

教育支援を目標とした遠隔ペアプログラミング環境の構築

中村 一星, 松本 章代, Martin J. Dürst

青山学院大学理工学部情報テクノロジー学科

1 はじめに

近年, 開発手法の一つとして, ペアプログラミングが注目されている. ペアプログラミングの効果に, コミュニケーションをとることでのペア間の知識共有による教育効果がある [1]. ただし, ペアプログラミングは 2 人が 1 台のパソコンの前で行うため, 同じ場所に行なければならないという制限が生じる. その制限を解消するために, インターネットを用いた遠隔ペアプログラミングがある.

我々は, 大学の大人数による演習での使用を想定し, ペア入れ替え機能が含まれた統合的な遠隔ペアプログラミング環境の構築を目指す. 本研究ではエディタ部分を Web アプリケーションとして実装する. 遠隔ペアプログラミングをするためには, 各クライアントのエディタに表示されるソースコードは常に同期されている必要がある. Comet [2] を利用することにより双方向通信を実現する. そしてソースコードにバージョン番号をつけて管理することでリアルタイムでの同期を実現する. またコミュニケーションを支援する機能としてソースコードの装飾を実装する.

2 同時編集可能なエディタの概要

今回提案するエディタの概要を述べる. 片方のクライアントがソースコードを編集すると前後の差分をとる. 差分の結果や id などの情報が含まれたデータをサーバに送信する. データを受け取ったサーバは差分をもう片方のクライアントに送信する. 送信すると同時にサーバにあるソースコードに差分を適用し保存する. データを受け取ったクライアントは差分を自身のソースコードに適用して表示する. 同期の仕方については後に詳細に述べる. またソースコードは iframe の designmode を on にすることにより, 装飾が可能である [3]. 装飾は下線, 取り消し線, 文字列に対する背景色の 3 種類が使用できる. この装飾も同期することが可能である. 動作画面を図 1 に示す.

Construction of a Remote Pair Programming Environment Aimed at Education Support
 Issei NAKAMURA, Akiyo MATSUMOTO and Martin J. DÜRST
 Department of Integrated Information Technology, College of Science and Engineering, Aoyama Gakuin University
 5-10-1 Fuchinobe, Sagamihara, Kanagawa 229-8558, Japan
 issei@sw.it.aoyama.ac.jp, {akiyo, duerst}@it.aoyama.ac.jp

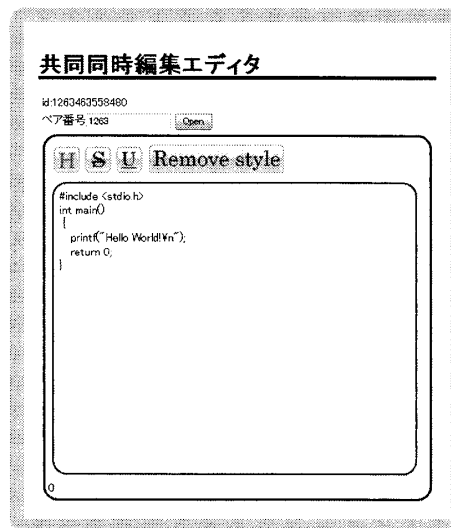


図 1: 動作画面

3 本アプリケーションの実装

遠隔ペアプログラミング用エディタには各クライアント上でソースコードを同期することが要求される. ほぼ同時にソースコードを編集した際, 各クライアントで表示されるソースコードに違いがでてしまわない. ここではソースコードの同期方法について述べる.

3.1 送受信するデータ

各クライアントで同期を行うための方法はいくつか考えられる. まずはソースコード全体を送受信する方法である. この場合クライアントではそのままデータを表示すればよく, 処理は単純となる. しかし, ソースコードが長くなるにつれ送受信するデータの量が多くなってしまいう問題がある.

そこで本アプリケーションでは差分を用いた同期方法を採用する. 編集前と編集後のソースコードの差分をとり, その差分を送信する. 差分を受信したクライアントはソースコードに差分を適用する処理を行う. この方法では送受信するデータの量はソースコード全体を送る方法に比べ少なくなる. また後述の管理方法により, 編集データが失われることもない.

本アプリケーションでは Google が提供している

google-diff-match-patch ライブラリ [4] を用いて差分を取得する。クライアント側では JavaScript, サーバ側では Python で記述されたライブラリを使用する。

送信するデータの一例を図 2 に示す。上からバージョン番号, カーソルの位置, 差分, クライアントの固有 id である。

```
{
  "rev": "488",
  "curP": "79",
  "patch_text": "@@ -1,6 +1,7 @@\n A\n+B\n C%3Cbr%3E\n",
  "id": "1261825220645"
}
```

図 2: 受信クライアントに送信するデータ

3.2 ソースコードのバージョン番号による管理

本アプリケーションでは編集データの衝突を避けるためにソースコードにバージョン番号をつけて管理する。処理の流れを図 3 の番号を用いて説明する。

各クライアントはサーバから取得したソースコードとバージョン番号を保持している。クライアントが送信するデータには「バージョンが何番目のソースコードに対する処理か」という情報が含まれている。クライアントがデータを送信しようとしたとき、サーバと送信するデータに含まれるバージョン番号が一致した場合のみ、クライアントが保持しているバージョン番号を書き換える。サーバでは差分情報が適用され、各クライアントに送信される (①~③, ⑥)。

サーバのバージョン番号と一致しない場合にはそのデータの送信は拒否される (⑤)。拒否された場合には別のクライアントが編集処理を行っていることを意味する。まずはその編集差分を受信し処理する (⑥, ⑦)。そしてバージョン番号を書き換え (⑧), 再度自身の編集による差分データを送信する (⑨, ⑩)。この仕組みにより, 編集データ衝突が回避できる。

この処理の仕組みは Subversion などのバージョン管理システムと類似している。本アプリケーションで異なるのは以下の 3 点である。ソースコードの衝突があった場合に自動で処理する。差分は 1 文字ごと, 連続で入力された場合には複数文字だけに限定される。差分適用箇所が必ず 1 つに限定される。

4 まとめ

本研究ではリアルタイムでの同時編集, 同期が可能なエディタの実装を目的として進めた。2 つのクライアント間で装飾を施したソースコードが同期できた。Firefox3.5 で動作することを確認した。今後はナビゲータの指示がより伝わりやすい装飾の検討, ペア入れ替え機能が含まれた統合的なペアプログラミング環境を構築する。

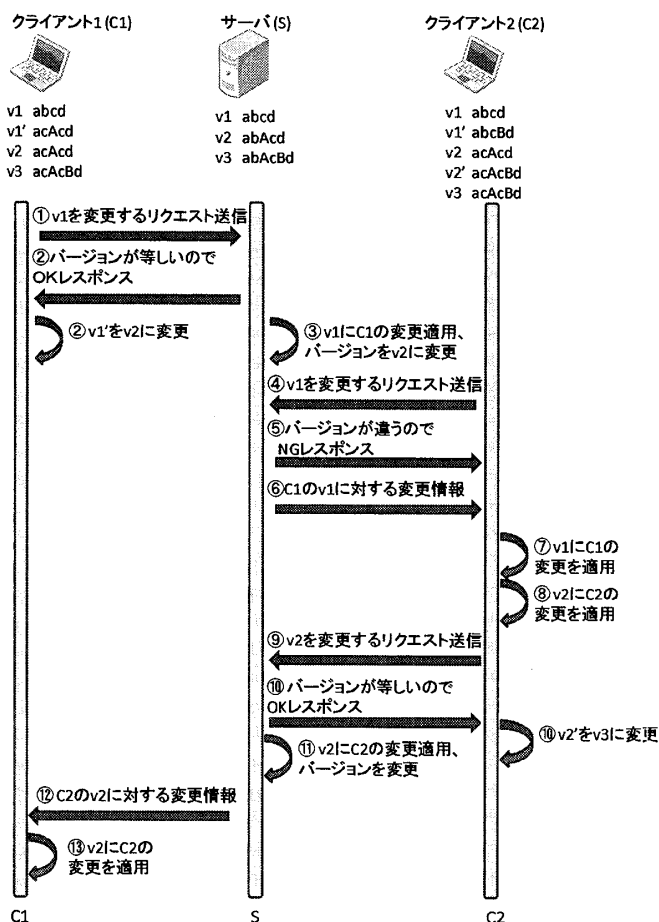


図 3: 本アプリケーションでの同期方法

謝辞

本研究は文部科学省科学研究費補助金 (基盤 C, 課題番号 21500905) の交付を受けている。

参考文献

- [1] 勝野進一. プログラミング実習におけるペアプログラミングの活用について. 実践教育, Vol. 21, No. 1, pp. 10-12, 2006.
- [2] Alex Russell. Comet. <http://alex.dojotoolkit.org/2006/03/comet-low-latency-data-for-the-browser/>.
- [3] Mozilla. Enabling rich text editing in your web page. <http://www.mozilla.org/editor/midas-spec.html>.
- [4] google-diff-match-patch. <http://code.google.com/p/google-diff-match-patch/>.