Vol.2022-GN-115 No.44
Vol.2022-CDS-33 No.44
Vol.2022-DCC-30 No.44
2022/1/21

多人数が参加するソフトウェア開発演習における進捗監視手法の検討

小野　光一[1,a)]　寺島　美昭[1,b)]

概要：本研究では，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 SNS(Social Netowork System) を用いて行われる進捗報告から助言を行う TA(Teaching Assistant) の進捗監視手法を検討する．TA の助言範囲は計画からテストまでの各段階であり，また，参加人数は 100 人程度であることから誰がどの段階まで進行しているのか把握が困難である．そこで，各学生の進捗報告から作業遅延を計算した深刻度を使い，その数値からどのタイミングでどのような助言を行うか判断することにより多人数に対応する．本稿では，深刻度を使って助言内容について検証を行った．

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: We propose a progress management method using teaching assistant (TA) for asynchronous software development courses. TA supports to student's progress from design to testing, and monitors about 100 student's progresses. So, it is ridiculous to understand student's development progress. Now, we propose to handle a large number of people using severity by calculating progress rate from student's development progress. In the proposed method, the timing for advising a student determines by the degree of his/her delay. We experimented for advice using severity.

Keywords: Asynchronous development, Teaching Assist

1. はじめに

本稿は非同期型で進行する多人数を対象としたソフトウェア開発演習授業において，学生の進捗を監視し必要に応じて助言を行う TA（Teaching Assist）− Bot の進捗監視手法の検討について述べる．本研究では，学生間での情報共有を SNS（Social Network System）上で行い，TA はこのツール内で行われる進捗報告から各学生に対して助言を行う．ソフトウェア開発授業では，学生のプログラミングスキルやコミュニケーション力等が原因で作業遅延が生じてしまう．作業遅延を最小限に抑えるために逐一アドバイスを提示する教員の他に TA が存在する．TA は学生が各自で作成した計画表を元に計画通りに遂行できるように，問題発生時に早い段階で問題解決に向けて助言を行うことが主な役割となる．ここで考えられる問題として TA の負担が挙げられる．TA が助言を行う範囲は対象ソフトウェアの設計からテストまでの各段階に及んでおり，また助言を多人数に向けて行う必要があることから，誰がどの段階の作業を行なっているのか把握が難しく一人ひとりへの対応が遅れてしまう．これにより作業遅延が拡大する恐れがある．本手法では，TA-Bot を SNS 内に導入し進捗管理と助言内容の提示を自動化することにより TA の負担を軽減し，学生の作業を円滑に行うことが可能となる．本稿の構成は以下の通りである．第 2 章で研究背景について，第 3 章では対象とする授業についての詳細と問題点，第 4 章では研究内容，第 5 章で検証実験の内容について述べた後，第 6 章でまとめとする．

[1] 創価大学
Soka University, Tokyo 192–8577, Japan
[a)] e21m5311@soka-u.jp
[b)] tyoshi@soka.ac.jp

Vol.2022-GN-115 No.44
Vol.2022-CDS-33 No.44
Vol.2022-DCC-30 No.44
2022/1/21

2. 研究背景

高校や大学での授業の形態のひとつとして PBL（Project Based Learning :課題解決型学習）型授業がある．この授業は, 与えられた課題に対してグループ内で解決に向けた手順を話し合い，一つの答えを出す授業である．これは従来の受動的な学習ではなく，自ら問題解決のためのアイデア，実現性等を学生自身が推考することで能動的な学習ができ，学生の学習意欲やプレゼンテーション能力の向上を図る．文部科学省では情報技術人材の育成のためにプロジェクト型の授業を推進している [1]．その中に PBL 型授業が導入されていることから，ソフトウェアのプロジェクト開発が重要視されている [2]．また，enPiT（成長分野を支える情報技術人材の育成拠点の形成）では，ビックデータ・AI，セキュリティ，組み込みシステム，ビジネスシステムデザインの４分野における高度 IT 人材の育成を目指しており，その中で PBL 型授業を通して実践的な学習を行うことで，実践課題の解決に取り組んでいる．現在，enPiT は各分野・大学で展開されており，今後この授業形態が小学校から大学にますます普及していくと考えられる．また、昨今における新型コロナウイルス感染症拡大により対面ではなく，SNS や Web 会議システムといったオンラインを活用したものが主流となっている．オンラインを活用した学習はコロナが収束した後もオンライン教育として発展していくことが考えられる．[3]

3. 対象とする授業

3.1 詳細

授業の概要図を図 1 に示す．対象としている授業は１００人程度が参加するプログラミング授業である．学生は課題として与えられたソフトウェアを独立して開発する．開発は各々の都合の良いタイミングで行う非同期型開発で進められる．学生は対象のソフトウェアの開発において完成に向けた計画表を各自で作成し TA に提出してから開発を行う．作業段階は設計，実装，テストの 3 段階で，学生は何日目の時点でどこまでの作業を終えているのか計画表に記載する必要がある．ソフトウェアの完成日はどの学生も同じであるが，各学生が独立して開発が行われる為，作業の各段階にかける日数は学生ごとに違っている．学生同士，TA・教員間での情報共有は SNS を通じて行われる．TA は SNS 内の発言を常に監視し，学生から提出される進捗報告から，助言が必要と判断した学生に助言を行う．

3.2 問題点と目的

TA は１００程度の学生の進捗を同時に監視し必要に応じて助言を行う必要がある．また，学生の作業状況は一人ひとり異なっており，非同期型で独立して開発していることから，どのタイミングでどの学生がどの段階の作業を行なっているか把握が困難である。そのため，TA は学生の発言を常に監視する．しかし，TA は SNS 内での発言を常に監視することはできない為，助言を必要としている学生に対して助言の提示が遅れる結果となり，作業遅延が発生している場合には早急の対処ができず，作業遅延が拡大してしまう．

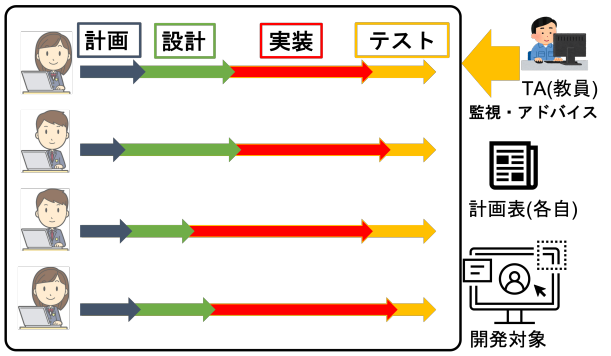

図 1 授業概要図
Fig. 1 A course description

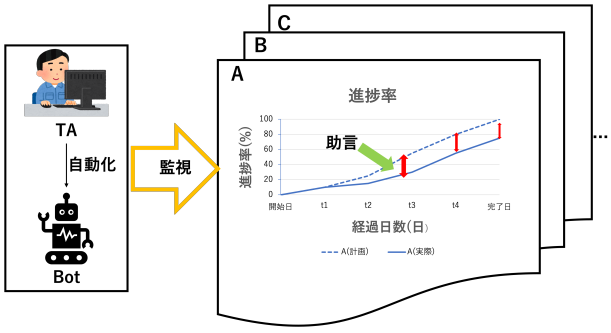

図 2 TA-Bot による進捗監視
Fig. 2 Monitoring student's progress by TA-Bot

そこで本研究では，TA–Bot を導入して進捗監視と助言内容提示を自動化することにより，１００人程度の学生の進捗の同時監視を自動化する．これにより進捗報告から助言が必要と判断した学生に対して即座に助言の提示が期待できる．（図 2）

3.3 研究課題

TA-Bot を構築するにあたり，研究課題として以下の 2 点を挙げる．

課題 1　学生の進捗計測
課題 2　状況に応じた助言内容の選択

開発において学生の進捗率は，学生の感覚で全体の何割進められたかを進捗率で報告する．進捗率の計測方法に関しては様々考えられるが，今回は学生の個人主観に任せたものを信用して使用する．しかし，Bot は学生の進捗を計測する指標を持っていない為，学生の進捗率を取得しただけではどの学生がどの程度遅延が発生しているのか把握す

Vol.2022-GN-115 No.44
Vol.2022-CDS-33 No.44
Vol.2022-DCC-30 No.44
2022/1/21

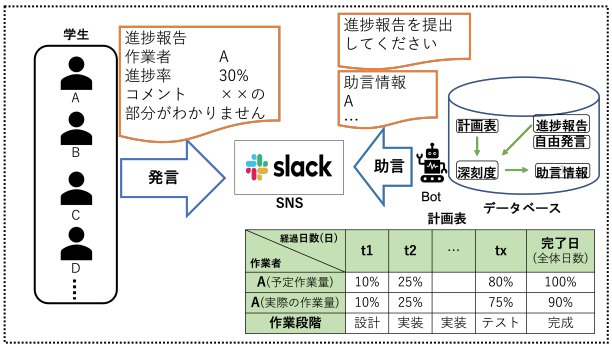

図 3 システム構成図
Fig. 3 System Structure

る必要がある．また，学生に提示する助言情報をどこから取得するのか，学生にとって適切な助言は何か，その助言をいつ提示するのか判断する必要がある．これらにより，学生に対して適切なアドバイスを提示することができず，作業遅延が拡大してしまう．人は学生に対してサボりや理解不足を支援 bot は作業遅延の支援、進捗監視

4. 提案内容

4.1 システム概要

システム構成図を図 3 に示す．SNS は Slack[4] を使用している．Bot はデータベースに学生から提出された計画表，助言内容，SNS 内での進捗報告を持っている．計画表には何日までに進捗率がどの程度進むのか記載されている．また，どの日までがどの作業段階であるかも記載されおり，Bot が進捗報告の進捗率からどの作業段階にあるのか把握できるようになっている．学生は日ごとにチャット内で進捗報告を行う．進捗報告の内容には，作業者・進捗率・コメント・ソースコードを記載する．Bot が進捗報告の発言であると認識できるように，文頭に進捗報告と記入してから上記の内容を 1 行ずつ入力するものを進捗報告フォーマットとする．（図 4）Bot はこの発言をデータベース内の「進捗報告」に保存する．この次に Bot は進捗報告内の進捗率（以降，実際の進捗率とする）と学生から提出された計画表内の進捗率（以降，予定進捗率とする）から後述する深刻度を算出する．深刻度の値から助言が必要と判断した場合，深刻度の値を元にデータベースから助言情報を選択し提示する．Slack 内には進捗報告以外の発言は自由発言とし，Bot は進捗報告の発言に対して助言を行う．学生の自由発言は今後の開発授業において過去事例として利用可能な情報が含まれていることが考えられる為，データベースには「自由発言」として保存をする．また，Bot は学生の進捗報告がないと深刻度の算出と助言の提示を行うことができない為，進捗報告を提出してもらうよう促す発言も行う．（図が追えていない状態）システム図の書き直し

表 1 計画表
Table 1 Development Schedule

作業者＼経過日数 (日)	t1	t2	…	tx	完了日
A(予定進捗率)	10%	25%		70%	100%
A(実際の進捗率)	10%	25%		75%	90%
作業段階	設計	実装	実装	テスト	完成

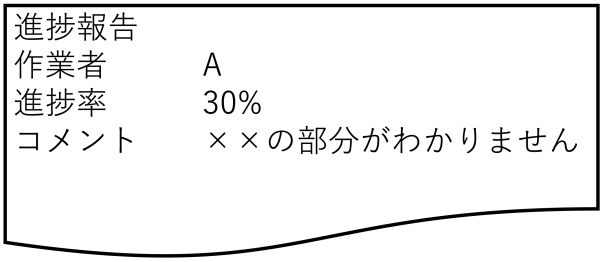

図 4 進捗報告
Fig. 4 Progress report format

4.2 深刻度

深刻度は学生の作業遅延を数値化したもので，学生ごとに計算を行う．日ごとに提出される進捗報告（4）の実際の進捗率と計画表の予定進捗率から深刻度の計算を行う．計算式を（1）に示す．Tnp は予定進捗率，Tna は実際の進捗率，E は経過日数，D は全体日数を表している．（2）この計算を日ごとに行い，数値の変動を見る．深刻度は上昇すればするほど作業遅延が拡大していることが確認できるようになっている．この深刻度に閾値を設けることで，作業遅延がどの程度のものなのか判断することができる．今回，深刻度による作業遅延の段階を「小・中・大」に整理した．小ならば計画に影響を及ぼすほどの作業遅延ではないと見なし，作業の催促が必要となる．中ならば計画に影響を及ぼしかねない作業遅延と見なし，影響が大きくなる前に対処が必要となる．大ならば計画に大きく影響を及ぼしている作業遅延と見なし，早急の対処が必要となる．

表 2 対応表

Tnp	予定進捗率
Tna	実際の進捗率
E	経過日数
D	全体日数

$$\frac{Tnp - Tna}{Tnp} \times \frac{E}{D} \tag{1}$$

深刻度を使うことで学生の開発パターンを整理することができる．今回，開発パターンを以下の三つに整理した．
パターン 1　序盤から徐々に遅延が拡大している
パターン 2　中盤から後半にかけて遅延が発生している
パターン 3　中盤までは遅延していたが後半には遅れを取り戻す

Vol.2022-GN-115 No.44
Vol.2022-CDS-33 No.44
Vol.2022-DCC-30 No.44
2022/1/21

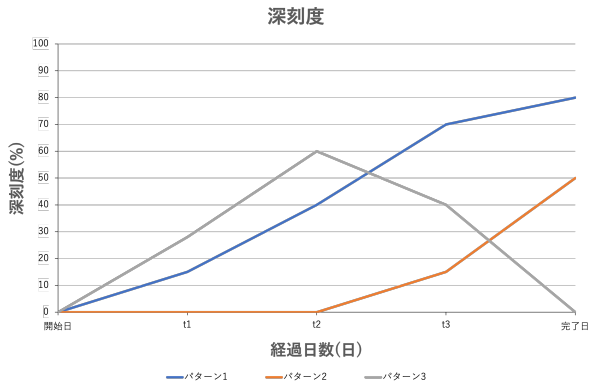

図 5 深刻度パターン
Fig. 5 Pattern of severity

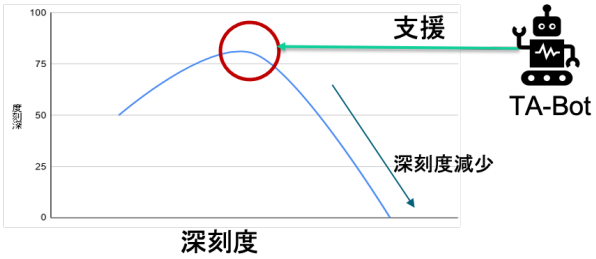

図 6 深刻度の利用
Fig. 6 Using severity

上記三つのパターンの深刻度のグラフを図 5 示す．パターン 1 は序盤から遅延が発生していることから，ソフトウェアの仕様に関して理解ができずに開発を進めている状態と見ることができる．その為，設計はできても実装の部分で作業遅延が発生してしまい，完了日までにソフトウェアを完成させることができない．TA は仕様に関する情報を学生に提示し，計画を意識した作業を進行するよう催促する必要がある．パターン 2 はソフトウェアの仕様は理解して設計はできているものの，プログラミングのスキルが不足していることから遅延が発生していると見ることができる．TA は開発を行う上での必要な知識に関する情報を提示する必要がある．パターン 3 はプログラミングの知識はあるが，ソフトウェアの仕様の理解及び設計に時間がかかってしまい，序盤に作業遅延が発生してしまったと見ることができる．TA は仕様に関する情報を提示する必要がある．このように深刻度を使用することで，学生一人ひとりの遅延状況を可視化できるようになり，それぞれの状況に合わせた助言を深刻度の閾値ごとにタイミングを設定することで深刻度を減少させて作業遅延を解消する．（図 6）

4.3 助言内容

助言内容の決定は以下の 2 種類に分けて行う．

- 静的アドバイス
- 動的アドバイス

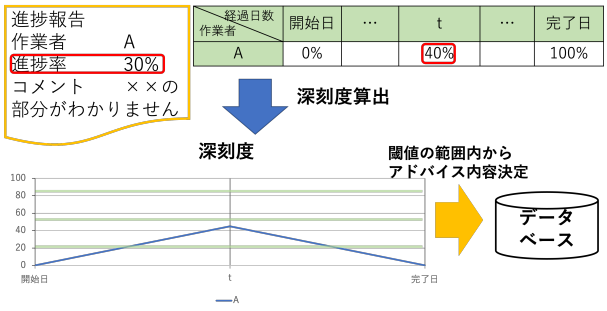

図 7 静的アドバイス
Fig. 7 Static Advice

この二つの助言方法は深刻度を見るかそれとも進捗報告のコメントを見ているのかの違いがある．これにより遅延状況からの助言の他に作業状況からの助言を可能とする．

4.3.1 静的アドバイス

静的アドバイスは前述した深刻度の値に閾値を設定し，その範囲内から助言内容を選択することを指す（図 7）．助言内容を表 3 に示す．助言の種類は「キーワードの提示」，「Web 情報の提示」，「プログラム内容についての説明」の 3 種類である．

深刻度の値が小であると判断した場合，大きな作業遅延ではないことから，開発を進める上でヒントとなるキーワードの提示を行う．キーワードは事前に TA が学生が取り組んでいる課題に取り組み，その中で学生がつまづいてしまう箇所を予測し，問題の箇所を解決するための技術をキーワードとして選択する．そのキーワードが複数あればその分キーワードとしてデータベースに保存しておくことで，他の問題にも対応できるようにする．

深刻度の値が中であると判断した場合，開発計画に影響を及ぼしかねない状態にあることから，開発に関わるヒントではなく，開発する上で必要な技術の使い方に関する Web 情報を提示する．Web 情報はキーワードの提示で使用したキーワードをを使用して，スクレイピング技術によって取得した Web 情報を提示する．

深刻度が大であると判断した場合，開発計画に大きな影響を及ぼしていることから開発に関わる技術に関する知識以前にプログラムの仕様を理解できていないことが考えられる．プログラムの仕様を理解して開発を進めてもらうためにプログラムの内容について説明を行う．この説明は事前に開発をした TA がどのような流れで処理が行われるのか記載されており，この説明文をデータベースに保存しておく．

4.3.2 動的アドバイス

動的アドバイスは進捗報告のコメントの内容から学生がどの状況にあるのか分析して助言内容を選択することを指す．（図 8）分析手順に関しては，コメントに形態素解析を行い，図 8 の部分の××をキーワードとして抽出し，このキーワードがデータベースに保存されているか探す．デー

Vol.2022-GN-115 No.44
Vol.2022-CDS-33 No.44
Vol.2022-DCC-30 No.44
2022/1/21

表 3　助言内容
Table 3　Advices

深刻度	助言内容
小	キーワード提示
中	Web 情報提示
大	プログラム内容についての説明

タベースに保存されていなかった場合，このキーワードを使って Web 検索を行い，それで得た情報を学生に提示する．ここで得た情報は過去事例としてデータベースに保存する．

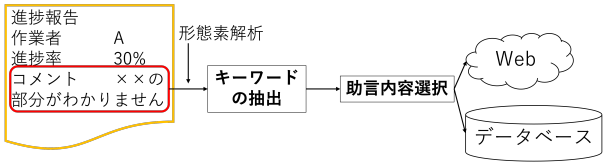

図 8　動的アドバイス
Fig. 8　Dynamic Advice

5.　実験

5.1　内容

検証実験では，深刻度による助言タイミングの判断基準と静的アドバイスの有効性の検証を行った．実験の規模は想定人数の１００人では大規模となり，実施が困難である為，今回は学生 2 人を対象に小規模で実施した．実験で用いた課題は Paiza ラーニング [5] の問題を利用した．問題数は 2 問用意し被験者に解かせた．作業期間は 6 日間で，作業終了後は毎日必ず進捗報告の提出を求めた．問題の一覧を表 4 に示す。左の作業番号は Paiza に記載されている問題番号を表しており、右の問題は問題名を示している。学生は上の問題から順に解き、完成したら次の問題に取り組ませた。進捗報告の内容は，作業者・作業番号・進捗率・コメント・ソースコードとした。作業番号を用意したのは、Bot が助言を提示する際に学生がどの作業に取り組んでいるのか把握するためである。進捗報告のコメント欄は，今後の動的アドバイスの検討の為に，過去事例として DB に保存する．進捗報告の発言以外に開発対象に関する発言は可能とした．毎日，正確な深刻度を算出できるように，進捗報告の提出を忘れさせない為に，Bot に毎日進捗報告の提出を求める発言をさせた．助言情報に関しては，TA が事前に問題を解き，どこの部分で遅延が発生しそうか予想したものを助言情報として DB に保存した．キーワードの提示と Web 情報に関しては，問題によって提示する情報が二つとなっているものがある．学生によっては余分な内容が含まれていることになる為，本来はコメントの発言から必要としている情報のみを提示すべきだが，発言解析による助言情報選択の実装はできていない為，助言情報が複数あるものは同時に複数提示した．

表 4　作業内容
Table 4　tasks

作業番号	問題
D012	エレベータ
D079	複数形への変換

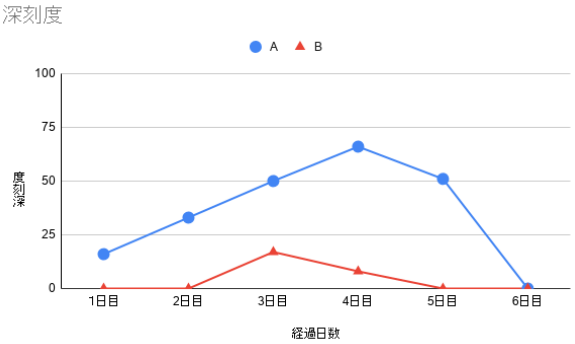

図 9　深刻度グラフ
Fig. 9　Severity graph

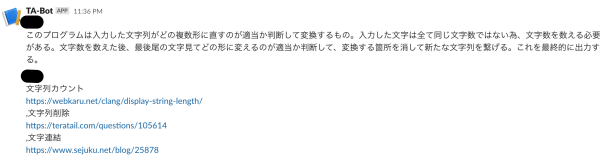

図 10　助言情報の提示
Fig. 10　Showing the advice

5.2　結果

本実験の深刻度のグラフを図 9 に示す．2 人の学生は最終的に完了日の 6 日目に与えられた課題を全て終えることができた．しかし，深刻度グラフを見ると数値が上昇している時があることから，途中で作業遅延が発生していることが確認できる．Bot は学生 A に対し，4 日目にプログラムの内容についての説明と参考となる Web 情報の提示を行なった（図 10）．助言提示後の 5 日目と 6 日目は深刻度の値が減少していることが確認できる．
学生 B は学生 A ほど作業遅延は発生していないが，3 日目に作業遅延が発生していることが確認できる．Bot は学生 B に対して助言は行なっていないが，それ以降の深刻度は減少していき最終的に遅延は発生しなかった．

5.3　考察

検証実験から，学生一人ひとりの進捗を深刻度を使って測り，適切なタイミングで提示ができた。助言情報提示後の深刻度の減少から作業遅延が解消の傾向にあることを確認した．

助言内容に関しては，どの情報がどの程度有効であったかは確認できなかった。提示後は，深刻度が減少していることから，助言情報の提示は作業遅延を意識して作業を進めるきっかけになった考えられる．今回用意した助言内容

は他の学生にも今回の学生と同様の状況下でも有効であるか判断するために，被験者人数を増やして検証および評価を行なっていく必要がある．

SNS 内での発言に関しては，進捗報告のみで学生同士の議論は行われなかった．原因としては，取り組んだ課題が 1 人で十分に取り組める難易度で，問題に直面しても 1 人で対応することができたために議論が行われなかったと考えられる．Paiza にはさまざまなレベルの問題が用意されているので，今後の実験では，今回より難易度の高い問題を選択して学生の作業遅延を意図的に発生させることにより、学生間での議論をさせてその発言内容から動的アドバイスの内容の検討を進めていく．

今回の検証実験の結果と，４．２項で述べた開発パターンを比較すると，学生 A の開発スタイルは第 4 章で述べた開発パターン１と類似する．しかし，学生 A はパターン１と違い，助言情報の提示が行われるまで作業遅延を意識していなかったことから深刻度の上昇に繋がった．開発パターン１と作業遅延を意識して作業していない学生には前半部分に関しては同様な深刻度の上昇が確認できた．このパターンの場合は，ソフトウェアの仕様に関して理解ができていない学生に対しては，仕様に関する情報を提示し，学生 A のような作業遅延を意識していない学生に対してはソフトウェアの仕様を理解できていない学生と同様の情報を提示をすることで自身の作業遅延を意識させて開発ペースを上昇させる．一方，学生 B の開発スタイルは開発パターン３に類似する．学生 B は基本的に作業は計画通りに進められており，計画に影響を及ぼすような作業遅延は発生しなかった．一度，発生した作業遅延は作業遅延を意識していなかったからであり，その後の作業は計画通りに進められていた為，大きな問題は発生していない．万が一，この作業遅延が拡大してしまった場合には，開発に関するキーワード及び Web 情報の提示を行うことで作業遅延の解消を狙う．

今回行った実験の人数が 2 人であった為，次回以降の検証では１５人～２０人程度を対象とした中規模での実験を実施し，Bot による助言情報の有効性及び助言提示による学生の作業遅延の解消の傾向を確認する．

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の助言の検討について述べた．提案手法では、深刻度を用いることで学生の遅延状況を可視化した．検証実験では，学生の進捗状況を深刻度を使って表し作業遅延を起こしている学生に対して助言を行うタイミングを深刻度を使って決定した。また，深刻度の数値に従って決定した助言内容の提示により学生の作業遅延解消の傾向に向かうことを確認することができた。これにより TA の役割である学生の発言監視及び助言作業をする負担を軽減できるようになると考えている．しかし，学生にとっては余分な情報が含まれていることが考えられる為，深刻度の値に従ってただ助言情報を提示するのではなく，学生のコメント内容を考慮することで提示する助言情報を選択することが必要である．また，提示した助言情報が学生にとってどの情報が役に立ったのか確認できていない．今後はデータベース内の助言情報の設計を引き続き行い，完成を目指す．また，チャット内での自由発言を分析して助言情報を選択する動的アドバイスでは，強化学習を用いることで，学生のさまざまな状況に合わせた助言情報提示を目指す。現在の検討では、個人開発を対象としている為、プロジェクト型の開発授業を対象に、個人に対してだけでなくグループ全体に対してにどのような助言が提示できるか検討していく必要がある。今後の検証実験では１５～２０人程度の中規模で実施し，様々な状況を模擬したシミュレーションによる実験及び評価を行なっていく．

参考文献

[1] 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⟩ (2021.12.19)
[2] 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⟩ (2021.12.19)
[3] 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⟩
[4] slack 入手先 ⟨https://slack.com/intl/ja-jp/⟩ (2021.12.19)
[5] Paiza ラーニング：入手先 ⟨https://paiza.jp/works⟩ (2021.12.16).
[6] 山本美幸,”ソフトウェア分散開発 Project Based Learning(PBL) のための進捗管理手法の評価”, マルチメディア, 分散協調とモバイルシンポジウム 2019 論文集,2019,1549-1554

正誤表

本稿の内容に誤りがありました。お詫びして訂正致します。

3ページ左5～6行目　(誤)；人は学生に対してサボりや理解不足を支援botは作業遅延の支援、進捗監視
(正)；不要な内容であるため、削除

3ページ左30行目　(誤)；(図が追えていない状態)システム図の書き直し
(正)；不要な内容であるため、削除

3ページ右6行目　(誤)；(2)
(正)；(表2)