

多人数が参加するソフトウェア開発演習を対象とした TA-Bot を用いた進捗監視手法の提案

小野 光一^{1,a)} 清原 良三² 寺島 美昭¹

概要: 本研究では多人数が参加するソフトウェア開発演習授業において学生の進捗監視及び技術的なアドバイスの提示を行う TA (Teaching Assistant) を Bot (roBot) を用いた進捗監視手法を提案する。学生の進捗監視を実現する為に、定期的に提出される進捗報告を解析して作業遅延の深刻さを示す深刻度を算出し、その値と深刻度の変化率、開発に関するキーワードを元に学生に提示する技術的なアドバイスの内容とタイミングを決定する。アドバイスの提示とタイミング判定の値を決定する際、学生を用いた実験では、各々でプログラミングスキルや開発形態は異なることから一律に決めることができない。そこでシミュレーションを用いることで学生の開発状況を再現し、アドバイス提示とタイミング判定のための具体的な値を決定する。開発を再現するための要素として、作業遅延のタイミング、作業遅延の改善、作業遅延の発生箇所の 4 パターンを定義して開発ケースを再現しシミュレーションを行った。この結果からアドバイス提示とタイミング判定の値を決定した。

キーワード: 非同期型開発, TA, 進捗監視

A Study of Progress Management Method using TA-Bot for Software Development

Abstract: In this study, we propose a progress management method that uses Teaching Assistant - robot (TA-Bot) to monitor students' progress and provide technical advice for software development classes. To make the TA-Bot monitor the students' progress, TA-Bot calculates the severity of developmental delay to analyze their progress report submitted regularly and determines technical advice and the timing for providing based on the severity, the change rate for the severity, and keywords for development. For determining the value of providing technical advice, we can't determine the value because their programming skills and development process is different. Therefore, we reproduce the student's development situation using the simulation. For this purpose, we defined four patterns for reproducing the developmental situations the timing of experiencing developmental delay, improving the developmental delay, and the phase of developmental delay reproduced developmental situations and simulated the student's developmental situation. From the result of the simulation, we determined the value of providing technical advice and timing.

Keywords: Asynchronous development, Teaching Assistant, Progress Management

1. はじめに

本稿は非同期型で進行するソフトウェア開発演習授業を対象に、学生の進捗を監視して必要に応じて技術的なアド

バイスをを行う TA (Teaching Assistant) - Bot (roBot) を用いた進捗監視手法について述べる。対象の授業はオンラインで進行するため、学生同士及び学生と教員間での情報共有は SNS (Social Network System) を用いて行い、学生は定期的に作業の状況を記載した進捗報告を提出する。TA は提出された進捗報告をもとに技術的なアドバイスが必要であると判断した学生に対してアドバイスを提示する。この TA の役割を Bot に行わせるために、アドバイスを提示するタイミングと実際に提示する技術的なアドバイスの内

¹ 創価大学
Soka University, Tokyo 192-8577, Japan

² 神奈川工科大学
Kanagawa Institute of Technology, Kanagawa 243-0292,
Japan

a) e21m5311@soka-u.jp

容を決定する必要がある。そのために、本研究では学生ごとに作業遅延を定量化し、その値と値の変化率、そして開発に関するキーワードの3つをもとに提示する技術的なアドバイスの内容を決定した。また、本手法を評価するためには学生を集めて評価実験を行う必要があるが、実験規模が大きいことや学生間のプログラミングスキルの差により意図的な状況を作り出す事が難しく、再現性がない。そこで学生の開発ケースを再現してそれぞれの状況に応じて技術的なアドバイスを提示していくシミュレーションを用いて評価を行う。再現する開発ケースは多々考えられるが、本稿では代表的なケースでシミュレーションを行い、本手法の評価を行った。本研究では、本稿の構成は以下の通りである。第2章で研究目的について述べた上で研究背景について説明する。第3章では対象とする授業の詳細、第4章では研究内容、第5章でシミュレーション実験の内容について述べた後、第6章でまとめとする。

2. 研究目的と研究背景

研究背景を説明した上で、本研究での目的を説明していく。

2.1 研究背景

昨今、新型コロナウイルス感染症拡大によりオンラインを用いた活動が急速に普及し定着している。オンライン授業は学生が遠隔地にいても参加が可能となり、授業の実施形態が対面の時よりも参加が容易になっている。一方で学生間でのコミュニケーションは対面の時と比較してお互いの反応が見えずらく難しくなっている。

TA[1]とは学部生の授業において教員の作業補助や学生へのアドバイスを行う役割を担う大学院生のことを指す。対面の授業では学生一人ひとりの学習状況を確認が容易で学生からの反応も確認できることからより効果的にアドバイスを行う事ができる。しかし、オンライン授業においては学生の顔を見てアドバイスをするとはいえないため、学生の反応を見る事が難しく効果的なアドバイスを行う事が難しい。また、授業の参加人数が増えてくると、教員はその分の学生の状況を把握して授業を進行していく必要があるが、参加人数が100人となってくると学生の学習状況の把握は難しくなってくる。

文部科学省では、成長分野を支える情報技術人材の育成拠点の形成(enPiT)において高度情報技術人材育成に力を入れており、全国の大学と連携して課題解決型学習(PBL;Project Based Learning)を用いて学習を進めていく。2021年4月からは各分野・大学で自主展開が行われている。この取り組みは高校や中学校にも範囲を広げて行われる事が予想される。[2][3][4]

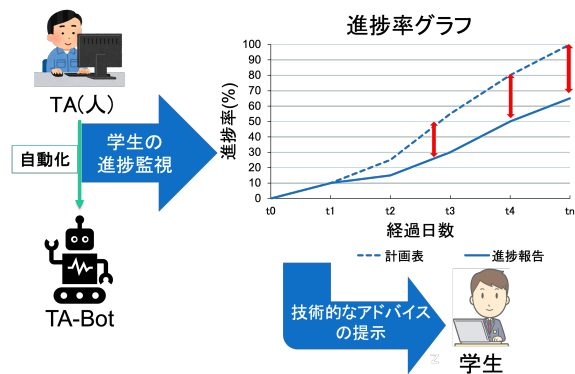


図1 TA-Botによる進捗監視

Fig. 1 Monitoring student's progress by TA-Bot

2.2 研究目的

本研究の目的はTAの作業効率化である。TAが作業監視の対象とする学生の人数が増えてくると、TAの負担は大きくなり、学生一人ひとりの進捗管理が難しくなってしまう。そこでTA-Botを用いてTAの役割である、学生の進捗監視と学生へのアドバイス提示を自動化することによってTAの作業効率化をして負担を軽減する。

研究背景における問題点としてTAの負担増大が考えられる。TAは多人数を同時に監視する必要がある、また、非同期型で開発が進められていることからどの学生がどの作業をしているどのような問題に直面しているのか把握が難しい。問題が発生した際には早急の対応が必要となるが、TAは常に学生を監視することは難しいため、技術的なアドバイスの提示が遅れることになり、問題がより深刻化してしまう。

そこで本研究では、学生の進捗監視及び技術的なアドバイスの提示をするTAをBotを用いてTAの役割である学生の進捗監視と技術的なアドバイスの提示を自動化することで作業の効率化を図る。提示するアドバイスに関しては対象としている授業が1回のみの実施でなく、毎年行われるものとしているので提示する技術的なアドバイスはパターン化されるものがある。そのパターン化されたものをTA-Botが提示をして、パターン化されていないものや複雑な問題に対してはTA(人)が対応する。それぞれの役割を明確に分担することでTAの作業の効率化を図る。(図1)

3. 対象の授業

本提案手法を適用する授業の詳細について説明する。授業の概要を図2に示す。学生の参加人数は100人程度の多人数で開発は個人で行い、学生間での情報共有や教員との連絡はSNSを用いて行われる。開発を行う前に各々で開発までの計画表を作成してTAに提出する。計画表にはどの時点で全体の何割の作業が完了しているのか記載する。また、TAがより詳細に状況を把握できるように開発段階

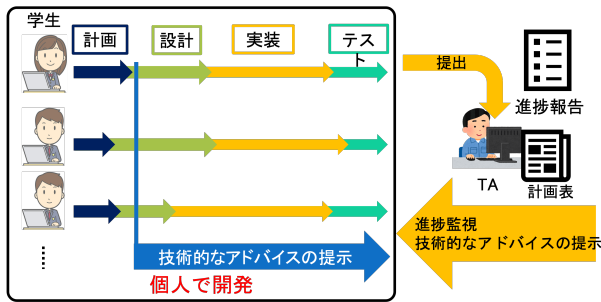


図 2 授業概要

Fig. 2 Software Development Class

の設計・実装・テストの3つのうちのどの段階であるかも記載する。開発形態は個人の都合の良いタイミングで行う非同期型で進行するため、学生は定期的に開発の進行状況を報告する進捗報告を提出して状況を報告する。進捗報告には進捗率の他に作業中の作業段階、状況の詳細を記したコメントをする。TAはこの進捗報告と計画表を比較して技術的なアドバイスをする必要がある学生に対してコメントを元にアドバイスを提示し、開発を円滑に進められるようにする。

4. 提案手法

TA-Bot の構成要素について説明をした後、提案手法全体の構成について説明し、学生の進捗監視及び技術的なアドバイス提示の際に用いる内容について説明していく。

4.1 TA-Bot 構成要素

本手法での TA-Bot を構成する要素として以下の2点を挙げる。

要素1 学生の進捗監視

要素2 適切なアドバイスの選択

TA は作業遅延が発生している学生に対して技術的なアドバイスを行う。TA は進捗報告内の進捗率やコメントから総合して提示する技術的なアドバイスの内容を決定する事ができるが、TA-Bot は学生の作業遅延が発生していることが計画表と進捗報告の進捗率の差分から確認できてもその作業遅延がどの程度深刻であるかまでは把握することはできないため、選択した技術的なアドバイスをいつ提示すべきか判断ができない。

また、TA-Bot は技術的なアドバイスを取得する際にはあらかじめ用意したデータベースから選択する。しかし、学生ごとで作業状況やプログラミングスキルの違いから提示する技術的なアドバイスの内容は異なる。そのため、どの技術的なアドバイスが学生にとって適切であるかを考慮して選択する事が難しい。

4.2 構成

本手法の構成を図3に示す。SNSはSlack[5]を用いて学

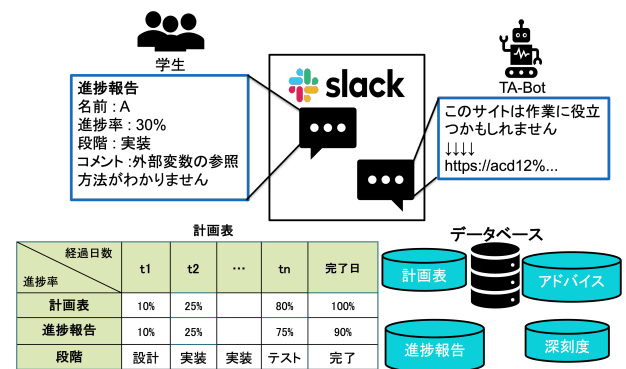


図 3 システム構成図

Fig. 3 System Structure

生間及び学生と教員・TAでの情報共有を行う。学生は定期的にSlack上で進捗報告の提出をする。進捗報告の内容は、全体の作業の何割が完了していることを示す進捗率と取り組んでいる作業段階(設計・実装・テストのいずれか)、作業状況を記したコメントを記載する。TA-Botはデータベースに学生一人ひとりの計画表を持っており、進捗報告の進捗率と計画表の進捗率から作業遅延の深刻さを表す深刻度を一人ひとり算出する。この深刻度と前回の深刻度の変化率、進捗報告内のコメントから抽出した開発に関するキーワードをもとに技術的なアドバイスをデータベース内にある情報から取り出して学生に提示する。

4.3 深刻度

深刻度は作業遅延の深刻さを定量化したものである。深刻度は学生ごとに定期的に行い、この値から技術的なアドバイスを提示するタイミングとアドバイスの種類を決定する。作業遅延のみを把握したい際は、計画表と進捗報告の進捗率の差分を取ることで把握が可能であるが、差分では作業遅延の深刻さを表す事ができておらず、技術的なアドバイスの提示のタイミングやアドバイスの種類を決定できない。図4.3では、左の図は計画表と進捗報告の進捗率の差分が期限日まで一定であった際の例を表している。新調率の差分が期限日まで同じであっても作業遅延の深刻に関しては期限日に近いほどより深刻さを増しているはずである。そこで作業遅延を定量化することにより、期限日に近いほど数値が上がりやすくして、より効果のある技術的なアドバイスを選択できるようにする。深刻度の計算式を(1)に示す。n回目での進捗報告の際の T_{np} を計画表の進捗率、 T_{na} を進捗報告の進捗率、 E を経過日数、 D を全体日数としている。まず進捗率の差分を求めてその差分がその日までの全体の何割であるのかを求める。そこに経過日数と全体日数を重み値としてかけることで深刻度を算出する。

$$S_n = \frac{T_{np} - T_{na}}{T_{np}} \times \frac{E}{D} \quad (1)$$

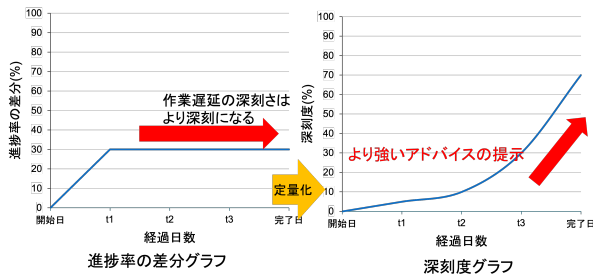


図 4 差分と深刻度の違い

Fig. 4 Quantifying the severity of the developmental delay

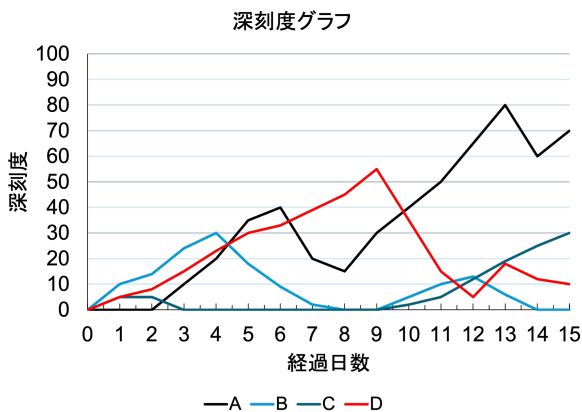


図 5 深刻度の利用

Fig. 5 Using Severity

図 4.3 に深刻度の利用例を示す。深刻度は学生ごとに計算することによってそれぞれの作業遅延の深刻さをグラフにすることができる。学生 A は前半に作業遅延が発生して一時的には解消傾向に向かったものの、その後は作業遅延が悪化してしまい、最終日に作業が完了せず計画が崩壊してしまっている。学生 B は前半に作業遅延が発生したがその後解消され、再び作業遅延が発生したが最終日には作業が完了しているので深刻度の値は 0 になっている。学生 C は序盤に作業遅延が発生したものがすぐに解消されてその後は作業遅延が起きることなく計画通りに作業が進められて最終日に作業が完了している。学生 D は前半から中盤にかけて作業遅延が少しずつ拡大しており、その後は解消傾向に向かったが再び作業遅延が悪化してしまい、最終日には深刻度の値が 0 ではないことから計画は崩壊してしまっている。

深刻度を下げて作業遅延を解消傾向に向かわせるために深刻度に閾値を設定することで技術的なアドバイスを提示するタイミングを決定することができる。閾値の設定に関しては 4.6 にて詳しく説明する。

4.4 変化率

変化率は深刻度の変化率のことを指し、 n 回目の深刻度の値が $n-1$ 回目と比べて何%の変化があったのかを表したものである。計算式を (2) に示す。

$$Z = \frac{S_n - S(n-1)}{S(n-1)} \times 100 \quad (2)$$

作業遅延が発生した際の変化率に関しては、 n 回目の深刻度の値が $n-1$ 回目からなだらかに上昇するものや急激に上昇するもの大きく 2 つが想定出来る。変化率の上昇の仕方に違いがあるということは作業状況に違いがあるということであり、提示すべき技術的なアドバイスの内容も違ってくる。深刻度の値とその変化率を見ることで提示する技術的なアドバイスの選択範囲を絞る。

4.5 開発に関するキーワード

キーワードは進捗報告のコメントに含まれている開発に関するキーワードのことを指す。学生ごとに直面する問題は異なるため、提示するアドバイスの種類は決まっても具体的にどのアドバイスを提示すべきかは深刻度及びその変化率では判断できない。そこでキーワードをコメントから抽出することで具体的にどのアドバイスを提示すべきかを決定する。これを決定するためにはコメント内に学生が開発においてどのような問題に直面しているのか把握できるように開発に関する言葉を入れてもらうようにする。

4.6 アドバイス選択

学生に提示する技術的なアドバイスを決定するために、上記で述べた深刻度と変化率、キーワードを使用する。アドバイス表を 1 に示す。技術的なアドバイスの選択手順について説明する。技術的なアドバイス選択アルゴリズムを図 4.6 に示す。まず、事前に深刻度に閾値を設定したアドバイス判定閾値からこの値の範囲内にあるのかを見る。次に深刻度の変化率の閾値がどの範囲を見て、コメントから抽出したキーワードがどれに当てはまるのかを見て提示する技術的なアドバイスを決定する。閾値に関しては学生一人ひとりに設定して提示するアドバイスのタイミングを調整する事が可能である。

次にそれぞれでの閾値に関して説明する。アドバイスをやるタイミングを決定するためにアドバイス判定閾値を設けて学生ごとに作業遅延の深刻さをレベルで分けてアドバイスの強さを決定した。これをアドバイスレベルと呼ぶ。今回は Level 1, Level 2, Level 3 の 3 段階に分けてアドバイスをやる内容の決定の材料にした。また、変化率にも閾値を設定してレベルで分けることで具体的なアドバイスの内容の絞り込みを行なっている。このレベルを変化レベルと呼ぶ。今回は Level 1 と Level 2 の 2 段階で分けてアドバイスの絞り込みを行った。これらの閾値を設定するためには実験を繰り返して行なって、調整していく事が必要となる。

提示するアドバイス情報に関しては、事前にデータベースに保存しておく。本研究の対象の授業は 1 回の実施ではなく毎年行われるものとしているので、学生に提示

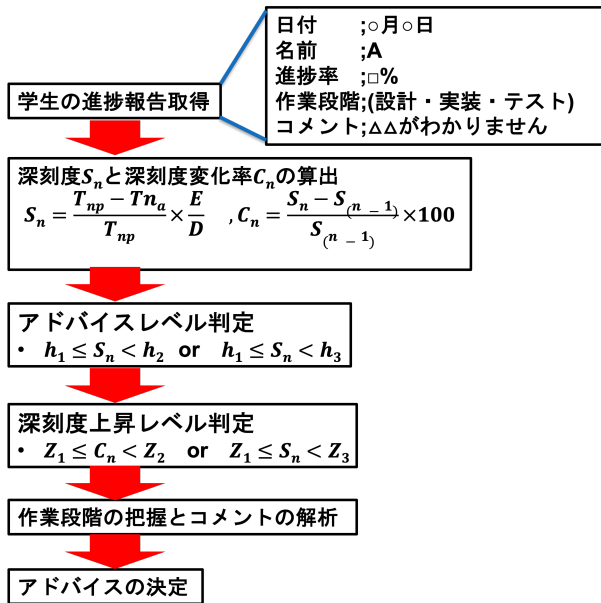


図 6 技術的アドバイス選択アルゴリズム
Fig. 6 Algorithm for Selecting Technical Advice

する技術的なアドバイスこれまでの提示したことのあるアドバイスを中心にデータベースに保存していく。また、ソフトウェアの開発技術は年々発展しており、新たな技術も交えて学生に技術的なアドバイスをして開発を進めてほしい為、データベースには新たなアドバイス情報を追加していけるようにしている。

アドバイスデータベースは、設計・実装・テストそれぞれで用意する。設計では、部分的な処理手順の確認とプログラム全体の処理手順を提示する。実装では、開発に関するヒントと開発に関するキーワードと使用例が記載されている Web 情報の提示をする。テストではそれぞれのエラーに対するアドバイスの提示を行う。各技術的アドバイスにキーワードを設定して TA-Bot が各学生にあったものを選択できるようにする。

表 1 アドバイス表

アドバイスレベル	変化レベル	キーワード	アドバイス
Level1	Z1	K1	A1
		K2	A2
	Z2	K3	A3
		K4	A4
		K5	A5
Level2	Z3	K6	A6
		K7	A7
	Z4	K8	A8

5. 実験

実験を行う上での問題点と今回用いたシミュレーションでの学生の開発状況再現について述べた後、実験の内容及

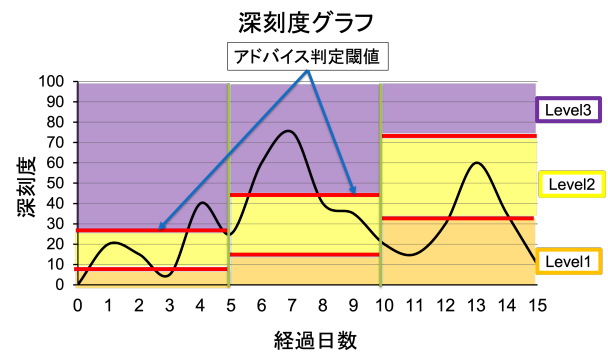


図 7 閾値の設定

Fig. 7 Setting Thresholds for Selecting Technical Advice

びその結果と考察について説明する。

5.1 実施における問題点

上記で述べた深刻度閾値と変化率閾値を決定するためには、学生にプログラミングの問題を解いてもらい、そこで発生した問題に対して TA-Bot が深刻度、変化率、キーワードから技術的なアドバイスを選択して提示し、そのアドバイスが提示した学生に対して作業遅延を解消する効果があったかどうかを確認していく実験を繰り返し行う必要がある。しかし、想定指定授業での実施となると、実施期間が長くなってしまふことや、規模が大きくなってしまふため実施が難しい。また、繰り返し実験を行う事ができず、閾値を決められるだけの十分なデータを集める事ができない。さらに、学生によってプログラミングスキルには差があるため、想定した状況下での実験の実施が難しくなり、再現性がない。

5.2 シミュレーション

そこで本研究ではシミュレーションを用いて閾値のおおよその設定を行う。学生相手ではないため、実際のデータとして得ることはできないがシミュレーションでは多くの開発ケースを再現できるため、同じ状況下で繰り返し実験を行う事が可能である。それぞれの閾値で決定した技術的なアドバイス内容が適切なものであるかどうかと提示する他ミシングは適切かどうか判断して、閾値の調整を行っていく。

5.3 開発の再現要素

次に学生の開発状況を再現するためのパラメータについて説明していく学生の開発状況を再現するための要素として、作業遅延のタイミング（14パターン）、作業遅延の改善のタイミング（3パターン）、作業遅延の発生箇所（7パターン）の4つから開発ケースを定義して合計で294ケースの開発を再現してシミュレーションを行う。

5.3.1 作業遅延のタイミング

作業遅延が発生するタイミングは学生ごとで異なる。今

回、対象としている授業は全15回構成の授業であるため、作業遅延が発生するタイミングは15回目の授業を除いて14回ある。15回目を除いているのは、これ以降の授業はないため、仮に技術的なアドバイスを行っても意味がないからである。

5.3.2 作業遅延の改善のタイミング

作業遅延の改善では、TA-Botが選択した技術的なアドバイスが必ずしも学生に効果があるとは限らない。そこで何回目のアドバイスの時点で作業遅延の改善の傾向が見られるかを設定する。今回は3回までとして3パターンの再現をする。

5.3.3 作業遅延の発生箇所

作業遅延はどの作業段階での作業遅延であるか考慮する必要がある。そこでTA-Botのアドバイス範囲である、設計・実装・テストの3つの段階の内、どの段階で作業遅延が発生しているのかを設定する。今回は3つの段階の中で1つのみで作業遅延が発生するのか、2つの段階で作業遅延が発生するのか、3つ全ての段階で作業遅延が発生しているのか

5.4 内容

作業遅延が発生するポイントと作業遅延の深刻さを自分で設定して進捗報告のコメントも設定して実験を行った。その時の深刻度の値と変化率、開発に関するキーワードからアドバイスを決定して、深刻度の変化を見る。アドバイスしたタイミングとその内容が効果があったか確認をする。深刻度の値によって提示するアドバイスは変わってくるので、今回は深刻度の式の $(T_{np} - T_{na}) / T_{np}$ の値が0.8と0.5と0.2の3つを設定して作業遅延の深刻さに違いを持たせて実験を行った。今回、作業遅延が最初に発生する箇所を実装段階にして提示するアドバイスが作業遅延解消の傾向を示した時とそうでない時の2通り、作業遅延の解消傾向を示したときにはあらかじめ各アドバイスに設定した、深刻度解消率をもとに深刻度を減少させた。作業遅延が解消しなかった際は、その次の深刻度及び変化率を計算し、その時のコメントをもとに新たに技術的なアドバイスを提示する。今回の実験では代表的なケースのみで実験を行い、いずれも最終日には作業が完了するようにして深刻度及び変化率の閾値の調整を行った。

今回設定した各作業段階でのアドバイス判定閾値と変化率閾値を表に示す。変化率閾値に数字が記入されていないものは提示する技術的なアドバイスが一つしている箇所であるためである。

5.5 結果

シミュレーション結果を図8、図9、図10に示す。シミュレーション1では、6回目の時点で発生した作業遅延の際に進捗報告から「ポインタ変数」に関するコメントを

表2 アドバイス判定閾値と変化率閾値の設定(設計)

Table 2 Threshold Deciding Technical Advice and Threshold Change Rate of Severity for Design phase

アドバイスレベル	アドバイス判定閾値	変化率閾値
Level1	5	30
		100
Level2	10	30
		100
Level3	20	30
		100

表3 アドバイス判定閾値と変化率閾値の設定(実装)

Table 3 Threshold Deciding Technical Advice and Threshold Change Rate of Severity for Coding phase

アドバイスレベル	アドバイス判定閾値	変化率閾値
Level1	5	30
		100
Level2	15	40
		100
Level3	35	

表4 アドバイス判定閾値と変化率閾値の設定(テスト)

Table 4 Threshold Deciding Technical Advice and Threshold Change Rate of Severity for Testing phase

アドバイスレベル	アドバイス判定閾値	変化率閾値
Level1	35	30
		100
level2	100	

受けたことからポインタ変数の使い方に関する情報を提示した。その後、作業遅延は解消傾向に向かったが12回目の時点でテスト段階の時に再び作業遅延が発生した。この時のコメントは「構造体の変数に関するエラー」であったことからプログラム全体の確認する旨のアドバイスを提示した。その後の作業遅延は解消傾向に向かった。

シミュレーション2では、8回目の時点で作業遅延が発生し、進捗報告から「構造体の使い方」に関するコメントが来たことから、深刻度の値も合わせて構造体の使い方がきざれているWeb情報の提示をした。その後の作業遅延は一時的に解消傾向に向かったが、11回目の時点で再び作業遅延が発生した。その時のコメントは「ポインタ」に関するもので作業段階は実装段階であったことからポインタの使い方が記載されているWeb情報の提示を行った。その後は解消傾向に向かい、最終日には作業が完了した。

シミュレーション3では、7回目の時点で作業遅延が発生し、進捗報告から「外部変数」に関するコメントを受けたことと深刻度の値から外部変数の使用例が記されているWeb情報を提示した。その後の作業遅延は一度解消傾向を示したが、9回目に作業遅延が悪化した。その時のコメントが「構造体」に関するものであったことと深刻度の値

と変化率の値から構造体の使用例が記されている Web 情報の提示を行った。その後、作業遅延は解消傾向を示したが深刻度の値が0になる前に作業遅延が再び悪化した。その時の作業段階はテスト段階でコメントは「コンパイルエラーの原因がわからない」ことであったことから、技術的なアドバイスとしてプログラムの随所に print 文を入れて問題が発生する箇所を見つけていく旨の内容を提示した。その後は解消傾向に向かい、最終日には作業は完了した。

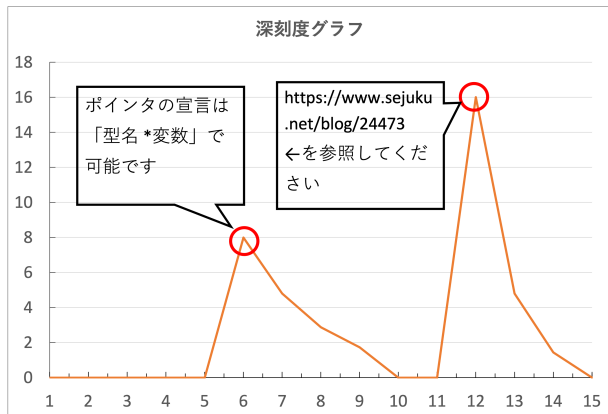


図 8 シミュレーション結果 1
Fig. 8 Result of Simulation1

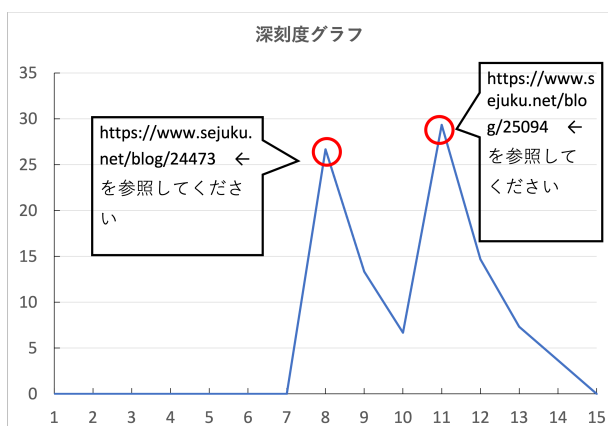


図 9 シミュレーション結果 2
Fig. 9 Result of Simulation2

5.6 考察

シミュレーション実験から深刻度の値及び変化率、開発に関するキーワードから技術的なアドバイスを取り出して学生の作業状況を意図的に作り出すことができた。提示したアドバイスに関しては、実際に学生に向けて提示した訳ではないため、どの程度の効果が見込めるかの確認はできていないが、提示したタイミングに関してはそれぞれの深刻度の値を考慮すると適切であったと考えられる。しかし、今回は設定した閾値では、シミュレーション 1 において深刻度の値がアドバイスレベル 1 で設定した閾値を初め

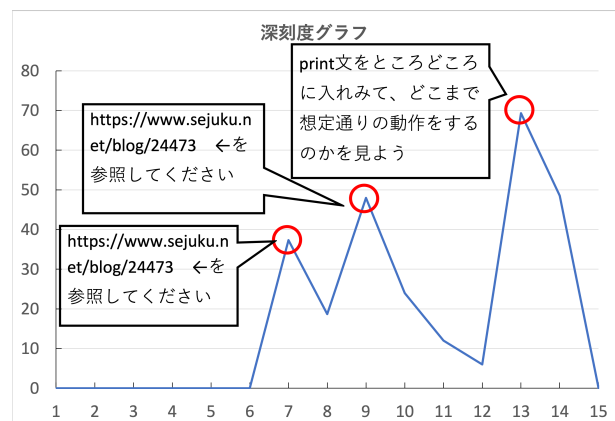


図 10 シミュレーション結果 3
Fig. 10 Result of Simulation3

から超えていた為、初めからアドバイスレベル 2 の技術的なアドバイスを提示していた。シミュレーション 1 のでの状況ではアドバイスレベル 1 の閾値を今回設定した閾値よりも高く設定しても後の作業には大きな影響はシミュレーション 2 とシミュレーション 3 と比較してみるとない為、次回の実験では閾値を高くして行う。また、変化率の閾値に関しては作業遅延解消傾向を示した後に再び深刻度が上昇することを考慮できていなかった部分があり、変化率の閾値の設定を大きく上回ったことがあった。そのため、選択されるアドバイスが各アドバイスレベルでの一番上の技術的なアドバイスとなっていた。シミュレーション 3 のような深刻度の値が大きく上昇する場合も考慮して次回の実験では変化率の閾をいずれのアドバイスレベルにおいても高く設定する。

シミュレーション実験の結果からそれぞれの状況に合わせて技術的なアドバイスを提示して効果があった際となかった際の深刻度の変動を見ることができた。また、提示した技術的なアドバイスが提示したタイミングと内容がそれぞれの状況に合わせてアドバイス判定閾値と変化率閾値の調節を行なって再度同じ状況下で実験を行うことができる。一度の調整では閾値の決定はできないため、今後もシミュレーション実験を行っていく必要がある。

実際の技術的なアドバイスの有効性の確認と調節した閾値の妥当性は実際に学生に提示してみないとわからない為、今後は 20～30 人を対象とした実験を行い、シミュレーションで調節した閾値の値をもとに技術的なアドバイスを行っていく必要がある。また、技術的なアドバイスに関しては自身の TA の経験をもとに作成しているため、データベースに保存されている情報は不足している。今後の TA 経験と別の TA のアドバイス経験をもとにアドバイスデータベースの拡張を行っていく必要がある。

6. まとめ

本稿では非同期型で進行するソフトウェア開発演習授業

における TA - Bot を用いた進捗監視手法について述べた。学生に技術的なアドバイスを提示するタイミングを図るために進捗率から作業遅延の深刻さを計算した深刻度を用いた。また、深刻度の値と深刻度の変化率の閾値・開発に関するキーワードを参照することで具体的な内容を決定して技術的なアドバイスを学生ごとに選択できるようにした。

また、深刻度の値と深刻度の変化率の閾値を調整するためにシミュレーションを用いた実験を行った。シミュレーションで学生の開発状況を再現するために、作業遅延のタイミング・作業遅延改善のタイミング・作業遅延の発生箇所の3つの要素から開発ケース294ケースを再現して代表的なケースで実験を行った。実験の結果から、提示した技術的なアドバイスの妥当性の確認と技術的なアドバイスをするために必要な深刻度の値の閾値と深刻度の変化率の閾値の調整をすることができた。学生の開発パターンは多岐にわたるため、開発を再現する要素をより増やすことでより細かい開発状況を作り出して閾値の設定をしていく必要がある。

本研究ではウォーターフォールモデルでのソフトウェア開発のための進捗監視手法について述べたが、企業でのソフトウェア開発を考慮すると、一般的にはスパイラルモデルやアジャイル開発が多数である。今後の展望としてそれぞれの開発形態に合わせて進捗監視の手段を変更していく必要がある。

参考文献

- [1] ティーチング・アシスタント (TA) について, 文部科学省 入手先 (https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo4/003/gijiroku/attach/1416216.htm) (2022.12.9)
- [2] 成長分野を支える情報技術人材の育成拠点の形成 (enPiT), 文部科学省 (オンライン) 入手先 (https://www.mext.go.jp/a_menu/koutou/kaikaku/enpit/index.html) (2022.12.9)
- [3] 成長分野を支える情報技術人材の育成拠点の形成 (enPiT) 入手先 (<https://www.enpit.jp/index-2.html>) (2022.12.9)
- [4] 文部科学省 (オンライン), コロナ対応の現状, 課題, 今後の方向性について, 今後の国立大学法人等施設の整備充実に関する調査研究協力者会議 (第5回) 令和2年9月24日 (木) 入手先 (https://www.mext.go.jp/content/20200924-mxt_keikaku-000010097_3.pdf)
- [5] slack 入手先 (<https://slack.com/intl/ja-jp/>) (2022.12.9)
- [6] Miyuki Yamamoto, Koichi Fukuoka, Ryoza Kiyohara, Yoshiaki Terashima. "Progress Management Method for Software Development Project-based Learning Using Automated Teaching Assistants" 2019 Twelfth International Conference on Mobile Computing and Ubiquitous Network(ICMU). IEEE, pp. 52-55 (2019)
- [7] T. Zimmermann, J. Coolen, J. Gross, P. -M. Pedrot and G. Gilbert, "Advantages of maintaining a multi-task project-specific bot: an experience report," in IEEE Software, doi: 10.1109/MS.2022.3179773.
- [8] I. Beschastnikh, M. F. Lungu and Y. Zhuang, "Accelerating Software Engineering Research Adoption with Analysis Bots," 2017 IEEE/ACM 39th International Confer-

ence on Software Engineering: New Ideas and Emerging Technologies Results Track (ICSE-NIER), 2017, pp. 35-38, doi: 10.1109/ICSE-NIER.2017.17.