

プログラム理解の記録・再利用のための 統合ソフトウェアシステム

須山 哲宏 寺田 実

東京大学大学院工学系研究科

プログラムの理解や学習を助けるために、多人数で読む「読み会」を開くのが効果的である。従来の読み会の場では、プログラムを紙に印刷したり、PCの既存のエディタを利用したりして読んでいたが、我々は先行研究で読み会用グループウェアを作り、PCを利用して読み会を支援することを可能にした。しかし、先行研究にはユーザインタフェースが不完全で使いにくいという問題と、読み会の記録が残らず復習に利用できないという問題があった。本稿ではユーザインタフェースを向上させた読み会支援システムを紹介し、チャットを併用して記録を残すための読み会環境の構築方法と、記録を付箋紙に書いてソースコードに直接貼り、簡単な方法で表示/非表示を切り替える方法を提案する。

Integrated Software System to Record and Reuse Program Comprehension

Norihiro SUYAMA and Minoru TERADA

Graduate School of Engineering, The University of Tokyo

Code walkthroughs (reading programs together with others) are effective for learning and comprehending code. We proposed a groupware system to support walkthroughs, which improved the environment for code walkthroughs, but had some problems: user interface is not so good, and result of the comprehension is insufficient for the later review. In this paper, we propose an integrated system for walkthroughs. The advantages of this system are: (1)improved user interface, (2)using chat system for preserving much dialogs about the program, (3)easy to control the visibility of the overlayed chat logs on the source code. Experiment shows the usefulness of the system.

1 はじめに

プログラムの内容を理解するためには、プログラム読解が必要不可欠である。プログラム読解の方法としては一人でソースを読む方法が一般的だが、複数人でソースを読むプログラム読み会による読解方法も存在する。プログラム読み会¹は教育機関を始めとした様々な機関で重要視されており、プログラムの構造を把握し理解できるほか、プログラム言語やアルゴリズムの習得が可能になると

いうプログラムを読む意味と、複数人が知恵を絞り合うことによって、バグ修正が容易になり、技術の伝授が可能になるという、共同作業を行う意味を併せ持っている。

我々の研究室でも以前から数万行のプログラムを20～30人で読む読み会を実施していたが、この規模の読み会に適した読み会環境が存在していなかったため、先行研究としてクライアントサーバシステムを利用した読み会支援システム Pierre を作った [1]。

Pierre は読み会の進行をサポートする様々な機

¹ソース読書会と呼んでいる機関もある。

能を持つ。ソースコードを読むのに必要な機能として、ファイル移動機能と、ソースコード中の検索したい文字列をクリックするだけで、関数やマクロ定義へジャンプする機能、及びボタン一つで呼び出し元へ復帰する機能を持っている。

また、読み会を運営するために必要な機能として、全参加者が操作可能な共有指示棒による視点の共有機能、司会者になって指示棒を追従させる (Master) / 指示棒に追従する (Slave) / 指示棒から独立する (Independent) をボタン一つで切り替える機能、共有シェルによるコマンド実行及びその実行結果の共有機能、記述する場所を選ばない書き込み機能などを提供している (図 1 参照)。

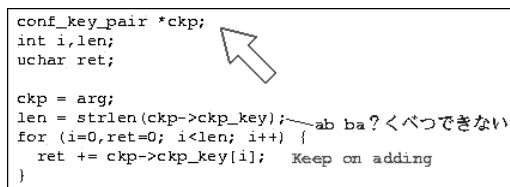


図 1: 共有指示棒と書き込み

中央の矢印が共有指示棒である。右下に書かれているのが書き込みであり、紙に書くように、場所を選ぶことなく書ける。

Pierre は従来の読み会の環境を向上させることに貢献したものの、書き込み機能が有効利用されることは殆どなかった。Pierre の書き込み機能は、ソースコードの画面の余白にメモを書くと、自動的に全員に共有化されるものであったが、メモ書きであるために対話的な書き方ができず、共有化されてしまうために確実に理解できたことしか書けず、そのため読み会の記録が十分に残らず、復習には全く役に立たないという問題点があった。

そこで本研究では文字によるオンライン会話システムであるチャットを利用することによって、記録が残らないという問題点の克服を試みる。手に入れた会話記録は付箋紙に書き込んでソースコード上に直接貼り、その表示 / 非表示を簡単に切り替える方法を提案する。その上で、得られた記録は復習にも有効であることを評価実験を通じて示す。

2 システム紹介

読み会システム Pierre に対し、チャットの記録を再利用するための機能を付け加えた新システム

の実行画面は図 2 である (チャットを利用するため参加者はこの他何らかのチャットクライアントを用意している)。以下にその新機能を紹介する。

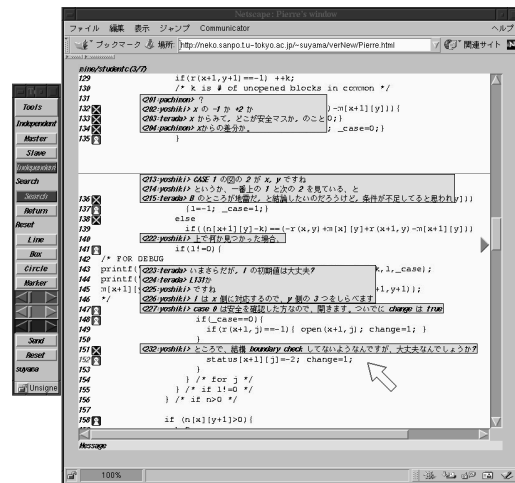


図 2: 新システムの実行画面

2.1 チャットの自動添付

チャット記録とソースコードの対応付けは、読み会が順調に進行している時には問題とはならないが、途中で以前の箇所を読み返したとき、或は復習に使おうとして読み返すときに、非常に厄介な問題になる。Pierre は行番号順に並んでいるが、チャットは時間順に並んでいるためである。

そこで、読み返す時はソースコード中心になりやすいことと、読み会では共有指示棒が示す行について議論されること (図 3) を踏まえて、チャットの発話は付箋紙へ書き込まれた後、自動的に共有指示棒の指す直前の行に添付され、同時に付箋紙の存在と状態を示すアイコンが行番号の右に生成される仕組みを作った (図 4)。この付箋紙は首藤らの研究 [2] による付箋紙を応用したものである。発話の追加があると、付箋紙は上方へ拡大して、最新の発話が最下段に挿入される (図 5)。

2.2 付箋紙の表示 / 非表示の制御

付箋紙をソースコード上に貼り、発話が追加される度に拡大させるだけでは、付箋紙同士が重なるばかりか、ソースコードを簡単に覆い隠してしまうことになる。そこで、付箋紙にはアイコンをクリックすることにより、表示 / 非表示を個別に切り替えられる機能が付いている (図 6)。

```

67
68 open(x0,y0);
69 while(1){ /* Continue BASIC OP
70 change=0;
71 for(x=0;x<8;x++){
72 for(y=0;y<8;y++){
73 m[x][y]=0;
74 n[x][y]=0;
75 if(status[x][y]>0){
76 for(i=x-1;i<=x+1;i++){

```

図 3: 共有指示棒の指す行をチャットで議論する。

```

67
68 open(x0,y0);
69 while(1){ /* Continue BASIC OP
70 change=0;
71 for(x=0;x<8;x++){
72 f<124:terada> 盤面を全スキャンして
73 m[x][y]=0;
74 n[x][y]=0;
75 if(status[x][y]>0){
76 for(i=x-1;i<=x+1;i++){

```

図 4: チャットへの発言は付箋紙に記録されて共有指示棒の指す直前の行に貼られる。行番号の右に付箋紙の存在と状態を示すアイコンが生成される。

```

67
68 open(x0,y0);
69 while(1){ /* Continue BASIC OP
70 chat<125:yoshiki> いろいろ処理をやるよ
71 for(x=0;x<8;x++){
72 f<126:yoshiki> 最初は、
73 m[x][y]=0;
74 n[x][y]=0;
75 if(status[x][y]>0){
76 for(i=x-1;i<=x+1;i++){

```

図 5: チャットへ発言が追加されると付箋紙は上に伸び、新しい発言ほど下にくるように追加される。

```

67
68 open(x0,y0);
69 while(1){ /* Continue BASIC OP
70 change=0;
71 for(x=0;x<8;x++){
72 for(y=0;y<8;y++){
73 m[x][y]=0;
74 n[x][y]=0;
75 if(status[x][y]>0){
76 for(i=x-1;i<=x+1;i++){

```

図 6: アイコンをクリックすると付箋紙を閉じることができる。

しかし、読み会中は付箋紙で隠されてしまうソースコードを見るために、頻繁に付箋紙の表示制御を行いたいの、アイコンのクリックによらずに極力簡単な方法で表示 / 非表示の切り替えができる以下の 2 つのモードを用意した。

- ここまでモード (図 7 参照)
- ここだけモード (図 8 参照)

図 7: ここまでモード

ページ上端から共有指示棒 (またはマウス) の指す行までの付箋紙が表示される。

図 8: ここだけモード

共有指示棒 (またはマウス) の指す行の付箋紙だけが表示される。

これらのモードでは、マウスを上下移動させるだけで、表示したい位置まで / だけの付箋紙を表示することができる。マウスがウィンドウ外にある時は、共有指示棒に合わせてその位置まで / だけの付箋紙が表示されることになる。

2.3 チャットによる共有指示棒の操作

チャット読み会中は進行役が書記係を兼任することになるため、共有指示棒を動かす操作とチャット

トへの書き込みをするために、新システムとチャットクライアントのウィンドウ間を頻繁に往復することになり、面倒である。そこでチャットクライアントから特定の文字列を入力すれば、コマンドと認識して指示棒を操作できる機能を付けた。

3 評価

3.1 チャットを引き出す読み会スタイル

チャットを使えば対話的に書けるので書きやすくなるが、実際に十分な会話記録を残すためには読み会環境の提供方法について考察する必要がある。

そこで表 1 の条件を設定してチャット読み会を行った。ここで「音声」は声を出して話すことを意味する。参加者全員が新システムとチャットクライアントを用意した。

表 1: チャット読み会の条件

方式	空間	音声	記録者
個人	同室対面型	あり	書きたい人
書記	同室対面型	あり	書記係中心
進行	遠隔分散型	禁止	進行役中心

3.1.1 読み会の結果

結果を表 2 に表す。書記係のいない個人方式では全く書き込みがなく、書記係のいる書記方式でも十分な書き込みが得られなかった。しかし、音声を使わない進行方式では十分な書き込みを得ることが分かった。その後の実験で 2 つ以上のグループに分かれて別々の部屋で遠隔チャット読み会を行えば、部屋単位で音声を使って議論しても、十分な書き込みが得られることが分かった。

表 2: チャット読み会の時間と書き込み行数

方式	時間	行数	行数/時
個人	約 120 分	0 行	0 行
書記 (1)	約 40 分	7 行	10 行
書記 (2)	約 70 分	20 行	17 行
進行	約 210 分	565 行	161 行

3.1.2 書き込みの特徴

書記係を用意した書記方式では、記録を残すためには書記係が内容を理解することが必須条件であるため、難解な部分になると読み会は進行しても記録は残らないことがあった。音声のない進行方式の場合は、書き込みが進まなければ読み会が進行しないため、難解な部分になると理解した参加者が代わりに書き込むことや、積極的に議論が展開されることがあり、難解な箇所ほどよく説明された非常に好ましい記録が残った。但し、読み会中だけしか必要にならない司会者の交替の話や、閑数へ出入りするという宣言、それから雑談など、不要な発話も残った。

3.2 付箋紙表示制御方式

用意した 2 つの付箋紙表示制御方式が実際の読み会で役に立つかどうか調べるため、前節の進行方式に則ったチャット読み会を研究室内で行った。参加者は 8 名で、全員 Pierre による読み会を経験している。比較実験のために、表示 / 非表示切り替えはアイコンクリックのみで、共有指示棒やマウス位置に依存しない「全表示モード」を用意した。

10 分程度使い方を確認した後、3 つのモードを 20 分程度ずつ経験してもらい、その後は好きなモードを利用して最後まで読んでもらう形で読み会を行い、最後にアンケートによる評価を行った。

アンケートではそれぞれのモードについて操作性を、1 を非常に悪い、7 を非常に良いとする 7 段階で評価してもらったところ、表 3 の結果が得られた。表中の太字は中央値を表す。

表 3: 操作性

モード	1	2	3	4	5	6	7
ここまで			1		2	2	3
ここだけ		1	1		3	2	1
全表示	1	1	2	2	2		

また、従来の環境 (Pierre とチャットウィンドウを並べて利用し、付箋紙を貼らない) を含めた 4 つのモードについて、気に入った順に順位を付けてもらい、その平均を求めた結果が表 4 である。

結果から「ここまでモード」が高い評価を得た

表 4: 平均順位

	平均順位
ここまでモード	1.3
ここだけモード	2.5
従来の環境	2.6
全表示モード	3.6

ことが分かった。簡単に表示 / 非表示が切り替えられるので、「全表示モード」では認められなかった大量の付箋紙の同時表示が認められる結果になったと言える。「ここだけモード」は、同時に表示される付箋紙が 1 枚以下なので、画面がすっきりして見やすいことを積極的に評価する側と、マウスがわずかに動いただけで付箋紙が消えてしまう操作性の難点や、付箋紙 1 枚では文脈が掴みにくいという難点を挙げる側に意見が割れた。

また、全てのモードに共通して大きくなりすぎた付箋紙が見にくいという意見があった。

3.3 適切な付箋紙のサイズ

付箋紙が大きくなりすぎる理由には以下の 2 つがある。

- 議論が活発になされた
- 不要な発話が大量に含まれた

本節では付箋紙のどの属性がどの程度大きくなると不快に感じるかを調べ、不要な発話を削除することにより、付箋紙サイズの縮小を試みた。

3.3.1 付箋紙のサイズの基準

3.2 節で作られた付箋紙 48 枚を 6 名に見てもらい、内容を無視してサイズだけを考慮した場合に、読みたくないと感じる付箋紙を選択してもらった。

この人数を各付箋紙の不快感を表す属性として、付箋紙に関する他の様々な属性 (縦幅、横幅、面積、全文字数等) との相関が最も強いものを調べたところ、付箋紙の全文字数であることが分かった。更に全文字数が 210 文字以下の付箋紙は誰も不快に感じていないことが分かったので、全文字数 210 文字以下の付箋紙を増やすことを目標にする。

3.3.2 不要な発話の削除

付箋紙からは以下の冗長な発話を削除した。

- Pierre の操作方法
- 実験の方法
- 読み会後に不要な発話
- 雑談やタイプミス等

読み会後に不要な発話とは、司会者の交替に関わる発話や、関数への出入りを宣言した発話、或は会話の進行を促すための相槌などである。

この処理に要した時間は読み会時間の 1 割程度であり、その結果 3.3.1 節の目標に当てはまる付箋紙がどの程度増加したかを調べ、図 9 に表した。

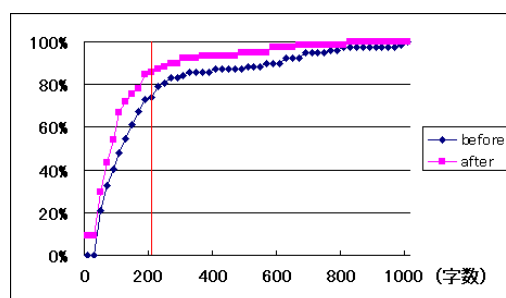


図 9: 付箋紙のサイズの変化

ここで横軸は全文字数を、縦軸はその文字数以下の付箋紙が全体の付箋紙に占める割合を表しており、全文字数が 210 文字のところに境界線が引かれている。図の下側の菱形でプロットされた線が削除前、上側の正方形でプロットされた線が削除後である。210 文字の境界線上を見てみると、削除前が全体の 75% を占めていたのに対し、削除後は全体の 85% 程度に増えているのが分かる。

しかし、幾分改善されたものの不快な付箋紙はまだ多く、残りの付箋紙に不要な発話がないことを考慮すると、サイズを小さくするには内容の編集作業が不可欠であると言える。

3.4 付箋紙の復習への利用

3.3 節で整理した付箋紙を利用して、付箋紙が復習にも効果があるか調べた。被験者は 7 名であった。Pierre の利用方法を簡単に説明した後に、付箋紙なしを含む 4 つの付箋紙制御方式をそれぞれ 5 分ずつ利用してもらい、アンケートで評価を行った。

付箋紙なし以外の3つのモード間に特徴的な差異は認められなかったが、3つ全てのモードが付箋紙なしに比べて遥かに好評であった。また、付箋紙は役に立つと思うかどうかについて7段階で評価してもらったところ、表5の結果が得られた。

表 5: 付箋紙の有用性

項目	1	2	3	4	5	6	7
付箋紙は役に立つか					1	4	2

以上の結果から、付箋紙は復習にも有効であることが分かった。また、被験者の中には今回初めてソースコードを読む参加者もいたが、付箋紙は好評であり、独習用にも効果があることが分かった。

4 関連研究

読み会を直接取り上げた研究はないが、企業がアプリケーションソフトを製作する過程において、プログラムの質を高め、バグを効果的に探すために利用するコードインスペクションと呼ばれる手法をそのまま採り入れた ICICLE[3] がある。役割分担を予め明確にする、参加人数に制限があるなどの特徴があり、本研究の目指す学習目的の読み会とは方向が異なっている。

また、チャットを直接貼る研究として、例えば Microsoft Word のドキュメントに対して特定の箇所にチャットを貼り、その部分に関して議論をし、文章を訂正することを目的とした Anchored Comment[4] がある。ソースコードとは異なり、ドキュメントの多くの部分は説明の必要がなく、記述が部分的で済む点、そして相手が遠隔地にいる時以外は必要にならない点が、本研究とは異なる特徴である。

5 まとめ

読み会支援システム Pierre とチャットを融合したチャット読み会システムを提案し、実装した。

評価実験により、2 つ以上の小グループに分かれて遠隔チャット読み会を行うことによって、小グループ内で音声を利用して議論できる上に、記録係を設けるよりも遥かに望ましい記録が手に入る

ことが分かった。

また、チャットの表示方法で採用した「ここまでモード」を利用すれば、多くの情報を同時に画面に表示できることが分かった。

付箋紙は読み会だけでなく、復習や独習時にも有効であることが示された。

6 今後の課題と展開

付箋紙にはチャットの全記録が記されるため、不要な発言や、冗長な発言が含まれるので、これを容易に編集できる手段を提供する必要がある。

構造が複雑なソースコードの読み会では絵を描きたいという要求も大きくなる。容易な描画が可能な描画ツールとの連携を可能にすることで、遠隔地でも読み会ができる可能性がある。

大規模な読み会でチャット読み会システムを使うとどうなるのか、実際に試してみる必要がある。

なお、本研究は学術振興会科学研究費補助金 基盤研究 C 「開かれたプログラム理解支援環境の構築と評価」(研究代表者: 寺田 実; 課題番号 13680395) を受けて行なわれた。

参考文献

- [1] 須山哲宏, “プログラム理解のためのグループウェア,” 情報処理学会第 61 回全国大会, 2000.
- [2] 首藤 達生, 寺田 実, “プログラム実行履歴情報に基づくソースコード読解支援システム,” 第 42 回プログラミング・シンポジウム報告集, pp.83-90, 2001.
- [3] L. Brothers, V. Sembugamoorthy, M. Muller, “ICICLE: Groupware For Code Inspection,” In Proceedings of the 1990 ACM Conference on Computer Supported Cooperative Work, pp.169-181, October 07-10, 1990, Los Angeles, California, United States.
- [4] Elizabeth F. ChurChill, Jonathan Trevor, Sara Bly, Les Nelson, Davor Cubranic, “Anchored Conversations: Chatting in the Context of a Document,” *Proceedings of the CHI 2000 conference on Human factors in computing systems*, pp.454-461, 2000.