

ソフトウェア開発PBL授業における 進捗管理手法の提案

山本 美幸^{1,a)} 福岡 宏一¹ 清原 良三² 寺島 美昭¹

概要: 近年, ソフトウェアの分散開発が一般になっている. 本研究では, 分散開発を想定した Project Based learning(PBL) 型の授業を対象とし, プロジェクトの開発進捗の管理負担を軽減することで, 開発の遅延を解決するための進捗管理手法を提案する. 提案手法では, 開発進捗を管理し, 進捗に応じた助言を行う. 開発進捗を管理するための指標として, 深刻度を定義する. 深刻度は, 遅延状況を, 他学生への影響を加味して計算したものである. さらに, 深刻度に応じた助言を BOT を介して自動的に提示することで, 従来, 開発進捗を把握する役割をもつ Teaching Assistant(TA) の役割を自動化することが可能になる. 提案手法を用いることで, プロジェクトの進捗管理負担を軽減し, 遅延の影響を抑えることが期待できる.

キーワード: 非同期情報共有, PBL, TA, 進捗管理

A Study of Progress Management Method for Software Development Project-based Learning

Abstract: Nowadays, distributed software development is a popular style for in software project. The proposed method of progress management is to suppress a delay time of development by reducing the management load of a project development progress, targeting Project Based learning type classes assuming distributed development. The proposed method manage development progress and give advice according to a progress situation. Our method defines "severity" as a factor for managing development progress. The degree of "severity" is calculated by the delay situation of taking into account the progress situation on other students. In addition, it becomes possible to automatically advice depending on "severity" the role of Teaching Assistant (TA). Therefore, it can be expected to reduce the load of the project management and suppress a progress delay time.

Keywords: Project Based Learning, Teaching Assistant, Asynchronous communication, Progress management

1. はじめに

本研究は, 分散開発を想定したソフトウェア開発型の PBL 授業における, プロジェクトの遅延を抑えることを目的とした進捗管理手法を提案する. 授業では, 学生数名がプロジェクトを組み, 一つのソフトウェア開発を行う. プロジェクトの進行状況を監視し, 助言を行う役割として, TA が存在する. TA の存在は, プロジェクトの遅延を早期

解決に導き, プロジェクトを円滑に進める効果がある. 分散開発を想定するため, 学生は相互に独立して開発を進める. 本稿では, プロジェクトの遅延を抑えることを目的とし, 分散拠点間の独立性を妨げることのないよう, 非同期型情報共有を用いた進捗管理を行う手法を提案する.

2. 研究背景

2.1 対象とするソフトウェア分散開発型 PBL 授業

本研究では, ソフトウェアの分散開発を想定した PBL 型の授業を対象にする. 近年, 企業等で分散開発は増加傾向にある. また, 大学等では, アクティブラーニングの導入により, プロジェクトベース型の授業形態を行う大学が

¹ 創価大学大学院 工学研究科
Soka University, Hachioji, Tokyo 192-8577, Japan

² 神奈川工科大学 情報学部 情報工学科
Kanagawa Institute of Technology

^{a)} e18m5204@soka-u.jp

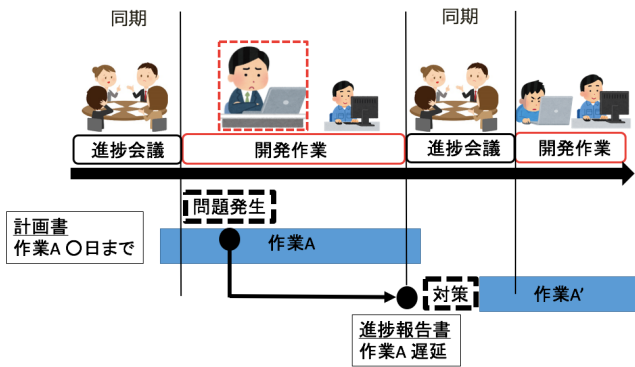


図 1 従来の開発形態

増加している。そのため、ソフトウェア開発の分散開発を想定した PBL 授業も今後増えると予測される。本研究で想定する授業では、分散開発を想定するために、学生が独立して開発を行い、対面で開発する時間を設けず、定期的に進捗報告を提出することで、プロジェクトを進度を把握し、開発を進める。授業で発生する課題として、分散開発において発生する課題に加え、PBL 型の授業特有の課題も発生する。

2.2 ソフトウェア分散開発型 PBL 授業における課題

近年は、ソフトウェアの内製化が進み、IoT 開発等に用いられるスパイラル開発が増加している。こうした開発では、仕様変更や作業遅延が頻繁に発生することから、遅延の危険性がある問題に対し、迅速な対応が求められる。また、リモートワークの推奨により、分散拠点で開発を行う形態も増加している。

分散開発における課題として、プロジェクト状況がプロジェクトメンバに伝わらず、問題発生時の対応が遅れるという課題がある。分散拠点間では、互いの独立性を妨ぐことなく作業をするため、定期的に関われる進捗会議で、対面で進捗を確認しあい、問題を報告する。独立性を保つ目的は、互いの開発に干渉する時間を抑えることで、個々の開発作業に集中するためである。進捗会議間の進捗度の管理は、進捗会議前に提出する進捗報告書にて行う。従来型の開発形態では、進捗会議間にプロジェクトメンバ間のコミュニケーションを取ることを前提としていない。そのため、進捗報告書だけの情報共有では、図 1 のように、問題発生から解決策を議論し、対策をとるまでの時間が開いてしまうという特徴がある。さらに、進捗会議間は、他のメンバの状況が把握できず、問題の早期発見が困難である。

次に、ソフトウェア開発 PBL 授業における課題として、TA のプロジェクト状況管理の負担が挙げられる。TA は、複数のプロジェクト状況を把握し、状況に応じた助言を行うことで、プロジェクトが円滑に開発に進むように支援する。プロジェクトごとに学生の能力は異なり、進め方も異なるため、プロジェクトの進度を正確に把握することは難

しく、複数のプロジェクトをリアルタイムに監視する必要がある。そのため、プロジェクトを監視する負担と状況进行分析する負担、プロジェクト状況に応じて適切なタイミングで助言をする負担がある。

以上のように、学生が独立して分散開発を行うことを想定した、ソフトウェア開発 PBL 授業における課題は、次の 2 点である。

課題 1 開発プロセスの遅延の拡大

他学生の遅延情報が把握できず、遅延がプロジェクト全体に与える影響がわからないため対応が遅れ、発生する。

課題 2 進捗管理と助言を行う TA の負担

TA は複数のプロジェクトを監視しなければならず、個人の作業遅延に敏感に反応することが困難であるため負担がかかる。

2.3 既存研究

これまでにソフトウェア開発における、プロジェクトの情報共有、進捗状況管理の負担軽減に関する研究がなされてきた。プロジェクト間の進捗管理や進捗状況把握のために、情報共有に加え、コミュニケーションをとることは重要である。また、コミュニケーションにより、課題点の共有と課題解決の議論が円滑に進む。さらに、プロジェクトにおいて開発プロジェクトの情報から、重要課題を早期に発見し解決をリードするための議論は、プロジェクトを計画通り遂行するため必要である [1][2]。

大平らは、プロジェクトにおける進捗状況を可視化し分析することで、管理者を支援しているシステムを提案している [3]。このシステムは EPM(Empirical Project Monitor)と呼ばれ、リアルタイムに開発支援フリーウェアと連携することにより、開発履歴データを収集することで定量的データ分析ができる。そして、管理者や開発者は常に進捗状況や作業情報を客観的に把握することが可能である。そのため、プロジェクトの問題点の発見と的確なプロジェクト管理が容易になる。問題が発生した場合は、プロセスを改善することで対応する。

また、玉田らは、プロジェクトの障害修正対応の進捗度を分析する障害対応遅延分析や、プロジェクト遅延リスク検出分析を行う分析器補助ツールを開発している [4]。

以上のように、ソフトウェア開発において、開発者の行動や進捗状況可視化による進捗管理の負担軽減を行う研究が盛んである。このことから、情報共有や進捗管理、進捗管理負担軽減の重要性が伺える。しかし、ソフトウェア分散開発型 PBL 授業に適用した場合、個人の進捗が全体のプロセスに与える状況が把握できない点が不十分である。

授業に視点をあてると、ソフトウェア開発 PBL 授業に関する研究が、様々に行われていることがわかる。福安らは、ソフトウェア開発 PBL におけるチームの状況に合わ

せた適切なフィードバックが重要と考え、指導者の評価負担の軽減を目的に、状況把握支援環境のツール開発と、3つの開発フェーズに合わせた適切なフィードバックについて検討している [5]。しかし、指導者の評価負担の軽減だけでは、学生プロジェクト遅延を軽減することができないため、本研究の想定する授業に適用する場合、学生に助言を行う点が不足している。

TAの負担軽減に関する研究では、プログラミング講義に関するものが多く、プロジェクト開発を想定していない状況で、コーディングの進捗状況を管理することを支援する研究 [6] や、TAが巡回する際に学生に声をかけやすくする研究 [7] などがある。しかし、プロジェクト開発では、個人の遅延だけでなく、プロジェクトに与える影響を考慮する必要があるため、既存手法を適用した場合、適切なタイミングで助言を行う点が不十分となる。

3. 進捗管理手法

3.1 想定する授業の流れ

授業の前提を次の通りに設定する。授業は半年間のプログラムであり、1つのプロジェクトは、3から5名の学生で構成される。1つのプロジェクトで1つのソフトウェアを完成させる。分散開発を体験する目的があるため、対面での開発は原則行わない。プロジェクト開始時に、プロジェクト計画を立て、計画書を作成する。計画書には、作業ごとに、余裕日を設け、1週間で行う作業を1作業と決める。また、開発を進める際に、非同期情報共有ツールを用いて、コミュニケーションをとる。開発進捗をはかるため、定期的に進捗報告書の提出を行い、対面での議論は原則しない。独立性を妨げない範囲で、進捗報告間の非同期型情報共有を行う。作業中に問題が発生した場合は、非同期情報共有ツールを用いて、メッセージのやり取りをし、連携をとる。問題が生じたらすぐ報告するようにルールを設ける。また、TAに対して質問がある場合にも非同期情報を介して発言する。

3.2 3段階支援モデル

本研究では、図2のように、ソフトウェア分散開発型PBL授業に適する進捗管理手法として、SNS等の非同期情報共有と3段階支援モデルで構成される手法を提案する。従来の分散開発に非同期情報共有を取り入れることで、学生の独立した開発を妨げることなく、進捗情報や学生間の議論情報が取得できる。非同期情報を活用し、進捗管理と学生への助言を行うために、3段階支援モデルを定義する。

ソフトウェア開発PBL授業における課題解決のための対策として、課題1、課題2のそれぞれ対し、以下のように対処する。

解決1 プロジェクト内での進捗状況を深刻度を用いて可

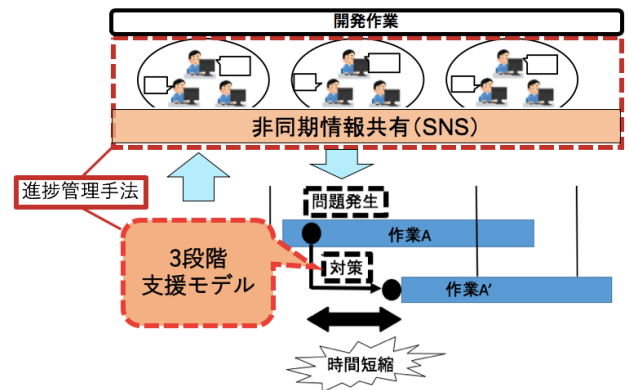


図2 非同期情報共有を取り入れた開発の場合

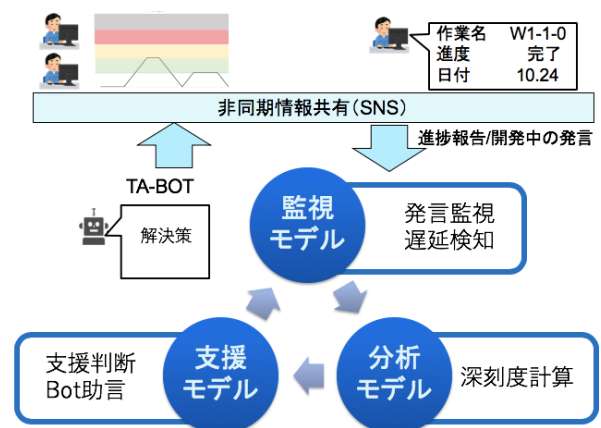


図3 3段階支援モデル

視化し、遅延を早期発見

解決2 TAが行うプロジェクトに対する支援を自動化し、負担を軽減

解決1により、プロジェクト内での進捗状況の早期共有と解決を実現する。非同期型情報共有では、情報共有ツールを用いて、メッセージのやりとりや学生の開発作業進捗を共有する。プロジェクト状況を深刻度という尺度を用いて可視化することで他学生の状況がわかり、遅延発見につながる。さらに、解決2により、TAが進捗を管理する負担が減り、深刻度に応じた助言をBOTを介して非同期情報共有上で支援することで、プロジェクトに助言する負担も軽減できる。

以上の解決策を実現するために、本手法では、図3のように非同期情報を活用した、「監視」「分析」「支援」の3項目のモデルを定義する。第一は「監視モデル」である。これは、進捗を管理し、遅延の早期発見と、遅延の影響を予測するためのモデルである。第二は「分析モデル」である。進捗状況を分析し、他の学生の状況を踏まえた個人の遅延を深刻度として計算する。第三は深刻度に応じて支援を自動化する「支援モデル」である。この3項目のサイクルを繰り返すことで、遅延検知後に問題解決をスムーズに対処する工程を支援し、プロジェクトにおける遅延の影響を抑える。

3.3 監視モデル

監視モデルでは、非同期情報共有における発言を常時取得し、発言監視を行い、プロジェクトの遅延を検知する。監視する発言には、開発に関する発言と進捗状況に関する進捗報告の2種類がある。プロジェクトの遅延を検知するためには、進捗報告の内容を用いる。学生は、フォーマットに従い、定期的に進捗報告を行うため、自動的に進捗報告を振り分け、データベースに格納することができる。

遅延検知は、発言情報から振り分けられた進捗報告と、プロジェクト開始時に学生が決めた計画書を用いて行う。それぞれの情報は、図4のように構成されている。遅延検知計算では、進捗報告の作業番号、報告日、進捗度と、計画書の作業番号、作業期限日、余裕日数を使用する。

計画書	進捗
作業番号: W1-1-0 作業日数: 4 作業開始日: 2018/10/1 作業終了日: 2018/10/4 余裕日数: 3 先行作業: 後続作業: W2-1-0 作業内容: 基本機能1 作業者: 塩田	作業番号: W1-1-0 報告日時: 2018/10/6 進捗度: 未完了 コメント:

図4 データベース上の計画書と進捗情報の例

● 遅延検知計算

データベース上の進捗情報の報告日を T_n 、計画書の作業終了日を T_p とする。進捗情報登録時に、進捗情報の進捗度が「未完了」の作業を対象とする。遅延検知時の手順を以下に示す。

- 1). 学生の進捗情報をデータベースに登録
- 2). 遅延判断の余裕日数を T_s とする。 T_n と T_p とを以下のように比較し、 T_s より小さければ遅延と判定する

$$(T_p - T_n) > T_s \quad (1)$$

- 3). 遅延情報をデータベースに登録

以上のように、遅延検知に進捗報告は使われる。

進捗報告以外の発言情報の履歴を管理し、活用する目的は、2つある。1つは、非同期情報共有上で議論された遅延原因の解決策等を、今後同じような問題が発生した場合に再利用するためである。2つは、学生がTAに対して質問する場合、質問に対して自動応答するためである。過去の解決策の再利用や、質問の自動応答に関しては、3.5の支援モデルで述べる。

3.4 分析モデル

分析モデルでは、監視モデルにて出力された遅延作業情報を用いて、作業ごとにプロジェクトに与える影響を計算

し、プロジェクトの深刻状況を分析する。従来のように、作業単独の遅れだけでなく、他の学生の影響を加味したプロジェクト全体の状況を表すことが目的である。プロジェクトの状況を表す指標として、“深刻度”を定義する。深刻度は、プロジェクト状況によりLEVEL1からLEVEL5までの5段階を定義する。5段階にレベル分けを行う判断基準として、遅延作業以外の学生にも遅延の影響があるかどうか、余裕日数がどれほどあるか、遅延原因は何か、の3点を余裕日数と結合点との関係を分析し、判断する。

余裕日数とは、プロジェクト初期に入力する計画書の中で、クリティカルパス上にない作業に設けられた作業マージンである。結合点とは、作業関連図において、図5で示すような作業と作業を取りまとめる地点である。ポイント地点を時間順に結合点1、結合点2、... 結合点Nと定義する。図5を例にすると、学生1の結合点1までの余裕日数は3日であり、学生2の結合点1までの余裕日数は0日となる。

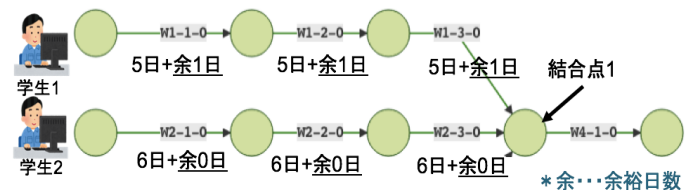


図5 余裕日数と結合点

計画書には、プロジェクト初期段階であらかじめ先行作業が登録されている(図4)。そして、プロジェクト初期段階で、先行作業を紐づけた作業関連図は作成済みである。例として、W3-2-0という作業が遅延した場合の作業関連図を図6に示す。図6のW3-2-0に続く作業を取りまとめる結合点を、最短結合点と呼ぶ。W3-2-0の遅延日数がW5-1-0の作業開始日を過ぎてしまう場合には、他学生に遅延の影響が出ることになる。このように結合点を用いて、今後遅延の影響を受ける学生を抽出する。

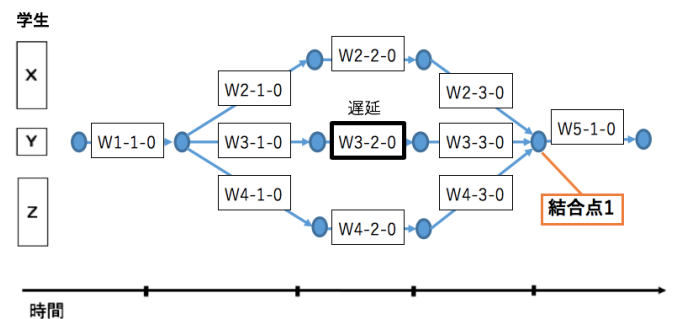


図6 作業関連図の例

余裕日数と結合点の関係、他学生への影響、遅延原因を分析し、深刻度としてレベル分けし、整理したものを表2に示す。このように、レベルが上がるごとにプロジェクトの状況が深刻であることを表す。

LEVEL1, LEVEL2 は、作業の遅延が、当初計画の余裕日数でカバーできる軽度の遅延であるため、学生個人の知識、経験不足によるものと判断する。LEVEL1 と LEVEL2 の違いは、遅延作業から最も近い最短結合点までの、当初計画の余裕日数から遅延日数を引いたものが、当初計画の余裕日数の 50%以上か 50%未満かどうかという点である。

LEVEL3 は、作業の遅延が、最短結合点までの余裕日数だけではカバーできず、他学生に影響を与える遅延であるが、最短結合点の次の結合点までの余裕日数まで考慮すれば、カバーできる遅延である。遅延原因を早期対処できれば、プロジェクト終了日までに間に合わせることができる程度の遅延である。

LEVEL4 は、作業の遅延が、最短結合点までの余裕日数だけではカバーできず、最終結合点までの全ての余裕日数を足し合わせなければカバーできない遅延である。この場合、プロジェクトの学生と協力し、早急に遅延を対処しなければ間に合わない、深刻な状況である。

LEVEL5 は、全ての余裕日数を考慮しても、カバーできない遅延であり、計画の見直しが必要な状況である。

3.5 支援モデル

支援モデルは、従来 TA がプロジェクトに対して行う助言を、自動化するため、非同期情報共有ツール上で、BOT を用いて、プロジェクトの”深刻度”に応じた支援をする。

支援の種類は、(A) 常時プロジェクトに対して支援を行うものと、(B) 深刻度に応じて支援を行うものの 2 種類がある。(A) の常時行う支援では、

- i). TA に対する質問応答
- ii). 作業関連図表示
- iii). 深刻状況グラフ

の 3 点がある。1 点目の TA に対する質問応答では、学生が発言した作業に関する質問に対して、過去の質問履歴と回答履歴から自動で検索し、回答するものである。データベース上に、過去の質問と回答を対応させたテーブルを作成しておき、学生が発言した質問内容をデータベースと照らし合わせることで、回答を抽出し、BOT を介して学生に応答する。

2 点目の作業関連図表示では、プロジェクト初期段階で作成した作業関連図を表示し、作業の遅延状況に応じて、作業関連図上のノードとリンクを色付けて表示する。プロジェクト全体の作業状況をリアルタイムに可視化することで、どの作業がプロジェクトの遅延に大きく影響を与えるのかを学生に示すことができる。遅延している作業を赤、影響がある作業を緑で示す。

3 点目の深刻状況グラフは、図 7 のように、作業の進捗情報に対する深刻度をグラフ化したもので、プロジェクト内の他学生と個人の作業がプロジェクト状況にどのように影響するかを認知する効果がある。例えば学生 1 の作業が

順調に進んでいたとしても、学生 2 が大幅に遅延した場合、学生 1 と 2 の作業を T4 地点で結合する際に、学生 1 の深刻度は、学生 2 の影響を受け、共に深刻なレベルになると予測できる。このような状況を学生が、グラフを見ることで予測することができる。

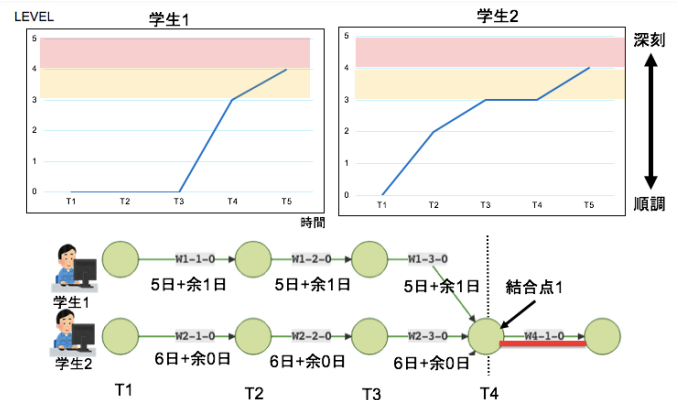


図 7 深刻状況グラフの例と作業関連図

(B) の深刻度に応じて行う支援では、深刻度の LEVEL ごとに内容を変えて、プロジェクトに対する支援を行う。プロジェクトの状況は大きく分けて 3 つあり、(a) 他学生に影響を与えない程度のクリティカルパス上にない遅延、(b) 他学生に影響があり、技術的な原因の遅延、(c) 他学生に影響があり、計画見積もりが原因の遅延である。(a) の状況で TA は、プロジェクト内で遅延情報を共有し、問題の対処をプロジェクトで解決するように助言する。(b) の状況で TA は、技術的な原因に対する解決案を助言する役割がある。(c) の状況で TA は、計画の見直しを助言する。これらの TA の業務を BOT を用いて自動化するため、次の 3 点を支援する。

- a). 遅延作業・影響作業に遅延情報を通知
- b). 遅延作業に遅延作業の解決策を提示
- c). 今後の遅延作業予測とプロセス再構築を助言

(a) の状況の例と、支援内容のイメージを図 8 に示す。学生個人の遅延で、プロジェクトの他学生に影響がない状況のため、遅延した学生に対して遅延状況を認識させることと、今後遅延し続けた場合に影響を受ける学生への遅延状況に関する情報共有が必要である。そのため、BOT による支援内容は、遅延作業と遅延作業、遅延日数、今後影響が出る作業とその作業の情報の、結合点までの余裕日数をプロジェクトメンバに通知する。LEVEL1 と LEVEL2 の違いは、遅延状況を共有する範囲が異なる点であり、LEVEL1 では、遅延作業だけに遅延状況を通知し、LEVEL2 では、他学生にも遅延状況を通知する。

(b) の状況を例と、支援内容のイメージを図 9 に示す。(b) の状況では、遅延の影響が他の作業員に出てきており、早期解決が必要な状況であるため、遅延作業員に対して解

表 1 深刻度

	プロジェクト遅延状況	分析	遅延原因
LEVEL1	他学生に影響しない	最短結合点までの余裕日数が計画時の 50%以上	知識/経験不足
LEVEL2	他学生に影響しない	最短結合点までの余裕日数が計画時の 50%未満	知識/経験不足
LEVEL3	他学生に影響する	最短結合点の次の結合点までに余裕日数がある	技術的スキル不足
LEVEL4	他学生に影響する	最終結合点までの余裕日数が閾値以上ある	計画見積り誤り
LEVEL5	他学生に影響する	余裕日数がない	計画見積り誤り

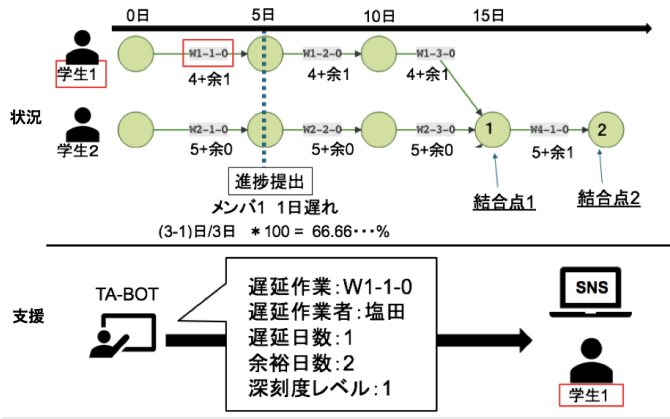


図 8 LEVEL1 の状況の例と支援

決策について助言をする必要がある。支援では、BOT による助言として 2 種類ある。1 点は、過去の履歴を検索し、解決策を提示する支援であり、2 点は、進捗報告の内容と計画書の作業内容から、web 上の情報を検索し、支援を行うものである。

(c) の状況は、遅延の影響が深刻であり、計画の見直しが必要な段階である。そのため、支援では、BOT がプロジェクトに対し、計画見直しを促すとともに、現状のまま作業を進めた場合の遅延予測状況を表した、深刻度グラフをプロジェクトに通知する。

(a), (b), (c) それぞれの支援と深刻度の状況をまとめたものを図 9 に示す。この図の「○」は、支援のタイミングを表す。図 9 で示すように、深刻度のレベルが上がり、深刻さが増す程、支援内容は増加する。

(B)	支援内容					
	(a)	(b)	(c)	(d)	(e)	(f)
深刻度	遅延情報	影響情報	Web情報	過去事例	深刻度グラフ	計画見直し
LEVEL1	○					
LEVEL2	○	○				
LEVEL3	○	○	○	○		
LEVEL4	○	○	○	○	○	
LEVEL5	○	○			○	○
期待される効果	遅延認識		技術原因解決		作業効率向上	

図 9 深刻度と支援

4. 実装

4.1 実装方針

3 章の進捗管理手法で定義したモデルを用いて進捗管理

表 2 実装諸元表

ソフト名	version	内容
MAMP	4.1	Web プラットフォーム
Apache	2.2.31	Web サーバ
MySQL	5.6.34	関係データベース管理システム
MacOS	10.13.6	実装環境
Slack	3.0.2	チームコミュニケーションツール
Python	3.7.0	言語

システムを実装する。非同期情報共有ツールとして、Slack と web ブラウザを用いる。Slack は、グループごとに情報共有スペースを作成でき、グループ内でいくつかのスペースを作り、話題ごとに議論することが可能である。学生は Slack で発言を行い、web ブラウザで作業関連図と深刻状況グラフを閲覧する。TA の作業を自動化する目的の BOT は、Slack 上に導入する。情報共有ツールは、分散拠点間の独立した開発作業を妨げることないよう、時間を制約しない、チャット形式をとる。

4.2 動作確認

実装環境を、表 2 に示す。サンプルのプロジェクトを作成し、一部の支援機能について動作を確認した。常時支援する機能である作業関連図の深刻状況グラフは、図 10 と図 11 のように表示される。

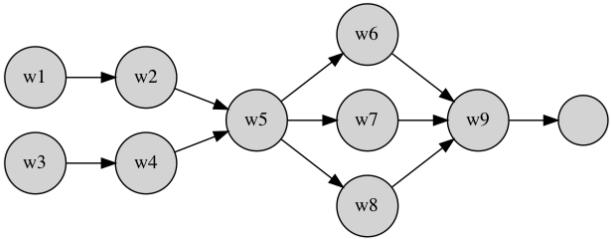


図 10 作業関連図

深刻度に応じて行う支援では、LEVEL2 は図 12 のように表示され、LEVEL3 は 13 のように表示されることを確認した。

今後のシステムの実験方針として、実際に学生同士のプロジェクトを何ケースか作成し、実験を行う場合、評価方法としてアンケートを行う手法がある。しかし、学生の主観にデータが偏ってしまう可能性がある。さらに、何ケースか同じ学生同士を集めたプロジェクトを行う場合でも、

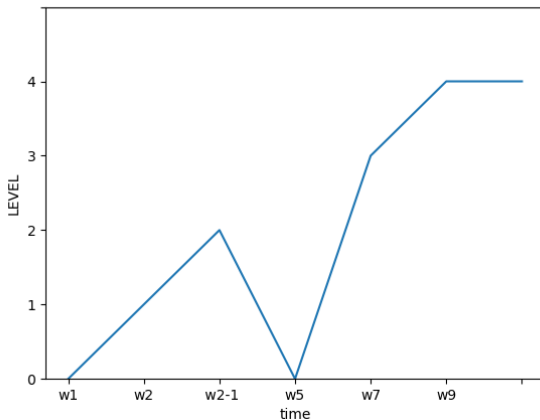


図 11 深刻状況グラフ

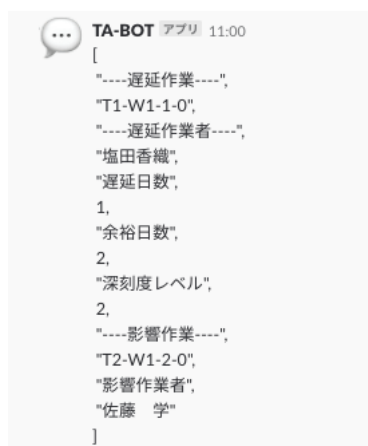


図 12 LEVEL2 の支援 (遅延情報・影響情報通知)



図 13 LEVEL3 の支援 (Web 情報取得)

作業のレベルを毎回同じ内容のプロジェクトにすることは困難であり、再現性がない。

そこで、学生を集めた実証実験を行うと同時に、様々なプロジェクトの状況を想定した、遅延シミュレーションを行う。様々な遅延を想定することで、実際に人手で実験する場合に比べて多くのデータ収集を行うことができる。そのため、遅延シミュレーションができるよう、システムを拡張する。具体的にはプロジェクトの初期計画の生成と、進捗報告を複数生成し、様々な状況のプロジェクトに対す

る遅延の予測を分析できるようにする。

5. 今後の課題

今後は、プロジェクトの試行実験と、様々なプロジェクト状況の自動化シミュレーションを行い、遅延の予測を立てられるよう、あらゆる状況を想定したデータを作成する。そして、長期プロジェクトでの検証と複数のプロジェクトでの検証を行う。実験を行う際に、過去事例の作成や支援機能に対する効果等のデータ数を増やすことが課題である。データを蓄積することで、分析モデルに学習させ、遅延予測を行う機能を追加する。遅延予測では、過去の発言情報や開発遅延の傾向を分析し、精度向上を行う。

6. おわりに

本稿では、分散開発を想定したソフトウェア開発 PBL 授業における、進捗監視手法を提案した。提案手法では、分散開発開発中の問題の早期解決と TA の負荷軽減という 2 点の目的を実現するため、非同期情報を用いた 3 段階支援モデルを定義した。手法により、(1) プロジェクト内での進捗状況を深刻度を用いて判断することと、(2) TA の役割であるプロジェクトに対する助言を自動化することができる。また、手法を元に、進捗管理システムの基本機能を開発した。今後は、マンマシンインタフェースの改良と、シミュレーションを行うための拡張を行う。そして、学生による実証実験とシミュレーションを行う。今後、分析モデルを用いて遅延を予測するため、大量のデータを蓄積し、学習を用いて精度向上を行う機能について検討する。

参考文献

- [1] 佐藤義男:改定新版 PMBOK 第 4 版による IT プロジェクトマネジメント実践法, ソフトリサーチセンター (2009).
- [2] 小浜耕己: 図解で磨くプロマネ技術実践マニュアル, 日経 SYSTEMS(2009).
- [3] 大平雅雄, et al. "ソフトウェア開発プロジェクトのリアルタイム管理を目的とした支援システム." 電子情報通信学会論文誌 D 88.2 (2005): 228-239.
- [4] 玉田春昭, et al. "プロジェクト遅延リスク検出を目的とするソフトウェア開発プロセス可視化ツール: ProStar." 奈良先端科学技術大学院大学テクニカルレポート, NAIST-IS-TR2007002 (2007).
- [5] 福安直樹, et al. "チーム内の役割分担を考慮したソフトウェア開発 PBL の評価基準と状況把握支援." 電子情報通信学会論文誌 D 98.1 (2015): 117-129.
- [6] 井垣宏, et al. "プログラミング演習における進捗状況把握のためのコーディング過程可視化システム C3PV の提案." 情報処理学会論文誌 54.1 (2013): 330-339.
- [7] 市村哲, 梶並知記, and 平野洋行. "プログラミング演習授業における学習状況把握支援の試み." 情報処理学会論文誌 54.12 (2013): 2518-2527.