様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　２ 年　 ７ 月 １２ 日現在

機関番号：１１６０１
研究種目：挑戦的研究（萌芽）
研究期間：2018～2019
課題番号：１８Ｋ１８６２２

研究課題名（和文）第２初学段階のプログラムロジック構成躓き推定のためのソースフュージョン機構

研究課題名（英文）Source Fusion Mechanism for Estimating the Stumbling in Program Logic Construction of the Second Novice Stage

研究代表者
中村　勝一（NAKAMURA, Shoichi）

福島大学・共生システム理工学類・准教授

研究者番号：６０３６４３９５

交付決定額（研究期間全体）：（直接経費）　　4,800,000円

研究成果の概要（和文）：プログラミング演習における学習者の躓きのうち，第２初学段階の「思い描く処理像をプログラムとして形作る段階での躓き（ロジック構成面の躓き）」は，重要性が指摘されながらも，これまで有効な把握支援方法が確立されて来なかった．本研究では，「学習者によるコンパイル作業の推移」と「ソースコード間の類似度推移」という性質の異なるソース系から，ロジック構成面の躓きを推定し，教授者に提示する新たなメカニズムを提案した．また，分析の過程で，分析戦略を動的に切替える仕組みについて検討した．これにより，コンパイルエラーに表出しないロジック構成面の躓きの把握について，新たな支援の可能性を示した．

研究成果の学術的意義や社会的意義
プログラミング演習において本来重要な要素でありながら，躓き状況把握の負担のために，これまで思うように実施できなかった「プログラム完成の成否に止まらない，各学習者の誤った認識や癖への対応など，踏み込んだ丁寧な指導」への教授リソース傾注を可能とする教育支援充実の意義を有する．

また，「コンパイル作業の推移」と「ソースコード間の距離推移」という性質の異なるソースを同時に活用する分析メカニズムに，教育工学，情報科学，認知科学に及ぶ分野融合的アプローチで挑戦する本研究は，従来とは違うレベルでの知見積み上げを可能とし，教育工学研究の新たな可能性を提示する意義を有する．

研究成果の概要（英文）：In programming exercises, it is important for instructors to successfully grasp the circumstances of each learner such as stumbling in order to provide effective guidance. In particular, the stumbling that students face when attempting to express the intended processing as a program (i.e. stumbling in construction of program logic) should be successfully grasped. However, there have been no effective methods for detecting the stumbling above since it does not result in compilation errors. In this research, we proposed the methods for estimating the stumbling in construction of program logic based on the analyses of compilation histories and transitions in the similarity between source codes. Consequently, the new possibility of support way for grasping the stumbling in construction of program logic was suggested.

研究分野：教育工学

キーワード：プログラムロジック構成　躓き推定　ソースフュージョン機構　ソースコード間距離　プログラミング学習支援

様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９（共通）

１．研究開始当初の背景

プログラミング演習では，一般に，多数の学習者に対して少人数の教授者で対応する必要がある．そのため，的確な指導を行うためには，学習者の躓き等の状況を上手く把握することが重要だが，容易ではない．

これに対して，既存研究の多くが，いわゆる「文法エラー」を扱ったものであった．一部，いわゆる「論理エラー」を対象とした取り組みは報告されているが，初学者の技能習得フェーズを鑑みるとき，「思い描く処理をプログラムとして形作る段階での躓き（ロジック構成面の躓き）」と，いわゆる論理エラーは，関連はすれども性質を異と認められる存在である．このロジック構成スキルは，言わば「第２の初学側面」にあたり，後のより高度な学習段階にも大きな影響を及ぼすものであり，その把握を現実的に支援する方法が必要と言える．

２．研究の目的

本研究では，プログラミング演習における「学習者によるコンパイル作業の推移」と「ソースコード間の距離推移」という性質の異なるソース系から，ロジック構成面の躓きを推定し，教授者に提示する新たなメカニズムの開発を目指す．これにより，コンパイルエラーに表出しないロジック構成面の躓きの把握について，新たな支援の可能性を探る．

３．研究の方法

まず，学習者によるコンパイル履歴の分析に基づいて，躓きの候補を抽出する手法を開発する．次に，抽出した候補について，各コンパイル時点におけるソースコードと正解コードの類似度を算出し，その推移の分析によりロジック構成面の躓きを抽出する手法を開発する．また，分析の過程で，分析戦略を動的に切替える仕組みについて検討する．その上で，これらの手法を導入したロジック構成躓き把握支援システムのプロトタイプを開発する．最後に，実験を通して，ロジック構成面の躓き推定手法の有効性検証と知見整理を試みる．

４．研究成果

４．１　ロジック構成面の躓き推定手法

（１）推定の手順

以下の手順に基づいて，ロジック構成面の躓きを推定する．

① 一次候補抽出：学習者のコンパイル履歴を分析し，エラーを伴わないコンパイルが一定回数以上続いているケースを抽出し，各コンパイル時点でのソースコードを取得する．

② ソースコード間の類似度算出：①で抽出した各コンパイル時点のソースコードについて，正解ソースコードに対する類似度を算出する．

③ 躓き推定：②の算出結果から「類似度の推移」を分析し，ロジック構成面の躓きを推定する．

（２）コンパイル作業の推移分析による一次候補抽出

コンパイル履歴を分析し，エラーの無いコンパイルが繰り返されている（しかし，完成には至っていない）ケースを抽出し，該当する学習者，コンパイル日時，ソースコードを一次候補として抽出する．例えば，図１では，２回目から４回目のコンパイル実施時が一次候補となり得る．

（３）ソースコード間の類似度推移分析

抽出した一次候補について，当該ソースコードの正解ソースコードに対する類似度を算出し，その推移を分析する．エラーの無いコンパイルが繰り返されており，なおかつ，正解ソースコードに対する類似度が上昇していない場合（図１の例では，0.6→0.5→0.5）に，ロジック構成面の躓きと判定する．

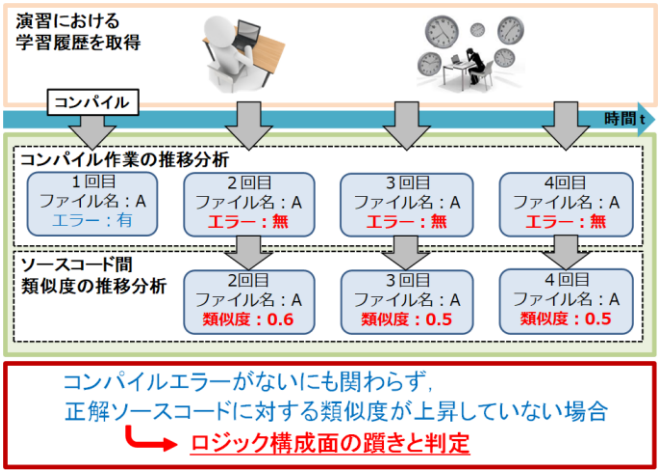

図１　ロジック構成躓きの推定

（4）ソースコード間の類似度算出

ソースコード間の類似度算出には，ソースを一種の文書と捉えた編集距離に基づく方法と，ソースコードから生成したツリーの類似性に基づく方法を用いる．ここでは主に，以下の手法を用いた．

① 編集距離に基づく類似度算出
- Order Levenshtein Distance (OLD)
- Jaro-Winkler Distance (Jaro)
- Dice Coefficient (Dice)

② ツリー表現による類似度算出
- Tree Edit Distance (TED)
- Tree Overlapping (TO)

4．2　支援システム

（1）支援システムの概要

本システムの概要を図 2 に示す．本システムは，初学者の多いプログラミング演習授業において，学習者の振る舞いの監視，履歴データやソースコードの収集を担う．本システムは，収集したデータの分析に基づいて，ロジック構成面の躓きを起こしている学習者を推定し，その結果を教授者に提示する．あわせて，教授者が学習者に対して指導し易い環境を提供する．このように，プログラミング演習において，ロジック構成面の躓きを起こしている学習者に対する指導を支援する．

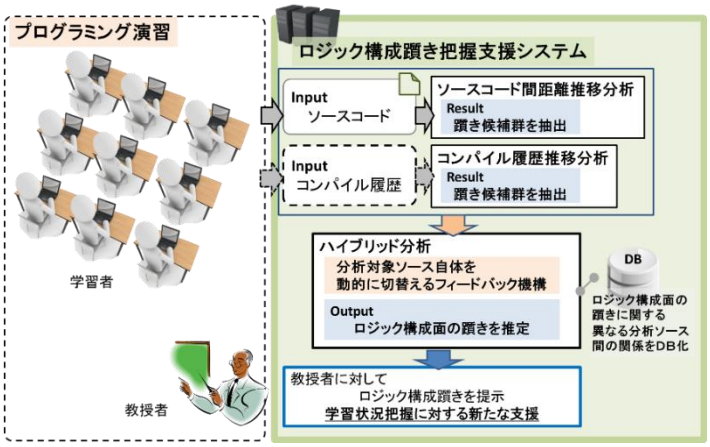

図 2 支援システムの概要

（2）プロトタイプの設計・開発

本支援システムの要件を，以下の通り整理した．

要件 1: 学習者の学習履歴を蓄積できること
要件 2: ソースコード間の類似度算出ができること
要件 3: ロジック構成面の躓きを推定できること
要件 4: 教授者に躓き状況を提示できること

本システムの構成を図 3 に示す．上述の要件を満たすために，本システムに，プログラミング演習における学習履歴の取得・蓄積，ロジック構成面の躓き推定などの機能群，および，各種データの管理やユーザ管理等に対応する基本機能を具備させる．

これらの機能を実現するために，コンパイル履歴取得，ソースコード間の類似度算出，各種分析に対応するモジュール群を備える．また，支援システムで扱うデータ類を管理するために，データベースを構築した．

設計に基づいて，プロトタイプを開発した．ソースコード間の類似度算出など主要部分は，Python を用いて実装した．また，コンパイル履歴，ソースコード，ソースコード間の類似度，それらの分析に関わる中間データなどを管理するデータベースは，MySQL を用いて構築した．

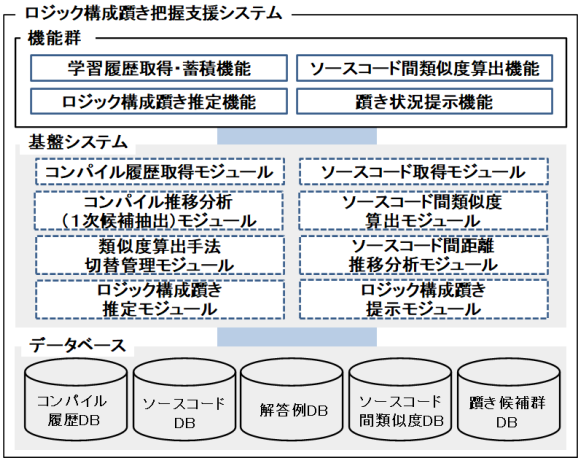

図 3 システム構成

４．３　実験と考察

（１）実験概要

ロジック構成面の躓き推定におけるソースコード間の類似度算出手法の性質検証と，課題抽出を目的として実験を行った．具体的には，まず，プログラミング演習における例題を想定した正解ソースコードと，正解ソースコードに余計な記述部分を追加したものを準備した．その上で，各ソースコードの正解ソースコードに対する類似度を算出した．ここでは，上述した類似度算出手法のうち，OLD，Jaro，Dice の 3 つを用いた．最後に，算出した類似度を精査し，手法毎の特徴等を精査した．

（２）結果と考察

表 1 ソースコードの類似度算出結果

コード	OLD	Jaro	Dice
1	0.981	0.938	1.000
2	0.962	0.934	1.000
3	0.942	0.930	1.000
4	0.923	0.926	1.000
5	0.904	0.922	1.000
6	0.885	0.962	0.989
7	0.865	0.922	0.977
8	0.846	0.919	0.966
9	0.827	0.927	0.943
10	0.808	0.921	0.930

正解ソースコードに対する類似度の算出結果を表 1 に示す．まず，今回の実験では，コード番号が小さいほど余計な記述部分が少なく，コード番号が大きくなるにつれて余計な記述部分が多くなっている．したがって，正解ソースコードに対する類似度は，ソース 1＞ソース 2＞…＞ソース 10 となることが望ましい．表 1 より，OLD を用いた場合，ソース 1 からソース 10 に向かうにつれて類似度が減少し，狙い通りの値が得られた．また，Jaro は，一部想定と異なる値が生じているものの，概ね狙い通りの値が得られた．一方，Dice は，他の 2 つの手法に比べ，ソースコードの違いが類似度として現れる度合が低い結果となった．

これらの結果より，同じソースコード間の比較でも，算出手法によって，類似度として「差異」が抽出される度合が異なることが確認できた．これは，本研究における算出手法の動的切替の趣旨に符合する結果と言える．また，類似度算出法について，各々特徴的な傾向を伺うことができた．

この他に，例えば，余計な記述の分量，または，必要な記述の不足規模は同様でも，当該部分の記述内容（命令文・制御文の種類）によって，類似度に表れる度合に違いがあるとの感触を得ている．これらは，今後の検討を進める上で貴重な知見と考えられる．

本実験は限定的なものであるが，提案手法の有効性検証を考える上で良好な感触を得ることができた．個々の命令・制御文に焦点をあてることによる精度向上と，提案手法による対応範囲の確保は難しい問題であるが，動的切替のための指標充実に向けて，実際のデータを用いた検証・知見集約を重ねていくことが重要と考える．

５．主な発表論文等

〔雑誌論文〕　計7件（うち査読付論文　3件／うち国際共著　0件／うちオープンアクセス　0件）

１．著者名 Junichi Tachibana, Ryo Onuma, Hiroki Nakayama, Hiroaki Kaminaga, Youzou Miyadera, Shoichi Nakamura	４．巻 -
２．論文標題 Combined Analysis of Compilation History and Transition in Similarity between Source Codes for Detecting Stumbling in Construction of Program Logic	５．発行年 2019年
３．雑誌名 Proc. IEEE Conference on e-Learning, e-Management & e-Services	６．最初と最後の頁 20-24
掲載論文のDOI（デジタルオブジェクト識別子） 10.1109/IC3e47558.2019.8971784	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

１．著者名 Shota Kawaguchi, Tadashi Nishikawa, Yoshiki Sato, Ryo Onuma, Hiroki Nakayama, Shoichi Nakamura, Youzou Miyadera	４．巻 -
２．論文標題 Development of a Training Data Creation Support Environment for Estimating Programming Learning Situations	５．発行年 2019年
３．雑誌名 Proc. 18th International Conference on Information Technology Based Higher Education and Training	６．最初と最後の頁 -
掲載論文のDOI（デジタルオブジェクト識別子） 10.1109/ITHET46829.2019.8937339	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

〔学会発表〕　計4件（うち招待講演　0件／うち国際学会　0件）

１．発表者名 立花隼一，中山祐貴，大沼亮，神長裕明，宮寺庸造，中村勝一
２．発表標題 プログラムロジック構成躓き推定のためのコンパイル履歴とソースコード間類似度のハイブリッド分析
３．学会等名 第44回教育システム情報学会全国大会
４．発表年 2019年

１．発表者名 川口翔大，西川直志，佐藤克己，大沼亮，中山祐貴，中村勝一，宮寺庸造
２．発表標題 プログラミング学習状況推定のための教師データ作成支援システムの開発
３．学会等名 日本情報科教育学会 第12回全国大会
４．発表年 2019年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

―

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考
研究分担者	宮寺　庸造 (MIYADERA Youzou) (10190802)	東京学芸大学・教育学部・教授 (12604)	
研究分担者	中山　祐貴 (NAKAYAMA Hiroki) (80761569)	早稲田大学・グローバルエデュケーションセンター・講師（任期付） (32689)	