

# テキストチャットによるペアプログラミング学習システム： 学習者同士のインタラクションに着目した検討

佐々木 孝輔<sup>†1</sup> 林 勇吾<sup>†2</sup> 井上 智雄<sup>†2</sup>

**概要：**プログラミング学習手法の1つに、コーディングを行う Driver と、Driver に対してアドバイス等を行う Navigator の役割に分かれて作業を進めるペアプログラミングがある。このペアプログラミングにおける学習者の行動および会話と、学習の成否については関係性があることが報告されている。本研究では分散同期環境を想定した、テキストチャットを用いるペアプログラミング支援システムを開発するため、音声対話における行動および会話と学習の成否についての関係性がテキスト対話においても認められるかを検証する実験を行った。その結果、Driver が課題において発生するエラーを解決するために要する時間が長くなるほど、2者のテキスト対話後における Driver のコーディング動作の頻度、および両者によるチャットの頻度が減少することが判明した。また本研究では、この結果を生かしたシステム開発の指針についても言及する。

## Pair programming system based on text chat: A preliminary study on learner-learner interaction

KOSUKE SASAKI<sup>†1</sup> YUGO HAYASHI<sup>†2</sup> TOMOO INOUE<sup>†2</sup>

### 1. はじめに

学習の一形態として、協調学習と呼ばれる学習手法がある。協調学習のうち、特にネットワーク上などの分散した環境にいる学習者間、すなわち地理的に離れた地点にいる複数の学習者間での協調学習を分散協調学習という [1]。この分散協調学習を支援するためには、大きく2種類のアプローチがあり、1つは高度なネットワーク技術および人工知能技術を前提にしたもの、もう1つはコミュニケーションによる学びが発生し得る環境を提供することを主眼とするものである。前者についてはさらに、参加者間の相互作用を重視しその促進を目指す協調的側面を支援する場合と、学習課題に関する参加者の知識獲得の促進を目指す学習的側面を支援する場合の2通りに分別できる [1]。特

に、協調的側面を支援するアプローチを試みた研究としては、協調学習のための適切なグループ構成を行うことにより、参加者間の活発な相互作用を期待する三宮ら [2]、玉置ら [3] の研究や、言語的相互作用状態を同定することを目指した Lund ら [4]、稲葉ら [5] の研究など、複数の研究報告がある。

また、プログラミング学習手法の1つにペアプログラミングがある。コーディングを行う Driver と、Driver の書いたコードをチェックしアドバイス等をする Navigator の役割に分かれ、これを随時交代しながら作業を進めていく手法である。もともとは Beck ら [6] によって提唱された、ソフトウェア開発手法として定式化されたエクストリーム・プログラミングの一種である。しかし、ペアプログラミング自体はプログラムの作成を協調的に作業することができ、また、プログラミングに関する専門知識の共有および拡張という点でも有効という指摘があり [7]、協調学習の一手法として知識の獲得に有効な手法であると考えられる。

このペアプログラミングにおいて、学習がうまくいって

<sup>†1</sup> 筑波大学情報学群情報メディア創成学類  
College of Media Arts, Science and Technology, School of informatics, University of Tsukuba

<sup>†2</sup> 筑波大学図書館情報メディア系  
Faculty of Library, Information and Media Science, University of Tsukuba

いないペアを支援することを目的として、学習者の行動・発話と学習の成否の関係を調査した結果が、平井ら [8] や、Inoue[9] によって報告されている。これらの研究では、学習が失敗する場合には、成功する場合と比べ次の項目が言えると述べられている。

- 発話 1 回の発話長が長い
- 説明を繰り返す回数が多い
- 一方的に発話する回数が多い
- 2 人による会話後のコーディング動作回数の割合が小さい
- 2 人による会話後のコーディング動作頻度が少ない

本研究では、以上の知見を元に、学習者がペアプログラミング学習をより効率的に行えるよう、分散同期環境を想定した支援システムを開発することを目的とした。本システムには Web ベースによるテキストチャットシステムを導入し、また自律型インタフェースエージェントを実装する。学習者のインタラクションパターンに前述の傾向が確認された場合、エージェントが介入を試みることで、ペアプログラミング学習がより効率的に行えるようにしたい。そこで本研究では、対面環境で認められた知見が、テキスト対話においても同様に確認できるかを検証する実験を行った。

## 2. 関連研究

### 2.1 分散協調学習に関する研究

中山ら [10] による研究では、専修学校の授業における演習課題を、遠隔地の学習者 4 人 1 組と教室内の学習者 4 人 1 組のそれぞれが協同で解決し、遠隔地間における分散協調学習と対面環境における集合協調学習の様子を比較した。また、双方の環境とも講師および学習支援者が進捗状況に合わせ、学習者群に介入できるようになっていた。その結果、分散協調学習のほうが学習成績が高かったことが述べられている。また同実験における主観評価の結果から、分散協調学習を行った学習者の方が「講師がよく学習者の状況を把握している」と評価することが判明している。これは分散協調学習の方が、より講師と学習者がコミュニケーションをとっていると感じられることを示している。

一方松田ら [11] は、議事進行モデルを実装した統合的な学習環境において、チュータと呼ばれるエージェントがグループ学習に必要なコミュニケーションを支援するシステムを提案した。このシステムでは、個人作業環境のチュータと協調作業環境の議長役チュータの 2 種類のチュータが存在する。議長役チュータは議論の状態を認識し、議論が行き詰まった際に各チュータに対し、学習者に対する教授を行うよう指示をする。個人作業環境のチュータはこの指示を受け学習者に対し適宜助言等を行い、学習者の理解状態を議長役チュータに報告する。議長役チュータはこの報告から、議論を再開するための発言を行うことができる学

習者を見つけ、その学習者に討論の再開を促すことで行き詰まりの解消を試みる。このように、エージェントが学習者に対して対話の展開や議事進行に関する示唆を与えるインタラクションを試みることで、協調的側面を支援している。

前者の研究は分散協調学習の有効性を、後者の研究は分散協調学習を支援するエージェントの行動方針についてそれぞれ言及している。この双方の研究における実験では、共に議論を進行させるための議長役的な存在があり、この議長役が学習者同士のコミュニケーションをより円滑にする要因の一つと考えられる。

### 2.2 システムが学習者に及ぼす影響

白水ら [12] の研究では、大学生のキャリアカウンセリング、及び大学生同士による物理課題の協同問題解決場面のそれぞれにおいて、ロボットが聞き手として機能しうるかを実験を通じて検討している。どちらの実験場面においても、ロボットは新しい情報を提供するのではなく、実験参加者の発話を再提示することで、参加者の再考を促す支援を行った。特に協同問題解決場面における実験では、議論する学習者を観察しているロボットが、学習者の発言したキーワードを繰り返すリボイス動作を行うことで、学習者が主体的にキーワード、すなわち自ら発した断片的な情報を統合しようとする様子を確認した。またリボイス動作については、学習者の発言をそのまま繰り返す単純リボイスと、学習者の発言の語尾に評価的な文言を付け加えることで、議論の誘導を可能とする誘導リボイスの 2 種に分けて検証を行っていた。結果としては、単純リボイスを行うロボットの方が、誘導リボイスを行うロボットに比べ、学習者による「自分と同等な友達」という認識が強く、また「ロボットと再度学びたい」という気持ちが強く表れることが判明した。また、学習成果の観点からも、単純リボイスを行ったロボットと学習した学習者の方が、自力で知識構成を進める過程が起きやすいことが伺え、転移課題に対しても根拠を含めて正答できる割合が高かった。このことより、協調学習支援の形として、システムが新しい情報を与えるのではなく、学習者の言動をそのままリピーティングすることで学習者に主体的な内省を促す支援を行うという形式も検討すべきであると考えられる。

### 2.3 ペタゴジカル会話エージェントを用いた協調学習支援

エージェントとは、ユーザの問題解決を支援するための代理人として振る舞ってくれるような知的なソフトウェアの総称で、協調学習においてはエージェントが学習者同士のインタラクションに介入することもある。エージェントにおける特性を実現するためには問題解決処理モジュール、知識モジュール等といった応答の方針を決定するモジュールを必要とする [1]。本研究にて行う実験から得られる知

見をこれらのモジュールに機能として実装することで、その知見を生かしたエージェントを構築することができる。

エージェントの振る舞いにも、擬似的な学習者として振る舞う、対象領域の知識を教授しようと振る舞う、協調学習中の議論を支援するよう振る舞う、など複数の種類が存在する。林ら [13] の研究では学習者が協同で行う説明構築活動に注目し、テキスト対話によって行われる学習者同士のコミュニケーションに対し、その会話中に出現したキーワードに反応して、ペタゴジカル会話エージェント (PCA: Pedagogical Conversational Agent) が介入を試みる。このとき、PCA が提示する suggestion prompt には、ポジティブな感情、ニュートラルな感情、ネガティブな感情がそれぞれ含まれる。この研究では、PCA によるポジティブな感情表現を受けた学習者が、他の二種の感情表現を受けた学習者と比べて、質的に良い説明活動を行っていたことが明らかになった。この研究におけるエージェントは、「協調学習中の議論を支援する振る舞い」を行うエージェントであり、本研究が目的とするエージェントの型でもある。

## 2.4 ペアプログラミング学習の成否と学習者の会話行動に着目した研究

平井ら [8], Inoue[9] による研究では、2人1組の学習者にC言語のプログラミング課題を与え、その様子を観察していた。課題は大学の情報系学部で、講義時間の一部を使って実施しており、30分の制限時間が設けられていた。実験時間が短いため、ペア内のDriverとNavigatorの役割は交代させず、その時間内で学習中に発生する「つまずき」の解決の成否によって、「成功事例」と「失敗事例」に分けて分析を行った。その結果、失敗事例のほうが成功事例に比べ、(1) 発話1回の発話長が長い、(2) 説明を繰り返す回数が多い、(3) 一方的に発話する回数が多い、(4) 2人による会話後のコーディング動作回数の割合が小さい、(5) 2人による会話後のコーディング動作頻度が少ない、という結論を得ている。ここでの「つまずき」とは、「コンパイル時にエラーが発生した(コンパイルエラー)あるいはコンパイル時にエラーが発生しなくても実行時にペアが意図した出力がなされなかった(実行時エラー)こと」としていた。このつまずきに着目した検証を行うことで、「協調作業の状態を推定することが可能になれば、よりペアプログラミング学習などの効果が得られるようになるであろう」と述べている。

本研究で開発を進めるシステムでは、これらの研究で行われている実験のように学習者は音声で対話を行うのではなく、テキストでの対話を行う。学習者の対話方法をテキスト対話と限定したのは、エージェントが学習者への介入を試みるために音声対話を認識する場合、その認識精度からシステム化が難しくなるためである。従って、システムの開発を進める上でテキストベースで行われる協調活

次の処理条件を満たすWebページを作成してください。  
ただし、**太字・下線付き**で書かれている内容は指示ですので、ページ上に表示させてはいけません。

【処理条件】

- HTMLファイルの名前は"hera.html"とします。
- ページタイトルは「ヘラブリナ釣り大会」としてください。
- 背景画像は"back.gif"とします。
- ページの最上段に「ヘラブリナ釣り大会」と表示してください。  
**フォントサイズは7、フォントカラーは緑色**を指定してください。
- 4の下に、画像"line.gif"を、**中央揃え**で挿入してください。
- 5の下に、「釣りの好きな人、経験のない人も歓迎します!」と表示してください。  
**フォントサイズは5、フォントカラーは赤色、配置は中央揃え**を指定してください。
- 6の下に、下記の表を表示してください。また、**枠線も表示**してください。

開催内容 ( <b>フォントサイズ6、太字</b> ) ( <b>背景は水色</b> )	6月15日 午前9時より ( <b>背景は水色</b> )	( <b>画像"pic1.jpg"を挿入</b> )
場所 ( <b>フォントサイズ6、太字</b> ) ( <b>中央揃え、背景は白色</b> )	周辺地図 ( <b>背景は白色</b> ) ( <b>8.の条件を守る</b> )	( <b>画像"pic2.jpg"を挿入</b> )

7. 6.の表中の、2行2列目にある「周辺地図」には、同じフォルダ中の"page.html"に繋がるリンクを挿入してください。

図 1 実験課題の一例

動の分析が必要である。以上の理由より、本研究では平井ら [8], Inoue[9] による研究を元に、実験の組み立て、および得られた結果の分析を行った。

## 3. 実験

### 3.1 手順

実験では2人1組の参加者を募り、遠隔環境を想定した実験室にてペアプログラミング課題に取り組んでもらった。ペアプログラミング課題はHTMLおよびCSSを用いた1ページ程度のWebページを作成するもので、その課題例を図1に示す。課題は、日本情報処理検定協会によるホームページ作成検定試験 [14] (HP試験)の問題を元に、実験用に問題文の表記を変えたものを紙面にて提示した。

参加者には実験開始前に予めプレテスト及びプレアンケートを実施した。これは、学習者のHTMLおよびCSSに関する事前知識と経験を問うものであり、この結果によって、課題で使用される問題の難易度、すなわち参照元のHP試験問題の級を変えた。これは、学習者にとって課題が容易すぎると、本研究において着目すべき「エラー」の事象が確認しにくくなってしまいうためである。

参加者は、日常会話程度の日本語が扱え、HTMLタグを用いてWebページを作成できる大学生6組12名(平均年齢21.17歳の男性8名、女性4名)であった。そのすべてが友人同士であり、男女の組み合わせには制限を加えていない。実験中は互いの姿を視認できないよう、両者の間にパーティションを挟むことで、擬似的な遠隔環境を作った(図2,3を参照)。Driverの使用するPCは、OSにWindows 8を搭載したノートPCで、画面サイズは15.6型(1366x768dpi)である。またWebページのコーディングにはWindows標準のメモ帳を使用し、ブラウザでテスト表示する際にはInternet Explorer (Ver. 10.0)を使用してもらった。また、DriverがPCを操作する様子は26型(1920x1200dpi)のミラーリングディスプレイを通じて、

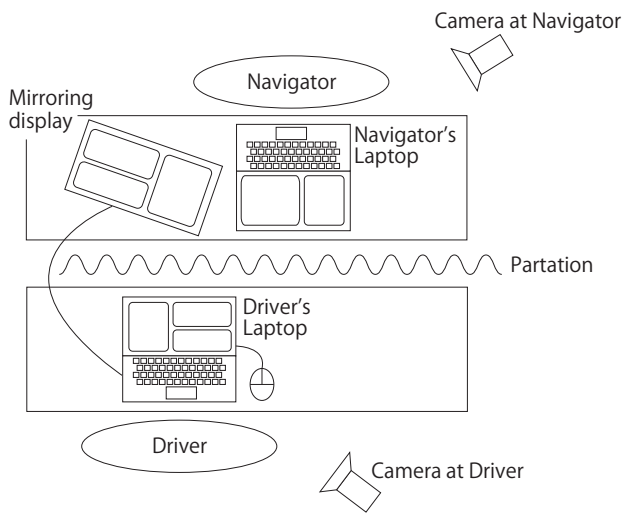


図 2 実験環境の模式図



図 3 実験の様子 (左は Driver, 右は Navigator の様子)

Navigator 側も確認できるようになっており、この画面を見ながら Navigator は Driver に指示を送るなどを行ってもらった。なお Navigator には、持参してもらった個人の PC を使用してもらった。

また、互いは声を出して行う音声対話を行うことを禁止した代わりに、テキストチャットシステムを用いて対話を行うことを許可した。チャットシステムの様子は図 4 に示す。このチャットシステムは次の要件を満たすように実験的に構築したものである。

- 同期型のテキスト対話ができるよう、リアルタイムで動作するチャットシステムが構築されていること
- 分散型であるために、ネットワークに繋がればどこでもこのシステムを利用できるよう、Web ベースなシステムであること
- 後のデータ収集のために、学習者のデータが一意に区別できるようセッション管理等を用いたアカウントシステムが実装されていること

加えて、実験中は平井ら [8], Inoue[9] による研究に則り、参考文献の閲覧を許可した。この参考文献とは、HTML の Web リファレンスの 1 つである HTML クイックリファレンス [15] を、予めローカル環境に保存したものである。

なお、ペアプログラミングは本来、Driver と Navigator は随時役割を交代していくものだが、平井ら [8], Inoue[9] の研究で行われた実験ではペア内で役割を交代しなかった

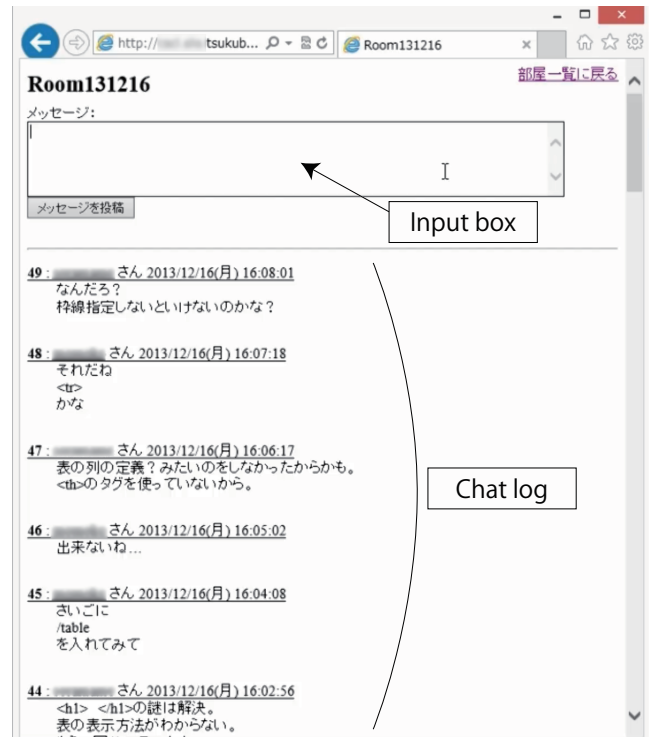


図 4 実験で使用したチャットシステムの様子

ことから、本実験においてもこの役割交代はしないこととした。

その他、実験における学習者への主な指示は以下の通りである。

- テキストチャットシステムの利用、および参考文献の閲覧は、いつどのタイミングで行っても構わないこと。
- テキストチャットシステムの利用、HTML ファイルの確認、および参考文献の閲覧用途以外でのインターネットブラウザの使用は禁止であること。
- 課題に指定されている条件をすべて満たしているか、必ず 2 人で十分に確認すること。
- 課題に関する質問を受け付けることはできないこと。
- 課題はできるだけ早く終了させるよう心がけること。

この実験では、学習者 2 名の各後方から 1 台ずつカメラによる撮影を行い、また Driver が使用する PC にはキャプチャソフトを導入し、合計 3 種類の映像を撮影した。ただし、実験を行った 6 組の中で、1 組で撮影上の不手際があったため、結果として残ったのは 5 組分の映像である。また、チャットの内容はサーバ上にログが残っているため、発言内容はこのログを参照した。

### 3.2 データ収集

収集した映像のエラー区間に注目すべく、まずはこのエラー区間の抜き出しを行った。本研究では、このエラーの定義を「スペルミス、文法ミスなどにより、想定していたものとは別の結果がブラウザ上に表示されたこと」とした。また、エラー区間の定義を「エラーを含むコードをブラウ

表 1 各エラーの内容

ペア	error_ID	data length	エラーの内容
A	1	95.0 sec	">"のつけ忘れによるHTML文法エラー
	2	124.8 sec	CSSのセレクタ指定ミスによるレイアウトエラー
B	3	52.3 sec	CSSのID指定ミスによるレイアウトエラー
	4	1274.8 sec	CSSのプロパティ指定ミスによる表示エラー
	5	50.5 sec	指定文字の入力ミス
C	6	484.9 sec	tableタグの文法ミスによる表示エラー
	7	919.2 sec	画像リンク指定ミスによる表示エラー
	8	669.9 sec	span内にalign属性をネスト/CSS内の"/忘れ文法エラー
	9	310.2 sec	改行タグのつけ忘れによるレイアウトエラー
	10	294.7 sec	CSSのセレクタ指定ミスによる表示エラー
D	11	100.8 sec	centerタグ忘れによるレイアウトエラー
	12	334.8 sec	指定文字の入力ミス
	13	3545.3 sec	CSSのセレクタ指定ミスによる表示エラー
E	14	233.3 sec	画像リンク指定ミスによる表示エラー
	15	4311.3 sec	CSSの記述ミスによる表示エラー

ザ上に表示させた瞬間から、当該エラーを修正したコードをブラウザ上に表示させた瞬間」までとした。

撮影した映像の中で、エラーが発生した区間は 15 区間である。各エラーの詳細を表 1 に示す。表中の error\_ID はエラー区間の通し番号であり、エラーは作業中に複数発生したため、1 組のペアに対して複数のエラーが存在する。data length はそれぞれのエラー区間の長さを秒単位で表したものである。エラーの内容については、実験者が撮影した映像を観察することで判明したエラーの内容であり、学習者が実験中に気がつかなかった内容も含む。このうち、error\_ID が 10, 13, 15 のものに関しては、撮影に用いたビデオカメラの容量切れにより、エラーの修正を最後まで確認できなかったものである。

また、それぞれの区間における学習者のコミュニケーションデータを表 2 に示す。このデータは、収集した映像を、動画分析ソフトウェア ELAN[16] を用いて、ペアごとに Driver, および Navigator の行動、発話を記録したものである。

## 4. 結果

平井ら [8], Inoue[9] の研究では、2.4 節で述べた通り、つまずきの修正の成否で分析対象を分別し分析を行っていた。この成否とは、制限時間内に修正が完了したかどうかによるものである。しかし、本研究では課題解決のための制限時間は設けていない。そのため、これらの研究と同様の比較手法を用いることはできない。そこで、収集したデータから表 2 に示される値の算出を行い、「エラー区間の長さ (data length)」との相関を検証するための相関分析を実行した。これにより、修正の成否とではなく、修正にかかる時間との関係を導き出すことができ、エラーの修正にかかる時間が長くなるほどどのような傾向が見られるかを調査することができる。

比較の対象としたのは、撮影が最後まで完了できなかった error\_ID が 10, 13, 15 のものを除く 12 のエラー区間に

おける、data length と次の 8 項目である。これは、平井ら [8], Inoue[9] の研究で比較されていた内容と同様のものである。

- 発言頻度 (chat frequency)  
 data length に対する発言数の割合を表す。単位は回/分。
- 平均発言文字数 (average speak length)  
 チャットの発言 1 回あたりの文字数の平均値を表す。単位は文字/回。
- 連続発言率 (uninterrupted speak raito)  
 総発言数に対する、連続で行われた発言の回数の割合を表す。単位は%。
- コーディング動作率 (operation raito)  
 data length に対するコーディング動作の総継続時間の割合を表す。単位は%。
- コーディング動作頻度 (operation frequency)  
 data length に対するコーディング動作回数の割合を表す。単位は回/分。
- 会話後のコーディング動作率 (operation raito after dialogue)  
 全コーディング動作回数に対する、会話後に行われたコーディング動作回数の割合を表す。単位は%。
- 会話後のコーディング動作頻度 (operation frequency after dialogue)  
 data length に対する、会話後に行われたコーディング動作回数の割合を表す。単位は回/分。
- 平均コーディング継続時間 (average operation length)  
 コーディング動作 1 回あたりの平均継続時間を表す。単位は秒/回。

また、相関分析と合わせ、因果関係を調査するため単回帰分析も行った。その結果、以下の 2 項目について相関および因果関係が確認できた。

### 4.1 会話後のコーディング動作頻度

data length と operation frequency after dialogue の 2 者における相関分析の結果、負の相関関係が認められた ( $r = -0.662$ ,  $p = 0.019$ )。加えて、単回帰分析の結果は、 $y$  を data length,  $x$  を operation frequency after dialogue とおいた回帰直線式が  $y = 731.538 - 661.901x$  となり、5%の有意水準のもと有意な回帰直線と判断できた ( $F = 7.821$ ,  $p = 0.019$ )。この 2 者を散布図に表し、回帰直線を重ねて描画したものが図 5 である。これより、会話後に行われるコーディング動作の頻度が低くなるほど、エラーを修正するのに要する時間が長くなる事が分かる。

### 4.2 会話後のコーディング動作頻度

data length と chat frequency の 2 者における相関分析の結果、負の相関関係が認められた ( $r = -0.601$ ,  $p = 0.039$ )。

表 2 各ペアのコミュニケーションデータ

ペア error_ID		ペア A		ペアB			ペアC				ペアD			ペアE		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
data length	[sec]	95.0	124.8	52.3	1274.8	50.5	484.9	919.2	669.9	310.2	294.7	100.8	334.8	3545.3	233.3	4311.3
chat frequency	[times/min]	1.9	1.2	1.7	0.3	4.2	0.6	0.5	0.4	0.5	0.7	1.2	0.6	0.5	1.4	0.7
average speak length	[characters/time]	15.2	75.2	16.3	13.6	4.4	19.6	29.9	48.7	21.4	15.9	9.8	8.6	14.1	20.5	28.2
uninterrupted speak ratio	[%]	40.0	25.0	0.0	30.8	16.7	25.0	28.6	12.5	0.0	50.0	33.3	33.3	41.8	30.0	56.0
operation ratio	[%]	54.4	42.4	78.2	46.9	34.7	48.7	55.9	52.3	79.4	43.1	81.2	34.9	55.0	57.0	57.5
operation frequency	[times/min]	1.3	1.4	2.3	1.2	4.8	0.9	0.8	1.1	0.6	0.6	1.2	0.7	0.6	1.5	1.0
operation ratio after dialogue	[%]	50.0	66.7	50.0	19.2	25.0	14.3	33.3	8.3	66.7	66.7	50.0	50.0	35.1	16.7	24.6
operation frequency after dialogue	[times/min]	0.6	1.0	1.1	0.2	1.2	0.1	0.3	0.1	0.4	0.4	0.6	0.4	0.2	0.3	0.2
average operation length	[sec/time]	25.9	17.6	20.5	23.0	4.4	33.7	42.8	29.2	82.1	42.3	40.9	29.2	52.7	22.2	35.9

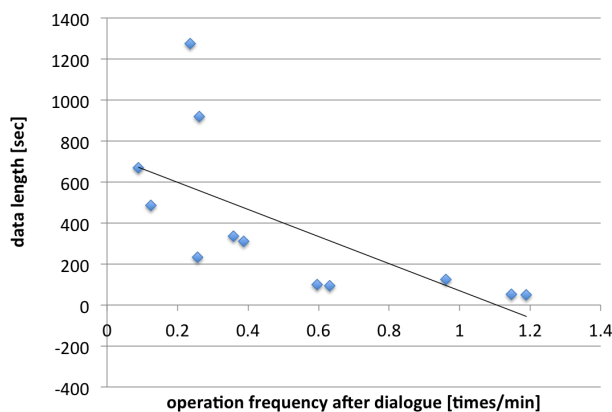


図 5 data length と operation frequency after dialogue の散布図

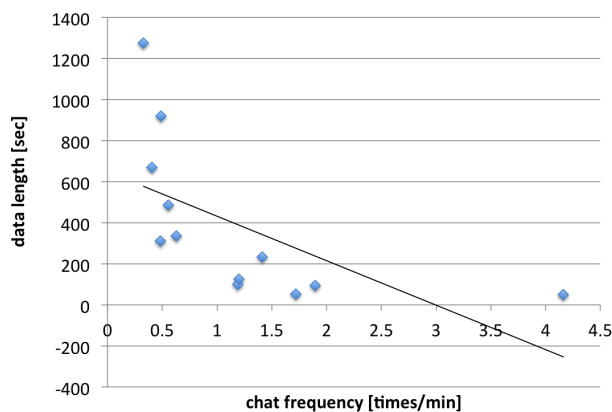


図 6 data length と chat frequency の散布図

加えて、単回帰分析の結果は、 $y$  を data length,  $x$  を chat frequency とおいた回帰直線式が  $y = 649.098 - 216.866x$  となり、5%の有意水準のもと有意な回帰直線と判断できた ( $F = 5.665, p = 0.039$ )。この2者を散布図に表し、回帰直線を重ねて描画したものが図6である。これより、チャット対話の頻度が低くなるほど、エラーを修正するのに要する時間が長くなること分かる。

## 5. 考察

本研究では、ペアプログラミングにおける学習者同士のインタラクションパターンを定量的に分析することで、エラー修正に要する時間と相関のあるインタラクションパターンを確認した。本節では、得られた結果について検討

を行い、最終的に目指すエージェントシステムの開発方針の示唆を論じる。

### 5.1 音声対話とテキスト対話の差異

1節で述べたように、平井ら [8], Inoue[9] の研究では5つの学習者の行動、すなわち (1) 発話1回の発話長, (2) 説明を繰り返す回数, (3) 一方的に発話する回数, (4) 2人による会話後のコーディング動作回数の割合, (5) 2人による会話後のコーディング動作頻度, のそれぞれが学習の成否と関係性があると述べられている。この5項目のうち、本研究で明らかとなったものは「2人による会話後のコーディング動頻度が少ない」のみである。特に、平井らの研究における学習者同士の「会話」に関する知見に関しては、今回の実験において同様の結果を確認することはできなかった。ここでは、この音声対話とテキスト対話の差異に注目して、同様の結果を確認できなかった理由を検討したい。

今回の実験を通してよく見られた光景の1つに、「Navigatorが参考文献の文章をコピーして、そのままチャットで同じ内容を発言した」ことが上げられる。例えば、Driverの手助けとなりそうなサンプルソースコードを見つけたNavigatorは、参考文献から該当部分をコピーし、それをそのままチャット欄に貼り付けて発言を行った。通常時に発言する文量とは大きく異なる場合もあり、通常は数十文字程度で行われている対話も、このコピーアンドペーストを行った発言だけ数百文字まで文字数が増えることもあった。平井らの研究では、「発話1回の発話長が長い」と学習は失敗に終わりやすいと述べられていたが、この例における文字数の増加は、そのままDriverへの有用な情報量の増加にも繋がるため、必ずしも平井らの研究とは同様の知見が表れないものと考えられる。

このように、テキスト対話と音声対話では、その性質の違いから、必ずしも関連研究と同様の知見を得られるわけではなさそうである。例えば、平井らの研究では、その考察の中で「Navigatorが連続して発話長の長い発話をしてしまったために、Driverが混乱をおこしてしまい、Navigatorに対して何もいえなくなったと考えられる」[8]と述べているが、テキスト対話では、発言を行うために一度チャットシステムに文字を入力する必要があり、この操作によっ

表 3 Navigator が Driver にアドバイスをする行動の一例 (error\_ID=10 における行動の一部を抜粋)

発言番号	発言主	発言内容
98	Driver	画像が貼れません… 自分の html と css みてもらえますか？
99	Navigator	うん！ わかった
100	Navigator	sample を rink にしてみたらどうか…？
Driver による operation		
101	Navigator	出来ないね…

て生じる発言の“間”によって矢継ぎ早に情報を流し込まれる状況が少ないほか、万が一相手の発言に追いつけなくなっても、チャットのログを見返すことができる。

以上の理由から、発話に関する知見はテキスト対話では確認することができなかつたと考えられる。

## 5.2 Navigator の発言に関する検討

実験中、エラーが発生している間に、Navigator が Driver に対し具体的な修正操作を要求するなどのアドバイスをしている場面がいくつか見られた。例えば、表 3 に示すような場面である。発言番号 98 において、Driver は自分の操作が行き詰まったことを Navigator に知らせている。その後、発言番号 100 にて具体的な指示を出した。この後、Driver は Navigator の指示通りの操作を行った。このように、Navigator が何らかの指示を含むアドバイスを発言した場合、Driver はその通りに従うことが多かった。

一方で、Driver が Navigator からの指示とは別の操作を行う場面も見受けられた。それが表 4 に示す場面である。発言番号 80 にて、Navigator が HTML のタグを記述する位置を変えるよう指示を行っている。しかしその後の Driver は、その指示に関する操作を行わず、別の HTML タグの修正を行った。結果、この事例に関してはエラーの修正まではたどり着けなかつたが、状況に進展が見られた。

このように、Driver が Navigator の指示とは違う操作を行う理由としては、「Navigator のアドバイスを無視した」場合と「Navigator のアドバイスをヒントに、Driver がエラーを修正するための別の良案が浮かんだ」場合の 2 種類が考えられる。表 4 の例がどちらの理由にあたるのかは判断し難いが、後者の可能性がある以上、Navigator の発言は、例えどのようなものでも Driver の動作に対し相互作用を引き起こしうることになる。今回の実験では、このような例は一度しか見受けられなかつたため検証はできなかつたが、Navigator の発言がどれほどペアプログラミング学習の効率に影響を与えるかは、今後の課題となる。もし Navigator の発言が学習に対し有用に働くのであれば、システム内のエージェントは Navigator に対しより積極的に発言を促すよう介入を試みる必要がある。

表 4 Driver が Navigator の指示とは違う操作を行う行動の一例 (error\_ID=13 における行動の一部を抜粋)

発言番号	発言主	発言内容
80	Navigator	title 閉じる場所変えてごまかしたりとかできないかな
81	Navigator	間違えた
Driver による operation		
82	Navigator	何が起こった？
83	Driver	<center>をやめてみた

## 5.3 data length と chat frequency の相関について

一般にこのような実験では、参加者の拘束時間が長引くほど実験参加者の実験に対する意欲が失われる可能性がある。今回の実験において、data length はこの実験の拘束時間を引き伸ばす要因となった。従って、data length が長引くほど拘束時間が伸び、それに従って参加者の意欲が削がれ、チャットでの発言頻度が下がった、とも考えられる。つまり、今回得られた data length と chat frequency の相関関係は、最初から予想できるものであった。しかし、予想通りであったとしてもこのように結果として出てきたことによって、data length を長引かないようにエージェントが学習者を支援する必要性を改めて確認することができたとも言える。

## 5.4 エージェントの開発に向けて

ペアプログラミング学習を効率化するためには、エージェントは学習者同士の悪いインタラクションパターンを発見し、これを打開するよう介入を試みる必要がある。悪いインタラクションパターンとは、すなわち 4 節で得ることのできた、(1) 会話後のコーディング動作頻度の低下、(2) テキスト対話の頻度の低下、のような状況である。これらとは逆の状況を作るようにエージェントが学習者に対し介入を試みることができれば、学習の効率が上がり、学習を支援できると考える。つまり会話後のコーディング動作がより積極的になるよう、またテキスト対話がより活発になるようエージェントが学習者に対し促すことで、ペアプログラミング学習支援を行うことができると期待できる。

加えてエージェントを開発する上で、本研究で得られた知見以外にも検討すべき内容として、エージェントの基本的な振る舞いや介入手段についても、エージェントの開発方針として一考するべきである。基本的な振る舞いやとしては、中山ら [10]、松田ら [11] の研究から、エージェントは議長的な役を受け持ち、議論全体の動向および学習者個々人の理解状況の認識に努める必要があると考えられる。このような振る舞いを行うエージェントを開発するためには、個々人の状況まで正確に把握するためのマルチエージェントシステムを構築する必要があるだろう。

介入手段としては、本研究で得られた知見以外にも様々

な介入が考えられるが、白水ら [12] の研究に則れば、学習者の発言をリボイスするような機能もエージェントには必要と考えられる。課題解決中にチャット内で特定のキーワードが出現した場合、その発言をリボイスすることで、学習者に再考の余地を与え、自ら発した発言の整理を行うことで、より理解の深まった学習効果を得ることが期待できる。また林ら [13] の研究で報告された通り、エージェントによる発言の仕方にも注意を払わねばならない。エージェントが発言を行う際は、ポジティブな感情表現を含むことで、学習者のエージェントに対して好意的な印象を持たせ、協調学習が意欲的に取り組まれるようにならなければならない。ただし林らの研究におけるエージェントは、白水らの研究によるところの「誘導リボイス」を行うエージェントに分類される。本研究の想定する学習課題であるペアプログラミングでは、エージェントは「誘導リボイス」を行うべきなのか「単純リボイス」を行うべきなのかは、別途実験を行うことで検証する必要がある。

また介入手段としては他に、林ら [17] が音声ナレーションによるエージェントの介入について検討を行っている。この研究では、テキストで課題を提示されテキストで対話を行う学習環境では、学習者の視覚認知に負担を掛けすぎているため、エージェントは視覚によって認知される介入ではなく、聴覚で認知される介入を行う方が学習者のパフォーマンスを促進するという結果を上げている。この結果より、本研究におけるエージェントの介入手段は、学習者同様チャット上にテキストを残すのではなく、音声等の視覚ではない別の感覚によって認知される介入を試みるべきだろう。

本研究で行った実験は、学習者同士のインタラクションを観察するための実験であり、学習者は外部からの介入を受けていない。今後エージェントの開発を行うために必要な実験として、まずは学習者同士のインタラクションにチュータ役の実験者が介入を試みることで、学習パフォーマンスの向上を検証する実験を進めたい。実験者はエージェントの機能として検討すべき振る舞いや介入手段などを網羅的に試みることで、どのような介入を行えば学習者がより効率的な学習を行えるかを検証する。その後、この実験で有効と確認できた介入手法をモジュールに組み込むことでエージェントを実装する。このエージェントがチュータ役として学習者に介入を試みる同様の実験を進めることで、最終的にエージェントが行うべき介入手法について検討する。これらの検証実験は今後の課題である。

## 6. まとめ

本研究では、ペアプログラミング学習を支援するシステムを開発する目的のもと、関連研究で得られた知見が本システムが想定する環境下でも同様に確認できるかを検証する実験を行った。その結果、会話後のコーディング動作頻

度が低いほど、またテキスト対話の頻度が低いほど、学習中に発生したエラーを修正するのに要する時間が長くなることが判明した。今後は、エージェントがこのようなインタラクションパターンを発見した際、この状況を打開するよう学習者に介入を行うことで、学習をより効率的に進めることができるシステムの開発を進めていく。

## 参考文献

- [1] 日本教育工学会編：教育工学辞典，実教出版（2000）。
- [2] 三宮毅，松浦健二，緒方広明，矢野米雄：Sharlok: Awareness を指向した開放型グループ学習支援システム，電子情報通信学会技術研究報告. ET, 教育工学, Vol.95, No.604, pp.109-116 (1996)。
- [3] 玉置亮太，ヘラルドアヤラ，矢野米雄：日本語協調学習環境 GRACILE：ソフトウェアエージェントのコミュニケーション，電子情報通信学会技術研究報告. ET, 教育工学, Vol.95, No.604, pp.101-108 (1996)。
- [4] Kristine Lund, Michael Baker, Monique Baron: Modelling dialogue and beliefs as a basis for generating guidance in a CSCL environment, *Intelligent Tutoring Systems Third International Conference*, pp.206-214 (1996)。
- [5] 稲葉晶子，岡本敏雄：分散協調学習支援システムのための Negotiation モデルの検討，電子情報通信学会技術研究報告. AI, 人工知能と知識処理, Vol.96, No.345, pp.9-14 (1996)。
- [6] Beck, K.: *Extreme Programming Explained: Embrace Change*, Reading, PA, Addison-Wesley (1999)。
- [7] Wray, S.: How Pair Programming Really Works, *IEEE Software*, Jan./Feb. 2010, pp.50-55 (2010)。
- [8] 平井佑樹，井上智雄：ペアプログラミング学習における状態の推定—つまずき解決の成功と失敗に見られる会話の違い—，情報処理学会論文誌, Vol.53, No.1, pp.72-80 (2012)。
- [9] Tomoo Inoue: Investigating the Relation between Behavior and Result in Pair Programming: Talk and Work Leads to Success, *The Journal of Information and Systems in Education* (in press)。
- [10] 中山実，竹内義良，関信仁，加藤真一，今長豊，清水康敬：分散協調学習と集合協調学習による学習成績の違いに関する検討，日本教育工学会論文誌, Vol.30, pp.153-156 (2006)。
- [11] 松田昇，榎場泰孝，佐々木宏，岡本俊雄：ネットワーク環境における分散協調型知的マルチメディア CAI の研究，電子情報通信学会技術研究報告. ET, 教育工学, Vol.94, No.59, pp.53-60 (1994)。
- [12] 白水始，中原敦：人の主体的な問題解決を促すロボットの役割，日本ロボット学会誌, Vol.29, No.10, pp.898-901 (2011)。
- [13] 林勇吾，小川均：Pedagogical Conversational Agent を用いた協同学習の促進—感情表出に着目した検討—，電子情報通信学会論文誌 D, Vol.J96-D, No.1, pp.70-80 (2013)。
- [14] 日本情報処理検定協会：ホームページ作成検定試験，[http://www.goukaku.ne.jp/test\\_home.html](http://www.goukaku.ne.jp/test_home.html) (2013.12)
- [15] HTML クイックリファレンス，<http://www.htmq.com/> (2013.12)
- [16] Tools — Language and Cognition — Max Planck Institute for Psycholinguistics, <http://www.mpi.nl/departments/language-and-cognition/tools/tools> (2014.2)
- [17] 林勇吾：Dual-Channel Moodel に基づく対話エージェントを利用した協同による説明活動の促進方法，電子情報通信学会論文誌 D, Vol.J97-D, No.1, pp.17-27 (2014)。