

ソフトウェア開発演習を対象としたTA-Botを用いた 進捗監視システムのシミュレーション評価

小野 光一¹ 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において学生の進捗監視及びアドバイスを行うTA(Teaching Assistant)-Botを用いた進捗監視システムのシミュレーションによる評価について検討する. 進捗監視システムには, 学生の進捗を計測する深刻度とその値に基づいてデータベースから学生ごとにアドバイスを取り出すアドバイス選択がある. このシステムを評価するためには実際に人を対象にして繰り返し実験を行わなければならない. しかし, 実験の規模が大きくなってしまふことや学生のプログラミングスキルが異なることから, 実施自体が難しく, こちらの意図する状況を再現できない. そこで, シミュレーションによって学生に開発状況を意図的に作り出すことによって進捗監視システムの評価を行う. 本稿では, 学生の開発状況を作り出すために必要なパラメータを設定し, シミュレーションの一例から学生ごとに異なる開発状況を再現することができた.

Evaluation by Simulation of TA-Bot based on Progress Management System for Software Development

KOICHI ONO¹ RYOZO KIYOHARA² YOSHIAKI TERASHIMA¹

1. はじめに

本稿ではオンラインを用いた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適宜アドバイスを行うTA(Teaching Asssitant) を Bot を用いて自動化する研究をおこなっている. 授業形態がオンラインであることから学生間及び教員との情報共有はSNS(Social Network System) 上で行う. TA は定期的に学生からの進捗報告を元に遅延が発生している学生に対してアドバイスをを行う. Bot がこの動作を行う場合, 学生がどの程度の遅延であるか把握し適切なアドバイスを提示する必要がある. そのために学生ごとに作業遅延を定量化し, その値からアドバイスをデータベースから取り出せるようにした. このシステムを評価するために人を集めて実験を行う必要がある. しかし, 実験規模が大きいことや学生のプログラミングスキル差により意図的な状況を作り出すことが難しく再現性がない. こ

の問題を解決するために, シミュレーション実験を行うことで学生の開発状況を意図的に作り出して評価を行う. これにより多くの実験を行うことができ, アドバイスの提示のタイミングの調整とアドバイスの効果の確認を行うことができる. 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述べ, 第3章で対象とする授業と進捗監視システムの内容について述べた後, 第4章で評価課題, 第5章で評価手法について説明し, 第6章でまとめとする.

2. 研究背景

現在, コロナ禍によりオンラインを用いた授業が急速に普及し定着している. 対面での授業と違い, オンライン授業は学生とのコミュニケーションを取るのが難しいため, 教員は学生の学習状況を把握することは難しい. これにより教員の負担が増大してしまう. [2]

また, 文部科学省では「成長分野を支える情報技術人材の育成の拠点の形成(enPiT)」において, 「ビックデータ・AI分野」, 「セキュリティ分野」, 「組み込みシステム分野」, 「ビジネスシステムデザイン分野」の4分野における高度

¹ 創価大学大学院 理工学研究科
Soka University, Graduate School of Science and Engineering
² 神奈川工科大学 情報学部 情報工学科
Kanagawa Institute of Technology

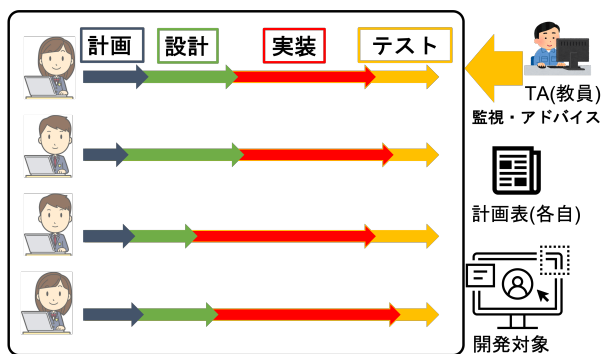


図 1 対象の授業

IT 人材の育成に取り組んでいる．[4] 育成環境を整えるために全国の大学と連携して授業に実施を行なっている．この取り組みは大学のみならず，高校や中学校にも拡大していくことが考えられる．

上記のことからオンラインを用いたソフトウェア開発演習授業が今後も行われていくことが考えられる．

3. 対象の授業とシステム内容

3.1 詳細

対象とする授業は，オンラインを用いたソフトウェア開発演習授業である．授業の構成図を図 1 に示す．授業の参加人数は 100 人程度で，学生はソフトウェアの開発を始める前に各自で計画表を作成し TA に提出する．計画表には何日までに全体の何割の作業が完了しているのか記載する．また，TA が学生の作業状況の把握を容易にするためにどの日までの作業がどの作業段階であるのかも記載する．作業フェーズは，設計・実装・テストの 3 つである．開発作業は学生個人が開発を行いたいタイミングで行う非同期型開発で進行する．学生同士，教員と学生間での情報共有は SNS を用いて行う．学生は定期的に SNS で進捗報告を行う．TA はこの進捗報告から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行う．

3.2 問題点

ここで考えられる問題点として，TA の負担増大が挙げられる．TA は多人数の学生を同時に監視しなければならない．また，学生ごとにプログラミングのスキルは異なってくるため，作業の進行度は異なる．さらに，非同期型で進行しているため，どの学生がどのタイミングで開発を行っているのか把握が困難である．そのため，作業遅延が発生している学生に対してアドバイスが遅れてしまう結果となり，作業遅延が拡大してしまう．

3.3 提案システム

上記の問題を解決するために，TA-Bot を用いた進捗監視システムを提案する．TA-Bot を用いることで TA の役

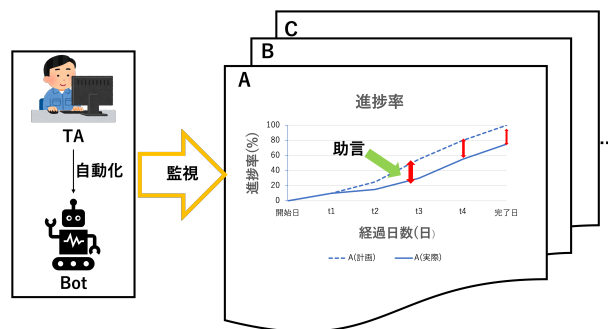


図 2 進捗監視の問題

割である進捗監視とアドバイス提示を自動化することにより，TA の作業効率化をして負担軽減につなげる．このシステムを構成するための要素として以下の 2 つを設定する．

要素 1 学生の進捗監視

要素 2 アドバイスの選択

学生に進捗監視においては，TA は遅延が発生している学生に対してアドバイスを行う．遅延が発生していることを確認するためには計画表での進捗率と進捗報告の進捗率の差分を見る必要がある．しかし，差分のみでは作業遅延が発生していることは把握できても，期限日に近くなっているほど作業遅延が深刻になっていることを表すことができない．そのため，TA-Bot は学生に対してより強いアドバイスを提示することができず，遅延解消につなげることができない．

アドバイスの選択に関しては，TA-Bot はデータベースに保存された情報から学生に対してアドバイスを行う．しかし，どのアドバイスが学生にとって適切であるか選択することが難しい．また，どのタイミングで学生にアドバイスをすべきか判断する必要がある．そのため，アドバイスを選択する際の指標となるものが必要となってくる．

3.3.1 システム構成

システム構成図を図 3 に示す．SNS は Slack を用いており，学生および TA・教員は Slack で情報共有を行う．学生は定期的に進捗報告を行う．進捗報告には進捗率とコメントが記載されている．Bot はこの進捗報告をデータベースに保存し，データベース内の各学生の計画表に記載されている予定進捗率と実際の進捗率から作業の深刻さを表す深刻度を算出する．Bot はこの深刻度の値に基づいてデータベース内にあるアドバイスを選択して学生に対して提示する．これを静的アドバイスと呼ぶ．アドバイス内容は過去の授業でも使用されたものも保存されており，随時追加できるようになっている．また，学生の発言を解析して開発に関するキーワードを取り出し，キーワードを使って得られた web 情報も新たなアドバイスとしてデータベースに保存する．これを動的アドバイスと呼ぶ．

3.3.2 深刻度

深刻度は学生の作業遅延の深刻さを定量化したものであ

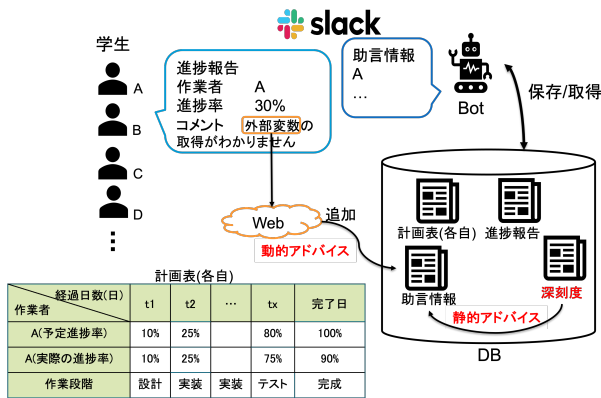


図 3 システム構成図

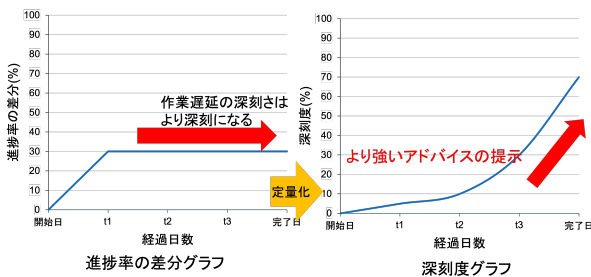


図 4 作業遅延の深刻さの定量化

る。計画表と進捗報告の進捗率の差分を見て進捗報告の進捗率が計画表での進捗率よりも低いと作業遅延が発生していることを確認できるが、その作業遅延がどれほど深刻であるのか表すことはできていない。図 4 では、左のグラフが進捗率の差分、右のグラフが深刻度を表している。進捗率の差分が一定で完了日まで続いていたらとすると作業遅延が発生していることに変わりはないが、完了日に近づくにつれて作業遅延の深刻さは増しているはずである。そこで、右のグラフのように作業遅延の深刻さを定量化することにより、TA-Bot がより強いアドバイスを提示できるようにする。定量化する方法は、計画表と進捗報告の進捗率の差分に日数を重み値として用いることで作業遅延の深刻さを確認できるようにする。これにより開発完了日に近づくにつれて作業遅延はより一層増していることを表すことができ、TA-Bot が学生に対してより強いアドバイスを提示することができるようになる。深刻度の計算式を (1) に示す。Tnp は予定進捗率を表しており、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。

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

3.3.3 アドバイス

アドバイス提示は静的アドバイスと動的アドバイスの 2 種類がある。この 2 種類のアドバイスの違いは情報の取得先が異なることにある。

静的アドバイスはデータベースからアドバイスを取り出して学生に提示するものである。静的アドバイスの提示ま

での流れを図 5 に示す。静的アドバイスでは、アドバイスを取得する際に深刻度を使用する。深刻度は数値が高いほど作業遅延がより深刻になっていることを表しているが、どの数値の時にどのアドバイスを提示するのか判定する基準がない。そこで深刻度内にアドバイスの種類を決定するアドバイス判定閾値を設定して TA-Bot がどのアドバイスの種類を提示するのか判断できるようにした。これにより、それぞれの学生の状況に応じたアドバイスが提示可能となる。この閾値は学生ごとに調節が可能で、提示するアドバイスのタイミングを学生ごとに変更することができる。アドバイスの種類を表 1 に示す。深刻度が小であると判断した際は、開発に関するキーワードを提示して遅延解消を狙う。深刻度が中であると判断した際は、先ほどのキーワードを使って TA-Bot がスクレイピングにより Web から情報を取得する。これを学生に提示する。提示する情報は Web サイトで文法の使用例が記載されているものを使用する。深刻度が大きであると判断した際は、開発を行なっているソフトウェアの処理手順について説明を行う。処理の手順を理解してもらうことで、ソフトウェアの仕様を把握して開発に取り組めるようにする。

アドバイスは過去の授業に提示したものが含まれている。学生には常に新しい情報を提示できるようにするため、Web から新規に情報を取得する。

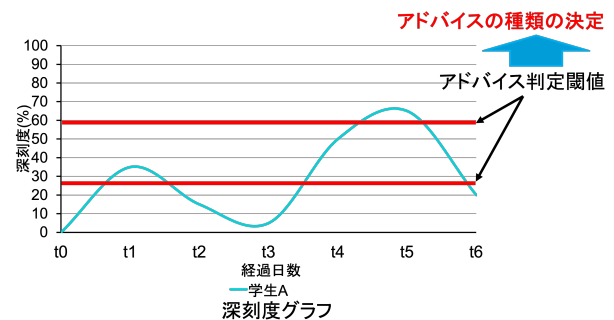


図 5 静的アドバイス

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

動的アドバイスは進捗報告のコメントを解析することによって得られたキーワードをもとに Web から情報を取得して学生に提示するものである。手順を図 6 に示す。学生のコメントを形態素解析することにより、単語ごとに分ける。そこから開発に関するキーワードを取り出す。キーワードに関する情報をスクレイピングを使って Web から情報を取得する。この報を学生に提示してデータベースに

保存する。ここで得られた情報は今後は静的アドバイスとして利用できるようにする。この動的アドバイスを利用することにより、学生に対してより柔軟に問題解決にあたることが可能となる。

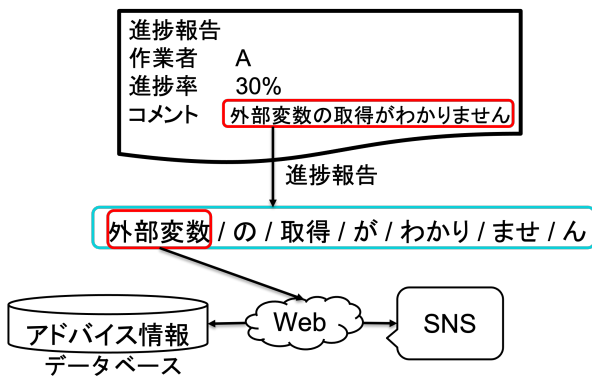


図 6 動的アドバイス

4. 評価課題

上記のシステムを評価するにあたり、学生ごとのアドバイス判定閾値の調整して決定すること、およびアドバイスの効果の2つを確認する必要がある。そのために、実際に被験者にプログラミングの問題を解いてもらうことで、学生ごとの作業状況から TA-Bot にアドバイスを提示してもらう。その後、提示したアドバイスが学生に対して効果を示しているかどうか確認していく。

しかし、人を集めて実施する実験では、想定している状況で実施するとなると、規模が大きくなってしまったり、人を集めることが難しいことから実施は難しい。そのため、人を集められたとしても数回の実験データしか集めることができず評価につなげるにはデータが足りない。また、学生ごとによってプログラミングスキルは異なるため、意図した状況を作り出すことが難しくなってしまう、アドバイス判定閾値の調整やアドバイスの効果の検証ができない。これにより、人での実験は再現性がなく、システムの評価が難しくなる。

5. シミュレーション評価

上記の課題を解決するためにシミュレーションを用いて学生の開発状況を意図的に作り出し、アドバイス提示を行うことで本システムの評価を行う。シミュレーションでの評価を行うことで多くの学生の開発状況を設定して何回でも実験を繰り返し実施することが可能になる。また、アドバイスの効果がなかった際にその原因を調べて同じ状況下で実験を実施することが可能となり、アドバイス提示のタイミングの改善にもつなげることができる。

シミュレーションモデル図を図7に示す。シミュレーションを行う際には、システムへの入力モデルと結果の出

力の表示を決定する必要がある。入力モデルには、用意した学生の進捗報告とする。各タイミングでの進捗報告全てを入力モデルとして用いて深刻度の算出をする。各タイミングでの深刻度に基づいて対象の学生に向けてアドバイスを提示する。出力結果には、各学生でどのタイミングでどのアドバイスをしたのか、その時の深刻度の値を記録する。この出力結果をログファイルとして蓄積していく。

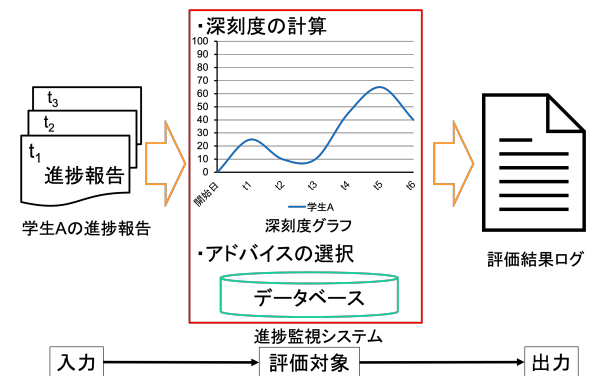


図 7 シミュレーションモデル

入力モデルとなる進捗報告を作成する前に、進捗報告を行う学生の開発状況を設定する必要がある。学生の開発状況を再現するための設定として、プログラミングスキルを考慮した学生のスキル分けと作業遅延が発生することを考慮した開発時期の分割を行った。

この2つの設定を組み合わせることでを使って学生の開発状況を作り出す。この作り出した開発状況をもとにシミュレーション実験を行うことで、学生一人ひとりの評価結果ログを出力する。このログを解析することでどのアドバイスがどの学生に対して効果があったのか、あった場合は深刻度はどれほど減少したのか、効果がなかった場合は深刻度はどれほど上昇したのか、実際はどのアドバイスを提示すべきであったかを考察し、システムの改良につなげる。シミュレーション実験は人での実験と違い、同じ開発状況での実施が何度でも可能であるため、一度実験を行った後に新たに設定が必要であると判断したパラメータ設定や既存のパラメータ設定の細分化を行い、再度同じ開発状況で実験を実施することができる。

5.1 実験

実験では、学生ごとに作業状況が違っていることを確認し、アドバイス判定閾値から提示したアドバイスが学生に効果があったか確認をする。開発を行う対象のソフトウェアは電話交換機の実装で開発期間は6ヶ月である。学生は1週間に一度進捗報告を行うため、全部で25回進捗報告を行う。この進捗報告から TA-Bot は学生に向けてアドバイスをを行う。この授業では過去に TA(人) が実際に学生に向けてアドバイスをを行った経験があるため、これまでに

学生に提示したアドバイスがデータベースに保存されている。TA-Bot はデータベースに保存されているアドバイスをを使って学生にアドバイスを提示する。実験結果はアドバイス履歴を出力して実験の解析を行う。この実験で新たに発生した事例は今後の実験では過去事例として利用するために、随時データベースに保存する。

5.2 学生のスキル分け

開発では、プログラミングスキルの差によりどの学生がどのような形で作業遅延が発生するのかそれぞれ異なる。また、その原因となるものも異なるため、TA-Bot が提示するアドバイスも異なる。そこで、学生のプログラミングスキルを以下の3つに分けた。

- 初級：参考書等を確認しながらコードを書ける
- 中級：参考書等を確認しながら簡単なソフトウェアを開発可能
- 上級：参考書等いらずでソフトウェア開発可能

初級では、基礎的なプログラミング学習は終わっているが、参考書や Web で情報を確認しないとコードを書き進められない学生を指す。中級では、基礎的な内容は抑えられているが、応用部分となると参考書や Web で情報を収集してなんとか開発ができる学生を指す。上級では、概ねのものは自主的に開発が可能で初級者や中級者に対してアドバイスもできる学生を指す。これら3つに分けることで学生の作業遅延の状況を再現することができ、TA-Bot がそれぞれの状況とスキルに適したアドバイスを提示することが可能となる。

5.3 開発期間の分割

ソフトウェア開発では、学生ごとに開発の進捗は異なる。また、作業遅延が発生するタイミングによって作業状況及び作業遅延の深刻さも異なる。これらの多様な学生の開発状況を再現するために、開発期間を以下の3つに分けた。

- 序盤：仕様の理解から計画表の作成
- 中盤：コーディング
- 終盤：拡張期の検討及びテスト段階

開発時期の分割を図8に示す。序盤は、開発するシステムの仕様の理解や開発計画の作成の期間を指す。中盤は、仕様の通りに開発を進める期間を指す。終盤は、機能の拡張やテストなどのソフトウェア完成までの期間を指す。この3つの期間のそれぞれで作業遅延の具合を大小の2つに分ける。遅延具合が大であるとき、作業遅延は非常に深刻であることを指す。遅延具合が小であるとき、深刻な作業遅延であるとは判断しないことを指す。これら2つが序盤・中盤・終盤の3つの時期で考えると、合計で8パターンの学生の開発状況を再現することができる。これに予知多様な学生の開発状況を再現することができる。それぞれの期間での遅延発生により TA-Bot が提示するアドバイス

はどれが適切であるか評価も可能となる。

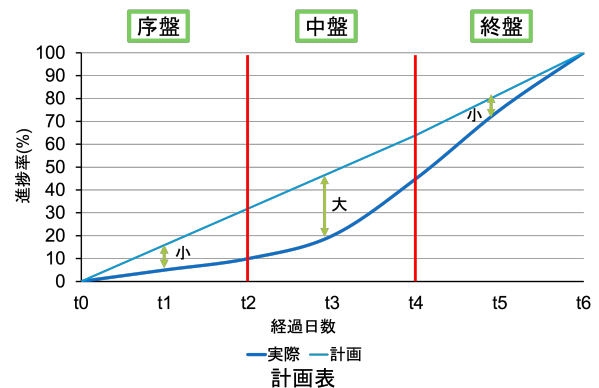


図 8 開発時期の分割

5.4 開発の再現

学生のプログラミングスキルの3パターンと開発時期の分割の8パターンを掛け合わせることで合計で24ケースの学生の開発状況を再現することができる。これらのケースをもとに進捗報告を作成してそれぞれの状況でのアドバイス提示後の学生の反応をシミュレーションを用いて評価を行う。

5.5 シミュレーションの一例

シミュレーションの一例を図9に示す。シミュレーションを行う際、最初に各学生のパラメータ設定を行う。今回は、学生A、学生B、学生Cの3人を用意する。それぞれの学生のプログラミングスキルは、学生Aは初級、学生Bは中級、学生Cは上級とする。計画表に関しては、3人とも同じものであるとする。進捗報告から実際にTA-Botがアドバイスをすると、提示したアドバイスがどの学生にも必ず効果があるとは言えない。そこで、学生のパラメータ設定としてアドバイスの効果があるまたはないのいずれかを設定する。今回は学生Aと学生Bはアドバイスの効果があり、学生Cにはアドバイスの効果がないものとする。これらの設定を行うことでそれぞれの学生の開発状況を作り出し、深刻度のグラフを出すことができる。それぞれの深刻度グラフを見てみると、深刻度が上がるタイミングとその下がり方は学生によって異なっていることが確認できる。これは学生のプログラミングスキルを分けたことによりそれぞれの開発状況の違いが深刻度を使って表すことができていることを示している。そこから各学生へのアドバイスを提示するタイミングとその種類を決定するアドバイス判定閾値の調整を行うことができる。

5.6 結果の解析

シミュレーションの結果の解析は深刻度グラフと出力結果ログを用いて行う。ログファイルは学生ごとに出力する。

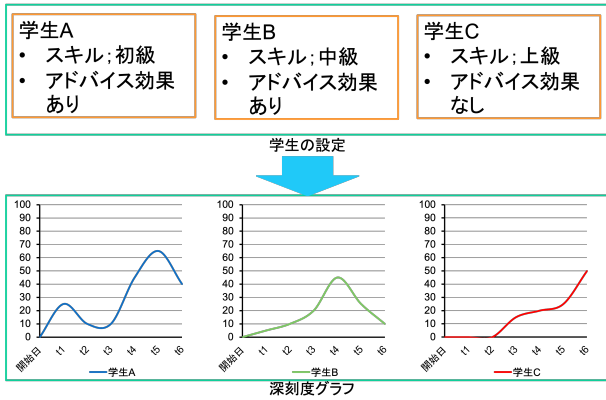


図 9 シミュレーションの一例

出力内容は、日時・アドバイス内容・深刻度である。ログの解析では、各タイミングで提示したアドバイスが学生にとってどれほど効果があったのか深刻度の変動具合を見て確認を行う。この解析結果からアドバイス判定閾値の調整を行う。図 10 に結果出力ログの一例を示す。t1 のタイミングで Web 情報に関するアドバイスの提示をした後は、深刻度は減少していることからアドバイスの効果はあったことが確認できる。しかし、t2 のタイミングでのキーワード提示によるアドバイスでは、t3 の深刻度を見ると上昇していることから t2 で提示したアドバイスの効果はなかったと見ることができる。また、t3 のタイミングで提示した t2 のときとは異なるキーワードに関するアドバイスは t2 の時と同様に深刻度が t4 には上昇していることから効果はなかったと見ることができる。t4 で提示したプログラムの説明に関するアドバイスではその後の深刻度を見てみると減少していることからアドバイスの効果があったことが確認できる。このように各タイミングで提示したアドバイスとその後の深刻度の変動を見ることでどの学生に対して提示したアドバイスがどれほど作業遅延解消に効果を示したのか確認できる。これらの解析結果から学生の開発状況をシミュレーション使って再現し、それぞれの状況でのアドバイスの効果を確認することができる。

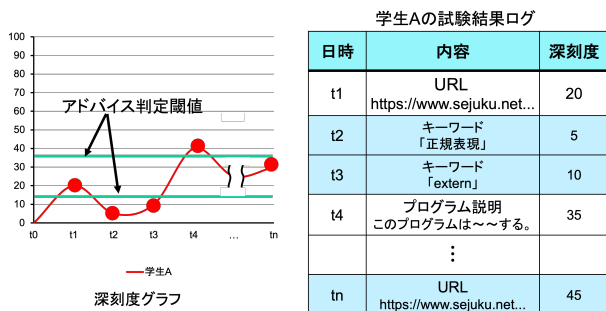


図 10 出力結果の解析

6. おわりに

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いて TA-Bot を用いた進捗監視システムについて述べた。学生の進捗監視では、深刻度を用いて学生一人ひとりの作業遅延の深刻さを定量化し、アドバイスを行うタイミングを決定した。また、深刻度にアドバイス判定閾値を設定することでアドバイスの種類を変更した。

システムの評価をするためにシミュレーションの一例から学生ごとの開発状況を作り出した。学生の開発状況を再現するために、学生のプログラミングスキル分けと開発時期の分割を行なった。この2つの設定から合計で24ケースの学生の開発状況を再現することが可能となる。シミュレーション実験では、アドバイス提示のタイミングの調整とアドバイス効果の確認を行う。人での実験とは違い、多くのケースを繰り返し実施することができるため、学生個人にあったタイミングで効果的なアドバイスを提示することができるようになる。

今後はシミュレーション実験の実施を行い、評価を進める。アドバイスの効果に関する評価に関しては、データベース内のアドバイス情報が不足しているため、アドバイス情報の追加も随時行う。アドバイス提示のタイミングに関しては、シミュレーションを繰り返し行うことで多様な学生の開発状況を模擬し、学生個人に合わせたタイミングでアドバイスを提示できるよう調整を行う。評価実験の実施に関しては、シミュレーションのみではなく実際に人を集めた形式でも実施し、システムの評価を行う。これら2つの実験からシステムの拡張及び修正を進める。

参考文献

- [1] 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
- [2] 文部科学省, "オンライン授業に係る制度と新型コロナウイルス感染症の影響による学生等の学生生活に関する調査" 入手先 (<https://www.mext.go.jp/kaigisiryu/content/000125290.pdf>) (参照 2022-5-23)
- [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) (参照 2022-5-23)
- [4] 文部科学省, 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>) (参照 2022-05-17)