を拡張し、ユーザビリティを向上させる。

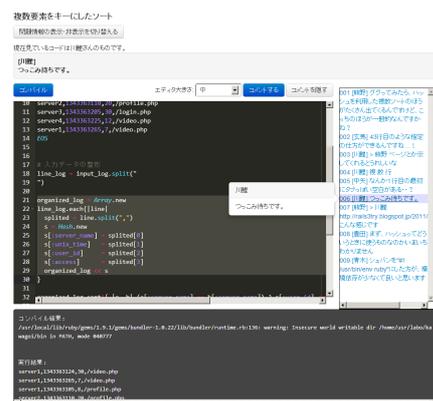


図6 コードレビュー機能の画面

## 参考文献

- [1] 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).
- [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 -試作とユーザ評価-", 信学技報, Vol.141, No.335, pp.41-46, (2011).
- [3] 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 -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-", 情処研報, Vol.2012-CE-119, No.20, pp.1-10, (2013).