

非同期型ソフトウェア開発演習授業において 進捗支援を行う TA Bot 機能の検討

小野 光一¹ 寺島 美昭¹

概要: 昨今における新型コロナウイルス感染症拡大によりオンラインでの活動が主流になってきている。そこで、本研究では非同期型分散 PBL(Project Based Learning) ソフトウェア開発授業を対象に、SNS を通じて情報共有を行い、発言内容から学生ごとに状況を分析して適宜支援を行う役割をする TA(Teaching Assistant) に着目し、作業を自動化する BoT(roBoT) の機能を提案する。対面での実施の場合、TA がその場で学生に対して支援を行うが、非同期型で分散開発である場合、どのタイミングで学生が開発を行なっているのか分からない為、監視するのが難しい。また、支援を求められている際にすぐに支援を行うことができず遅延が拡大してしまう恐れがある。そこで、提案手法では学生の進捗を BoT が自動で解析するために学生の遅延を数値化した深刻度を定義し、その数値から支援内容や支援を行うタイミングを判断する。実験では、開発の進捗が学生のスキルに左右することから再現が難しい。そのため、対象人数を 5 人とし規模を小さくした上で実験及び評価を行なった。

1. はじめに

本研究では各々の都合の良いタイミングで開発を行う非同期型分散 PBL (Project Based Learning) ソフトウェア開発授業を対象に、学生の進捗を監視し逐一支援を行う TA(Teaching Assistant) の役割に注目して研究を行っている。本研究では、学生間の情報共有や進捗報告などの発言を SNS (Social Network System) 上でやり取りを行い、状況に応じて各学生に対して適切な支援を行う TA の役割を BoT (roBoT) を用いて自動化する機能を提案する。対面でのソフトウェア開発では、学生のプログラミングスキル、コミュニケーション力等が原因で遅延が発生してしまう。このような状況を解決するために状況を監視し支援を行う TA が存在する。TA は発生した問題に対し、早い段階で解決に導き作業を円滑に進めることが役割である。そのため、TA は複数人の学生を同時に監視しなければならず、負担が大きくなってしまふ。また、想定している状況が各々の都合の良いタイミングで開発を行う非同期型開発であることから、TA は学生がいつ作業をしているのか把握することが難しい。そこに学生の遅延が発生した場合、すぐに支援が行われず遅延が拡大してしまう恐れがある。提案手法は、この TA の役割に着目し学生の進捗を監視し、状況分析から支援が必要であると判断した場合に支援を自動的に行う。

この手法を取り入れることで TA の負担を軽減し作業の効率化を狙うことができ、学生が円滑な開発を行うことが可能である。本稿の構成は次の通りである。第 2 章で研究背景について述べ、第 3 章では想定する状況、問題点及び研究目的について説明する。第 4 章で研究課題、提案手法について、第 5 章で今回行った実験について説明する。第 6 章でまとめとする。

2. 研究背景

近年、企業におけるソフトウェア開発は多様化してきている。その中の一つとして分散開発がある。分散開発は複数の拠点に分かれて、それぞれ担当を分担して開発を行い、最後に合わせて一つのソフトウェアを完成させる形態である。昨今の新型コロナウイルス感染症拡大の影響で対面で行われていた開発が行われずリモートで行う形が主流となってきている。ソフトウェア開発プロジェクトには開発者とプロジェクトマネージャーが存在する。プロジェクトマネージャーはプロジェクトの目的達成を目指して、開発者の進捗を常に監視し遅延が発生した場合は原因を早急に突き止め、活動を円滑に進める役割を担っている。近年のソフトウェア開発では、IT 技術の高度化、複雑化によりソフトウェアの信頼性がより一層高まっており、開発者そしてプロジェクトマネージャーへの負担は益々大きくなっている。また、プロジェクトは期日までに成果物を完成させる必要がある為、開発遅延を起こさないようにメンバーの構成や綿密な計画

¹ 創価大学 理工学研究科
Soka University, Graduate School of Science and Engineering

を立てて確実に達成させる必要があり、負担はさらに大きくなってしまっている。さらに、分散開発においてはお互いの進捗状況を逐一把握することが難しく、問題発生時の対処が遅れてしまった場合、結果として遅延が拡大してしまう。

3. 研究目的

本研究では、想定する状況を企業での分散ソフトウェア開発とするのは難しい為、非同期型で進行する分散型 PBL ソフトウェア開発授業を想定する。SNS 上で学生間のコミュニケーションや進捗報告などの発言を行い、状況によって支援を行う必要が出た場合は TA が学生の状況を分析して適切な支援を行う。この授業では、30~40 人規模で実施し、1 グループ 4~5 人程度で構成され、対面での実施は行わず、オンラインでの実施とする。作業に取り組むタイミングは各々の作業をする非同期型で授業は進行する。SNS を使って発言をし学生同士の情報共有や進捗報告を行う。各グループの開発状況を常に監視し、適宜支援を行うのが TA である (図 1)。ここで発生する問題点として、TA は複数のグループを同時に監視しなければならない為、TA の負担が増大してしまう。また、この授業は非同期型で進んでいることから、TA はいつ学生が作業を行なっているか把握することが難しい。さらに、遅延が発生した場合、TA が遅延に気づかず時間が経過してしまい、遅延が拡大してしまう可能性がある。

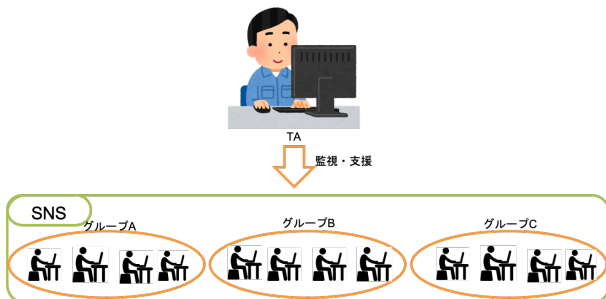


図 1 想定状況

するための方法がない。これにより学生に対して適切な支援を提示することが出来ない。

4.2 提案手法

これらの課題に対し、次のように対処を行う。

提案 1 各学生の状況を数値化した「深刻度」を用いて支援を行うタイミングを判断する

提案 2 深刻度の数値と発言内容から支援内容を Web 情報や過去事例から選択する

システム構成図を図 2 に示す。今回、SNS は Slack を用いている [1]。システムの流れとして、学生が Slack 内で発言をすると Bot はそれが「進捗報告」なのか「自由発言」であるか解析をする。解析後は、DB に過去事例として保存する。1 日の終わりに Bot は進捗報告から学生ごとに深刻度の計算を行う。深刻度の値からどの学生に対して支援を行う必要があるのか判断する。支援情報は Web 情報や過去事例として保存されている DB から抽出して選択し SNS 内で提示する。「進捗報告」は文の初めを進捗報告と書かれており、1 行ごとに作業名や作業内容、作業量など、Bot が進捗を把握するために必要情報を記載されているものを指す。「進捗報告」でないと判断したものは「自由発言」とみなす。進捗報告から 4.3 節で述べる深刻度を計算し支援を行うタイミングと支援情報を収集・選択する。

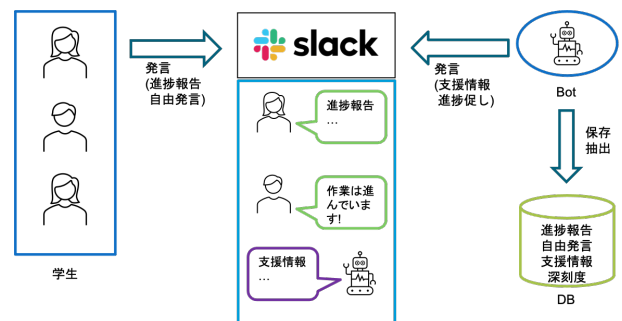


図 2 システム構成図

4. 研究内容

4.1 研究課題

本研究において、BoT を構築するにあたり、生じる課題として以下の 2 点が挙げられる。

課題 1 学生ごとの進捗状況を解析して遅延状況を把握する

課題 2 遅延原因を分析し、状況に適した情報を収集し選択する

TA の役割を自動化する際、BoT は学生ごとの作業進捗を自動で図るための指標がない為、誰が遅延をしているのか把握することができない。また、支援を求めている学生に対して BoT は学生がどのような支援を求めているのか判断

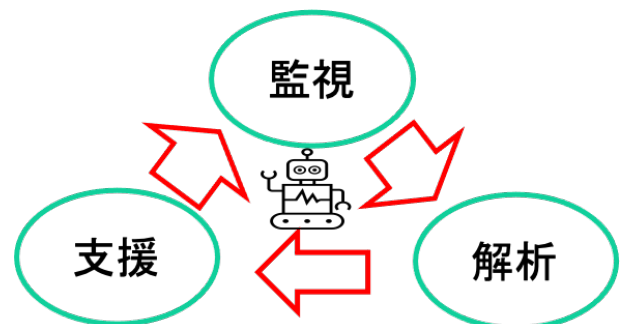


図 3 BoT システムの流れ

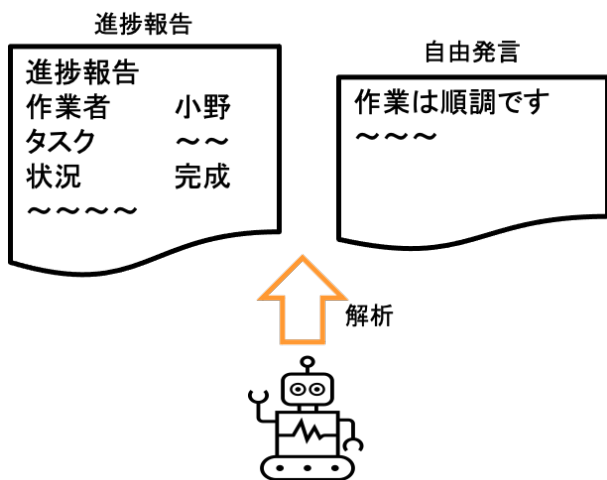


図 4 発言解析

4.3 深刻度

深刻度は作業の遅延がどれほど深刻であるかを数値化したものである。深刻度は学生ごとに計算を行い、学生ごとに状況を分析する。(1) に計算式を示す。Tnp が予定作業量、Tna が実際の作業量、E が経過日数、D が全体日数を表している。今回、作業量はプログラムの行数としている。深刻度は日を追うごとに上昇していくようになっており、数値が高ければ高いほど遅延は深刻であると判断することができる。TA による支援がなかった場合、学生は遅延の深刻さに気づかず作業を進めていく。遅延が続いた状態が長く続いた場合、他のメンバーに影響を及ぼしかねない。これを防止するために深刻度を用いて学生ごとの進捗を測ることで BoT はどの学生の遅延が深刻であるのか判断する指標を示すことができる。

$$\frac{Tnp - Tna}{Tnp} \times \frac{E}{D} \quad (1)$$

深刻度の使用例を図 5、図 6 に示す。X は予定通りの作業量、Y は予定の半分の作業量、Z は作業量が 0 である場合を表している。深刻度表を見てみると、予定通り作業が進んでいない Y と Z は予定通り作業が進んでいる X と比較して深刻度が上昇していることが確認できる。また、Y と Z で深刻度上昇の傾きを比較すると、全く作業が進んでいない Z の方が深刻度の上昇が高いことが確認できる。これにより、学生ごとで状況が違ってくると深刻度の数値も違ってくる為、それぞれの状況に合わせた支援が必要になってくる。

図 7 に示す通り、BoT は深刻度が上昇している学生に対して適切な支援を行うことで遅延を解消し、深刻度の減少を図る。

4.4 支援の仕組み

支援内容を表 1 に示す。今回、支援の種類はキーワードの提示と Web 情報の提示とプログラム内容の確認の 3 種類とした。支援情報の取得からチャット上での提示までの流

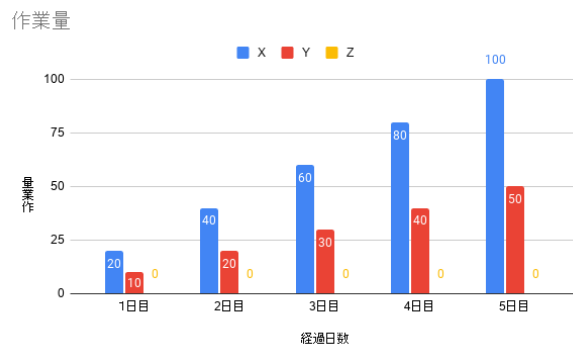


図 5 作業量

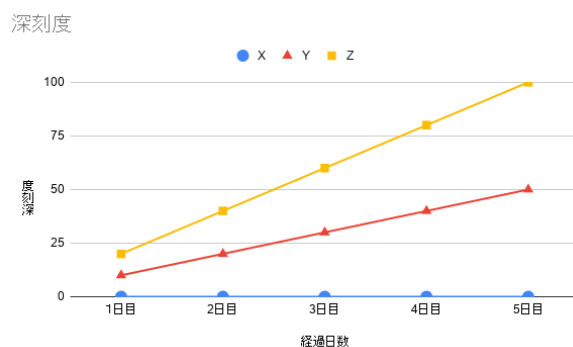


図 6 深刻度表

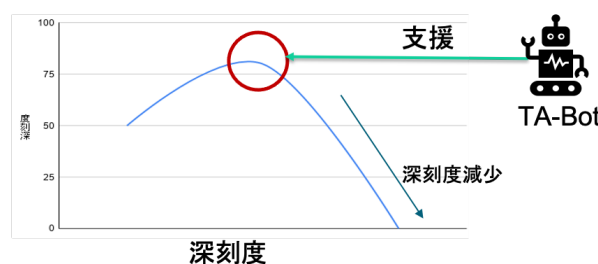


図 7 深刻度の利用

れを図 8 に示す。これらの支援内容は全て DB に保存されている。支援情報の抽出をするために、BoT はどの学生にどの支援情報を提示するのか把握する必要がある。そのために、進捗報告から学生がどのタスクに取り組んでいるのか把握する。次にその学生の深刻度を取得し、その数値から支援内容を決定する。支援情報は「キーワード」、「Web 情報」、「プログラム内容」の 3 つのカテゴリに分けて保存されている。さらにその中にはタスクごとにそれぞれ支援情報が保存されている。BoT は取得した学生の作業中のタスクと深刻度の数値を元に支援情報を抽出しチャット上で学生に提示する。チャット上での支援情報提示の図を以下に示す。(図 9)

支援情報の準備は TA が事前に問題に取り組んでおき、そこから遅延が発生するとみられる箇所を見つけて、それらを解決するためのキーワードを DB の「キーワード」に保存

する。Web 情報に関しては先程保存したキーワードを使って Web 検索をし、取得した URL を DB の「Web 情報」に保存する。今回、Web 検索には Google Custom Search API を利用している。[3] プログラム内容に関しては、TA がプログラム全体の流れを記述したものを DB の「プログラム内容」に保存する。これらの支援情報は過去事例として DB に蓄積しておくことで、今後 BoT が別の学生に対して支援を行う際により適切な支援情報を提示できるようにする。

表 1 支援内容

深刻度	支援内容	遅延原因
小	キーワード提示	技術の知識不足
中	Web 情報提示	文法の知識不足
大	プログラム内容の確認	プログラム内容理解不足

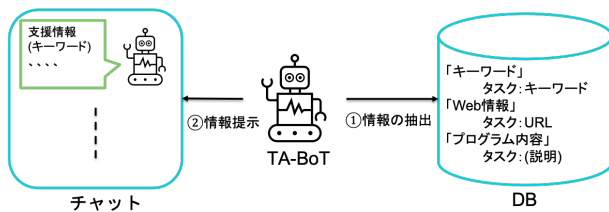


図 8 支援情報取得から提示まで

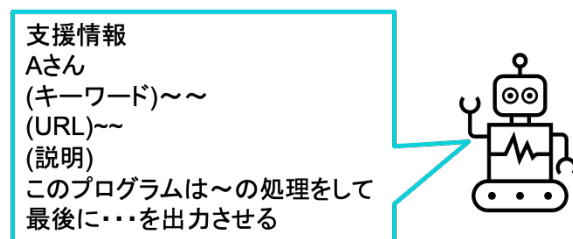


図 9 BoT による支援情報提示

5. 実験

5.1 計画

本研究の評価実験を行うにあたり、実際の想定人数である 30~40 人は人員の確保や規模の大きさから実施が難しい為、人数を 5~6 人の少人数にして行った。第 1 実験では学生の遅延状況を数値化した深刻度の妥当性の確認をし、第 2 実験では深刻度の値から決定した支援内容が学生に対して効果があるか確認を行なった。2 回の実験は共に、被験者は C 言語を学んだ経験があることから C 言語を使った問題を Web 上の問題から用いて実施した。[2] また、問題は事前に TA が作業したものを選択した。深刻度を計算する上で必要な予定作業量も算出した。

5.2 第 1 実験

5.2.1 内容

第 1 実験では、作業遅延に関して定義した深刻度を学生ごとに遅延状況を数値化して遅延状況の指標として有効か評価を行った。対象の人数を 5 人とし、期間を 10 分を 2 回、7 分を 1 回として時間ごとに進捗報告を提出させた。実験中、学生と TA はオンラインで繋がっていたが、実際の状況ではオンラインで繋がって直接学生同士で話し合って作業を進める形ではない為、実験中は Slack 内での会話のみとした。進捗報告には、取り組んだタスク・プログラム行数・状況 (完成または未完成)・コメント・ソースコードを記載させた。コメントには開発状況を学生に自由に記述してもらった。これは過去事例としてどのような発言があったか保存し、今後の実験でより適切な支援情報を提示するためである。今回取り組んだ問題を表 2 に記載する。

表 2 作業内容

作業番号	問題	作業量
D012	絶対値を埋めよ	8
D079	同じ文字	29
C049	エレベータ	19

5.2.2 結果

実際の作業量と深刻度のグラフを図 10、図 11 に示す。A~E が学生を表している。1 回目の作業の時点では予定通り作業が進んでいることから、BoT は作業の深刻さは低いと判断できる。2 回目になると、どの学生も予定作業量の半分となっていることから遅延が発生した。それに伴い、深刻度は上昇し、BoT は遅延状況がより深刻になっていると判断できる。3 回目になると、予定作業量と比較すると遅延が 2 回目の時より拡大した。これに伴い、深刻度はどの学生も 100 に近づいている。BoT は深刻度から 2 回目と比べて事態がより深刻化しているという判断をすることができる。

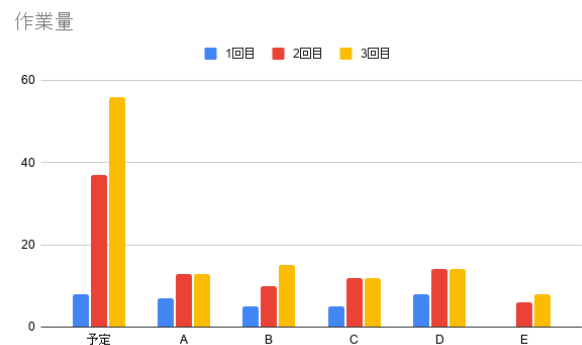


図 10 第 1 実験:作業量

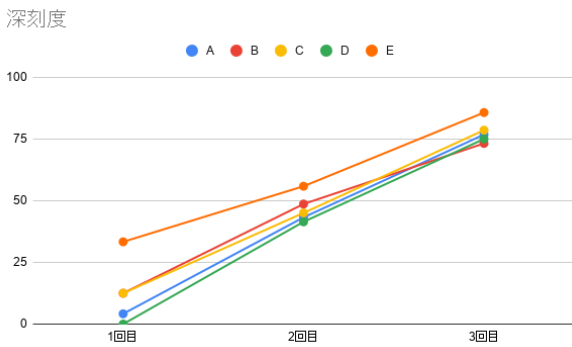


図 11 第 1 実験:深刻度

5.2.3 考察

第 1 実験からは、深刻度を用いて作業の遅延を数値で表すことができた。今回は支援を行わなかったため、深刻度が減少することは確認できていない。支援後の学生の作業への影響の確認が必要である。被験者のうち、1 人は遅延することなく作業を完了して、遅延している学生と深刻度の値の比較ができると考えていたが、どの学生の深刻度も近い数値になった。今回、深刻度が上昇した原因として考えられるのは、1 回ごとの作業時間が短く、問題内容を理解するのに時間を使ってしまった事が考えられる。コメントからも「内容に時間をかけて作業できなかった」といったものが寄せられた。また、オンラインで繋がった状態の PC で開発を行なった学生で、コメントから「パソコンが重くて作業が進まない」といったものも寄せられた。

5.3 第 2 実験

5.3.1 内容

第 2 実験では、第 1 実験で妥当性が確認できた深刻度を元に決定した支援内容が学生にどれほど効果があるのか評価を行なった。対象人数は 2 人で期間は 6 日間とし、毎日作業をする必要はないが毎日進捗報告を必ずとした。進捗報告には取り組んだタスク・作業時間・作業量・状況・コメント・ソースコードとした。ここでのコメントも第 1 実験と同様に過去事例として保存しておくためである。BoT は支援作業の他に 1 日に一回、進捗報告の提出を促す発言をさせた。これは作業をしていなかった日があると、進捗報告の提出を忘れてしまうことが考えられるためである。また、作業をしたのに進捗報告をしないしていると、その日は作業をしていなかったと判断することになるため、それを防ぐためでもある。今回取り組んだ問題を表 3 に記載する。各タスクのキーワードは事前に TA が取り組んだ中で、学生がつまづくと思われる箇所を予測し、それらに関する技術に関するキーワードを選んだ。タスクによっては複数あるものがある。本来、どの情報を学生が求めているのか分析が必要であるが、発言解析から支援情報を決めたものでないため、今回は複数あるものは同時に提示した。

作業番号	問題	作業量
D012	エレベータ	19
D079	複数形への変換	42

5.3.2 結果

実際の作業量と深刻度のグラフを図 12、図 13 に示す。青は A、赤は B の学生を表している。A の学生は 4 日目まで作業を行わなかったため、深刻度が 75 近くまで上昇している。BoT は 5 日目の時点で支援情報を提示した。この時点で行った支援内容は Web 情報とプログラム内容の確認である。実際の支援画面を図 14 に示す。支援情報提示後は深刻度は減少し、最終日の 6 日目には全作業が完了したことから 0 になった。一方、B の学生は作業をしなかった日はあったものの一定量の作業をこなしており、5 日目の時点で全ての作業が完了していた。

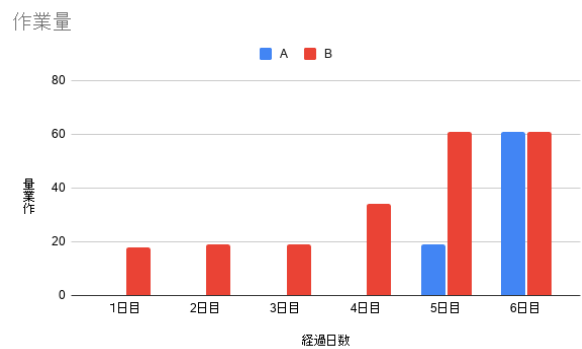


図 12 第 2 実験:作業量

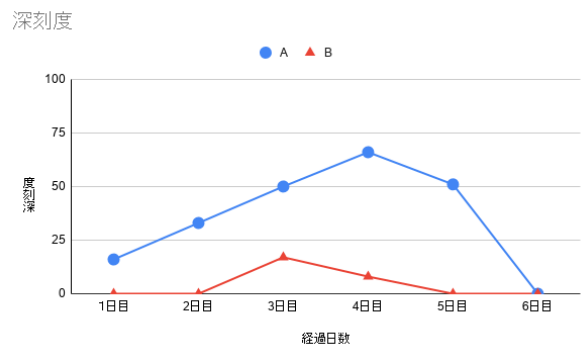


図 13 第 2 実験:深刻度

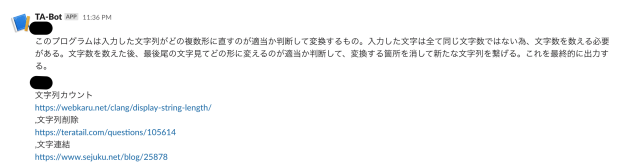


図 14 第 2 実験:支援情報提示

5.3.3 考察

第2実験から、深刻度を元に決定した支援内容が学生に対して効果があることが確認できた。また、2人の学生の深刻度の値が違っていただけから、作業遅延の深刻さに違いが出ていることも同時に確認できた。学生Aへの支援後の影響は、支援後の深刻度の数値が5日目から6日目にかけて減少していたことから、支援内容の提示は作業を進める必要がある認識を学生自身がしたこと、深刻度減少に繋がったと考えられる。深刻度の減少からは支援内容が有効であったのかは確認はできなかった。支援内容の有効性については、今後の実験の際に被験者にアンケートを取り、どの支援が有効であったのか調べる必要がある。

また、実験中は学生同士で情報共有をしながら作業を進めていく形を予想していたが、学生間での自由発言はなく進捗報告のみであった。これは、作業を個人作業としてしまったことから、自分のスキルのみで作業を完了させようとしたことからこのような結果になったと考えられる。今後は、自由発言が行われるように作業の取り組み方や問題の取り組み方の設定を工夫することで評価実験を進めていく必要がある。

6. おわりに

本稿では、非同期型で進行する分散型PBLソフトウェア開発を対象に、学生のSNS上での発言、学生への支援作業を行うTAの作業に着目し、TAの作業効率化を目的にTAの作業を自動化するBoTの提案をした。提案では学生の遅延状況を深刻度を用いることで数値化してBoTが支援を行うタイミングと支援情報を決定する指標を作った。実験結果から、深刻度から学生ごとに作業の遅延を図ることができた。深刻度を元に決定した支援情報の提示は、学生に対して効果があることを確認できた。しかし、支援内容が適切で深刻度が減少したのか、それとも単に支援が行われるまで作業しておらず支援後に作業を進めたから深刻度が減少したのか、数値の変動からは確認できないことがわかった。また、支援情報に学生にとっては余分なものが含まれていることがあり、BoTが学生の状況をより詳細に解析する必要があることがわかった。そのために解析方法に関しては深刻度の数値だけで判断するのではなく、発言内容を解析して学生ごとの状況を把握し、遅延原因を見つけ出して支援情報を集めていく必要がある。また、より適切な支援情報を提示するために過去の実験で収集した発言情報や支援情報をDBに残しておき、それらを機械学習に用いることが必要である。また、学生からの質問がくることも考えられるため、質問が来た際の対応についても考慮する必要がある。実験内容に関しては、被験者のスキルに合わせて課題を設定することに加えさまざまな状況をシミュレーションし、最終的には長期的な実験を行うことでBoTの評価を行う。

参考文献

- [1] slack, 入手先 (<https://slack.com/intl/ja-jp/>) (参照 2021-5-1)
- [2] paiza ラーニング, 入手先 (<https://paiza.jp/works>) (参照 2021-5-5)
- [3] Google Custom Search API, 入手先 (<https://developers.google.com/custom-search/v1/overview>) (参照 2021-5-10)
- [4] 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 (参照 2021-5-10)