

# ゲーム戦略を題材とする応用プログラミング演習における リファクタリングを動機づける自動的なフィードバック手法の提案

清水 赳<sup>†</sup> 富永 浩之<sup>‡</sup>  
香川大学<sup>†</sup> 香川大学<sup>‡</sup>

## 1. はじめに

本研究室では、ゲーム戦略を題材とするプログラミング演習を実施している。カードゲーム戦略を題材とするC演習のWinT、ボードゲーム戦略を題材とするJava演習のWinGの2つの演習を提案し、実践している[1]。各演習の支援システムを大会運営サーバにより提供している。受講者は、ゲーム戦略のコードを作成し、支援システムに対して提出する。支援システムでは、コードの実行を行い、結果や順位を全員に対して公開する。提出は何度でも受け付け、戦略の改良とコードの継続的な修正を促進する。

本研究では、コードの不吉な臭いに着目し、戦略コードの継続的なリファクタリングを動機づける自動的なフィードバック手法を提案する。本手法により、コード品質を考慮したソフトウェア開発を意識させる。また、本手法を実現し、教員の指導に対する負荷の軽減を目指す。

## 2. ゲーム戦略演習の概要

本研究室では、C演習のWinTとJava演習のWinGを実践している。各演習は、ローカル開発環境と、大会運営サーバによる支援システムを提供し、大学情報系学科3年生の必修科目の課題として、教育実践を行っている。

WinTは、ポーカー戦略を題材とする得点型の演習である。手札の5枚を山札と交換しながら、9種類の手役のうちの1つを作る戦略を作成する。1回のテイクでのチェンジ数を定めておき、十分にシャッフルされた1つの山札でテイクを繰り返す。そのため、受講者は、残りの山札を考慮した戦略の作成を行うこととなる。

WinGは、ボードゲーム戦略を題材とする対戦型の演習である。五目並べに石取りを加えたゲームの五五を採用している。連と取という2つの勝利条件があり、それぞれに攻撃と防御の優先度が考えられ、初心者でも戦略の個性が出やすい。受講者は、与えられた盤面に対して、どの枱に石を置くか決定する戦略を作成する。

## 3. 演習の教育目標

本演習では、2つの最終的な教育目標を設定している。1つは、問題解決の手段としてのプログラミング技能の習得である。これは、アイデアをプログラムとして表現するために必要である。本演習では、戦略作成の過程に必要な、配列の操作やソート、パターンマッチ、シミュレーション等を通して、これらの技能を定着させる。

もう1つは、継続的なプログラムの改善を行うための、プログラミング作法の習得である。これは、ソフトウェア開発手法に基づき、継続的な改善を行う業務としてのプログラミングに必要である。これらは本演習を通し、補助関数によるコードの整理、テスト、パラメタ調整の効率化や自動化、バージョン管理、リファクタリングを体験させることで身に付けさせる。

## 4. 演習運営の現状

本研究室では、ゲーム戦略の演習について、WinTは2010年度、WinGは2011年度から教育実践を行っている。受講者は、毎年40名程度である。2019年度は、各演習に43名が参加した。演習の提出総数は、毎年、WinTが1200~2000件、WinGが700~800件程度である。そのため、教員やTAが全戦略をレビューすることは困難である。現状、支援システムから、受講者に、コードメトリクス(CM)を提示して、リファクタリングを促している[2]。しかし、学生には解釈が困難であることがある。また、受講者の作成する各戦略で全く異なった設計や書法となるため、指導のために、コードの全部を閲覧する必要がある。そのため、教員やTAの指導に対する負荷は高い。

## 5. 自動的なフィードバック手法の提案

本研究では、コードの不吉な臭いに着目し、戦略に対して自動的なフィードバックを行う手法を提案する。不吉な臭いとは、リファクタリングの必要性を示す不吉な兆候のことである[3]。

本手法は、戦略コードのいくつかの臭いの定量的な検出を試み、受講者と教授側に提示する。このことで、受講者は、戦略の改善点を認知でき、リファクタリングが動機づけられる。また、教授側は、指導の基礎情報として、コードの着目すべき点と、問題の予兆を認知できる。このことで、コードレビューの範囲が限定され、教

授側の負荷が軽減される。また、臭いに即した指導の方針も決定でき、指導内容の充実が図れる。指導を行うため、教授側によるコードに対するコメント機能も備える。

## 6. フィードバックシステムの設計と機能

本手法を実現するシステムは、不吉な臭いの検出、教授側と受講者に対する情報提示、教授側に対する要指導な戦略の提案、教授側からのコードに対するコメントの各機能で構成される(図1)。開発は Ruby on Rails を用いている。

まず、受講者は、システムに対して戦略を提出する。システムは、ソースコード静的解析ツール Adlint[4]を用いて、戦略コードの CM を計測する。これを用い、臭いを検出する。臭いの名称と検出された箇所等を記録する。これらの処理の後、教授側と受講者に対して、戦略コードと併せ、検出された臭いを提示する(図2)。この時、教授側には、臭いの数に応じて指導の優先度も提示する。教授側は、必要に応じ、システムを用いてコードに対してコメントを行う。

現状、本システムは、C言語のみ対応しており、カードゲーム演習の WinT でのコードに適用できる。また、臭いに関しては、長すぎるメソッド(関数)と重複コードの検出に対応している。

## 7. 不吉な臭いの検出

本論における、不吉な臭いの検出方法を定義する。各値が閾値以上の場合、臭いがあると判定する。まず、長すぎるメソッドについては、ABCサイズの計測により検出する。この指標は、プログラムの規模を示すとされる。関数ごとに算出する。変数への代入、メソッド(関数)の呼び出し、条件文の各個数の二乗和平方根値である。

次に、重複コードに関しては、ソースファイル内のすべての関数の組み合わせに対して、文字列として、編集距離を算出し、長い方の文字列長で除算して標準化した値により検出する。

なお、これらの閾値は、現状、ABCサイズは15.0、標準化された編集距離は0.5を適用しているが、今後の検討が必要である。

## 8. 本手法の試験的な実践

本手法を試験的に実践した。対象は、大学情報系学科の学生4名である。全員を受講者として参加させた。自由記述のアンケートの結果、リファクタリングの目安として利用できる、コードの内容の評価は学習上参考になるという意見が得られた。一方で、本演習において学生は、成績に反映される得点や戦績などに注目しがちであり、コードの内部評価への関心は低い可能性があるとの懸念も示された。また、改善点として、臭いに対するリファクタリングの方針を

示してほしい、改善すべきコード位置を明示してほしい、コメントは教員側と受講者側の双方向で送受信できる方が良いという意見も得られた。この結果、本システムの演習への導入に関しては、機能的な改善点がある一方で、リファクタリングの実施に対しては有用である可能性が高い。また、内部評価への関心が低いという懸念に対しては、レポートや授業において、本システムでの指摘事項を参考にした改善への取組を、成績評価に含めることで、内部評価への関心を高め、プログラミング作法の習得を動機づけさせることが可能と考えられる。

## 9. おわりに

ゲーム戦略を題材とするプログラミング演習を実施している。カードゲームとボードゲームを題材とした演習を提案し、実際の授業で実践している。これらの演習では、支援システムを提供し、受講者から何度でも戦略の提出を受け付け、実行結果や順位を公開している。

本論では、コードの不吉な臭いに着目した自動的なフィードバック手法を提案した。臭いの検出を定量的に行い、結果を教授側や受講者に提示することで、演習における指導を円滑かつ的確に行うことや、受講者のリファクタリングに対する動機づけを図る。試験的な実践の結果、リファクタリングの参考として有効であり、ソフトウェア開発の学習に有用であることが示唆された。一方で、いくつかの改善点や懸念も示された。今後の課題として、多言語への対応、検出可能な臭いの拡充、閾値の設定に関する検討が必要である。これらにより、演習による学習効果の向上を図る。

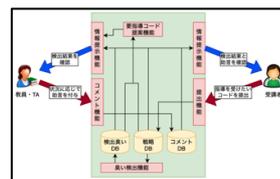


図1 システム構成図



図2 情報提示機能

## 参考文献

- 1) 花川直己, 玄馬史也, 富永浩之: ゲーム戦略を題材とする継続的なコード更新を促進する大会形式のプログラミング演習, 教育システム情報学会 第42回全国大会講演論文集, Vol.42, No.12-09, pp.317-318 (2017).
- 2) 玄馬史也, 富永浩之: ポーカー戦略を題材とする応用C演習の大会運営サーバにおける解答コードのメトリクス提示機能, 情報処理学会 78回全国大会, pp.889-890 (2016).
- 3) M. Fowler, K. Beck, 児玉公信, 友野晶夫, 平澤章, 梅澤真史: リファクタリング: 既存のコードを安全に改善する (2014).
- 4) 矢野尾裕: Adlint, <http://adlint.sourceforge.net/>.