

既存のエディタを利用する少人数向け共同開発環境の設計と実装

山下 剛† 佐藤 喬† 多田 好克†

†電気通信大学大学院情報システム学研究所

概要

インターネットを通じて小人数で開発する協調開発が日常的になってきた。協調開発では開発者同士が状態を把握している状態の Awareness が重要となる。

協調開発で Awareness を実現するシステムが提案されている。しかし既存のシステムは Awareness を実現するために、使用できるプログラミング環境の選択が制限されるものが多い。

本研究では Awareness をユーザの好みのプログラミング環境において実現する協調開発支援システムの枠組みを提案し、それを使ったシステムの実装を行った。

1 協調開発と Awareness

インターネットを通じて複数の開発者が共同でソフトウェアを開発する協調開発が一般的になった。

協調開発の問題として、作業を行なう者同士が同じ空間に居ないため互いの情報交換、作業の進捗の把握などの情報共有には、意識的な情報収集が必要である。

協調作業を行なっている者が互いの作業状況を把握している状態を Awareness と呼び、Awareness の実現に利用できる情報のことを Awareness 情報と呼ぶ。この Awareness 情報を開発者に受動的に示すことで、開発者は互いの状況の情報を確認する必要なく情報共有が可能となる。

2 協調開発支援システムの導入の問題点

協調開発の分野では、協調開発で Awareness を実現するシステムが提案されている。既存のシステムでは、エディタやソースコード管理システム (SMS) といったプログラミング環境から情報を収集し Awareness 情報を提供している。

しかし、既存システムでは Awareness 情報の提供のために、内部情報取得が容易な特定のエディタ、SMS を指定しているものが多い。既存のシステムを導入する場合には、開発者は指定されたエディタ、SMS という制限された環境での開発を行なうこととなる。

既存のエディタ、既存の SMS から情報を収集を可能にすることで、既存のプログラミング環境を保ったまま、Awareness を提供する協調開発支援システムの利用が可能となる。

本研究では、既存のエディタ、SMS を利用した Awareness 情報の配信を行なうミドルウェア CAEDE の設計および実装を行なった。さらに CAEDE を利用した協調開発支援システムとして、Emacs と、CVS を利用した協調開発支援システムの実装を行なった。

3 関連研究

協調開発支援では Awareness の実現のために色々な物が提案されている。CAISE[1] では、リアルタイムに他人の編集画面を確認できる Awareness 情報を提供している。GRAM[3]

Design and Implementation of Collaboration Development Environment for a small group with ordinary Editor

†Tsuyoshi Yamashita †Takashi Satou †Yoshikatsu Tada

†Graduate School of Information Systems, The University of Electro-Communications

では、ユーザの状況や編集の状況を確認できる開発環境を構築し、競合の発生を抑える Awareness 情報も提案されている。しかし、これらのシステムではシステム設計者が指定したエディタしか利用できない。本研究では、既存のエディタ、SMS を利用して Awareness を実現し、この問題を解決する。

ソフトウェア開発の手法は様々である。このため協調開発支援を行なうシステムは拡張性が高いものである必要がある [4]。この考えに基づき本研究で提案する CAEDE は、高い拡張性を意識したシステムを構築している。

4 既存環境からの Awareness 情報取得

ここでは既存のプログラミング環境である、エディタ、SMS からの Awareness 情報の取得について述べる。既存のプログラミング環境から情報を取得することで、既存の環境を保ったまま Awareness を実現する協調開発支援システムの導入が可能となる。

4.1 エディタからの情報収集

エディタから、今どのファイルを編集しているのか、そのファイルはどのように編集されたのかという情報を収集すれば Awareness 情報として利用できる。

既存の多くのエディタは、ファイルの保存時、ファイルの開閉時、指定した時間ごとにユーザが指定した任意のプログラムを実行可能である。この機能を利用し、ファイルの保存時、ファイルの開閉時、一定の時間ごとにプログラムを実行し、そのプログラムを通じてエディタの内部情報を収集する。こうすることで、情報収集の容易ではないエディタからでも情報収集が可能となる。

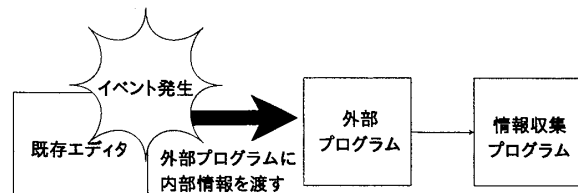


図 1: エディタからの情報収集

4.2 SMS からの情報収集

SMS から取得できる情報は Awareness 情報として有益である。例えば、SMS の情報を通知することで、最後にコミットしたものは誰なのか、その変更は何のために行なわれたのかといった情報を確認できる [2]。

SMS のコマンドのラッププログラムを作成することで情報収集が可能である。ラッププログラムで情報収集を行なうことで、SMS のコマンドを利用していたユーザは既存のコマンドの知識を流用できる。さらに SMS のコマンドから情報収集するため、サーバと独立して情報収集を行なうことができる。

5 Awareness 情報配信システム CAEDE

Awareness 情報配信を行なうミドルウェア CAEDE を実装した。CAEDE は Awareness 情報を既存のエディタ、SMS から収集を行なうサーバ・クライアント型のミドルウェアである。

図2が CAEDE の概観である。CAEDE クライアントは通信プログラム、ラッププログラムを通じて既存のエディタ、SMS コマンドと通信し情報収集する。CAEDE クライアントは収集した情報を CAEDE サーバに送信する。CAEDE サーバは集積された情報を元にして Awareness 情報を CAEDE クライアントに送信する。CAEDE クライアントは受信した Awareness 情報を Awareness 表示部に表示することでユーザの Awareness を実現する。

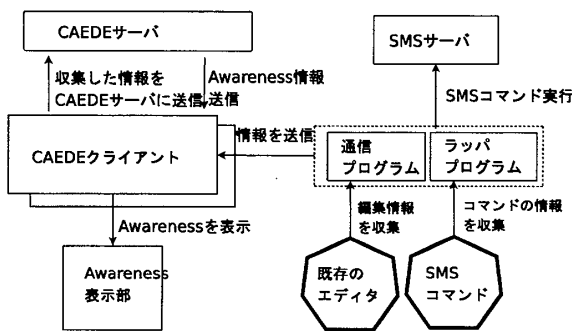


図2: CAEDE の概観

CAEDE で行なわれる通信は CAEDE プロトコルという独自のものである。CAEDE プロトコルは ':' で区切られたテキストベースの単純なものである。

単純なプロトコルで通信を行なうことで新しいラッププログラム、通信プログラムを作成することが容易となる。その結果、CAEDE に新しいエディタ、SMS を対応させることが容易となる。通信プロトコルを規定することで、別の協調開発支援システムが通信プログラム、ラッププログラムを一般的なエディタ、SMS の情報収集する際にも利用できる。

6 CAEDE と CVS、Emacs の連携

CAEDE と実際に既存のエディタ、SMS を連携させる例を示す。ここでは、エディタとして Emacs、SMS として CVS を選択した。

6.1 Emacs との連携

Emacs と CAEDE の連携のために transcaede というプログラムを実装した。図2の通信プログラムが transcaede に対応する。イベント発生時に transcaede を実行することで CAEDE と Emacs を連携できる。例えば fileA を開いた際に、transcaede を実行し、CAEDE クライアントに fileA を開いたという情報を伝えることができる。またこの構造であれば、このプログラムは外部プログラムを呼び出せる別のエディタからも利用できる。

Emacs の機能である EmacsLisp を利用し、以下のイベント発生時に transcaede を実行するよう設定した。

- ファイルを開いた時
- ファイルを閉じた時
- ファイルを保存した時

60 行程度の EmacsLisp を書くだけで上記のタイミングで transemacs を実行する設定を実現できた。CAEDE プロトコル

が単純であるため、CAEDE に情報の送信を行なう transecaede 自体は 50 行程度の小さな Ruby プログラムで実現できた。

同様の手法は他のエディタでも可能である。この場合も上記のタイミングで transcaede を実行する設定を書くだけで良い。この設定は Emacs の場合と同様短く書くことができる。

6.2 CVS との連携

CAEDE と CVS を連携させるために、mycvs というラッププログラムを実装した。これは図2のラッププログラムに対応する。

mycvs は cvs コマンドのラッププログラムとして動作し、以下の操作を行なう時に CAEDE に情報送信を行ない連携する。

- リポジトリにファイルを追加した時
- リポジトリからファイルを削除した時
- リポジトリのファイルを更新した時

このような構成にすることで、mycvs コマンド利用者は cvs コマンドと同じ方法で利用するだけで上記のタイミングの時自動的に CAEDE クライアントに情報が送信される。

この mycvs コマンドは CAEDE プロトコルに沿ったメッセージを CAEDE に送信するもので、transcaede と同様に 50 行程度の小さな Ruby プログラムで実現できた。また、subversion といった別の SMS にも対応する場合にも、mycvs と同様の操作を行なう簡単なラッププログラムを書くことで対応できる。

7 結論

本研究では、既存のエディタ、SMS から情報を収集し Awareness 情報として配信するミドルウェア CAEDE の設計および実装を行なった。協調開発において開発者は CAEDE を利用することで、既存のエディタ、SMS を利用したままで Awareness を実現可能となる。また、エディタ、SMS からの情報を取得するためのプロトコルを定義し、エディタ、SMS を CAEDE に対応させるプログラム作成を容易にした。

今後は CAEDE の動作の性能の評価を行なう予定である。特にネットワーク帯域が十分でない場合でも、遅延なく Awareness の実現が可能かどうかは協調開発支援システムにとって重要な課題である。そのため、低速なネットワークや頻りに通信を行なうような環境での性能評価を実施する。

参考文献

- [1] C. Cook and N. Churcher. Constructing real-time collaborative software engineering tools using caise, an architecture for supporting tool development. In *ACSC '06: Proceedings of the 29th Australasian Computer Science Conference*, pp. 267–276. Australian Computer Society, Inc., 2006.
- [2] G. Fitzpatrick, P. Marshall, and A. Phillips. Cvs integration with notification and chat: lightweight software team collaboration. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pp. 49–58. ACM, 2006.
- [3] K. Takata and J. Ma. Gram - a p2p system of group revision assistance management. In *AINA '04: Proceedings of the 18th International Conference on Advanced Information Networking and Applications*, pp. 587–592. IEEE Computer Society, 2004.
- [4] M.A.S. Mangan, M.R.S. Borges, and C.M.L. Werner. A middleware to increase awareness in distributed software development workspaces. *WebMedia and LA-Web, 2004. Proceedings*, pp. 62–64, Oct. 2004.