

SeeThroughTalk - a collaborative image space

SeeThroughTalk - 息吹の伝わるコラボレーション環境

1 背景

学術、および産業の両方面において、チームによるプログラミング環境は、今まで多くのものが考え出され、発表されてきた。しかしながら、それらの環境は主に「ソースコードが書かれたファイルの統合」というパラダイムにとらわれており、プログラミングにとっては、副次的な中間生成物であるファイルというものに重点が置かれすぎていたように思われる。本当に「チームで行う」プログラミングにとって必要なのは、チームのメンバー間が相互作用し、ダイナミックな関わりの中で自らの考えを形にし、プログラムとして昇華させることのできるような環境である。

本プロジェクトでは、本当にメンバー間が協力しながらプログラミングできる環境を指向する。これはソースファイルという物理的なものにフォーカスするのではなく、より人間の関わりに重点を置いたアプローチである。個人の集中的な作業を妨げることなく、必要に応じて各メンバーとスムーズにコミュニケーションが取れるような、個々人の間の「快適な距離」を自然な形で提供するようにしたい。

このような環境がマルチプラットフォーム、インターネットという文脈で動作することにより、様々な国の開発者同士が、遠隔地に居ながら協力してプログラムを作っていくことができるようになる。XP のペアプログラミングは、快適で生産性の高いプログラミング手法として認知されつつあるが、物理的に1つの場所にいなければならないという制約がある。本提案では、オリジナルのペアプログラミングが持つ障壁を取り払い、ペアないしマルチユーザがユビキタスに協働できる環境を開発する。

2 目的

もともと Smalltalk では、非常に対話的な方法でプログラムを作成できる。開発環境の中で、プログラム上のオブジェクトを自由にブラウズし、操作することができる。コードのある部分に問い合わせのメッセージを送ると、システムは、即座に結果を返してくれる。一般的に、興味深いアイデアを育てるには、フィードバックを対象から即座に受けとることが重要である。その点において Smalltalk は、非常に理想的なプログラミング環境といえる。現代の多くのプログラミング言語は、IDE によってそのような応答性を達成しようとしているが、オリジナルの Smalltalk のレベルに達していないのが現状である。しかし一方で古典的な Smalltalk の環境というのは、ただ個人のみをターゲットにしていたことも事実である。それはプログラミング(或いは、タイルスク립ティング)といった作業を、共同で行わせるための機能を持たない。Smalltalk 仮想イメージは、時折「孤独な場所」と評されるくらいである。興味深いアイデアを展開するためには、システムからだけでなく、様々な人々からのフィードバックを必要とするはずである。Smalltalk プログラミング環境を共有可能にすることは長年望まれてきた。

チーム開発環境を作るいくつかの試みは、主に商用の Smalltalk で行われてきたが、これらは伝統的なファイルベースの言語に影響を受けすぎており、プログラミングパートナーとの創造的な関わりを支援してくれない。結局、そうしたシステムが扱おうとしているのは、パッケージと呼ばれるソースコードの束であり、人間ではなかった。

我々は、Smalltalk で真の協調作業空間を実現したい。それは、人間の交流に焦点を合わせたものであり、快適な「間」を支援するものとなる。個人の集中的な作業を妨げずに、かつ、ダイナミックな人々との関わりを奨励するものだ。

これを実現するため、本システム SeeThroughTalk では、一般的なデスクトップメタファを拡張する。半透明の「オーバーラッピングデスクトップ」を既存の Morphic GUI に導入する。ユーザは、背景を通じてパートナーの作業をオブジェクトとして「見通す」ことができる。SeeThroughTalk によって、我々は各個人の集中を妨害せずに、協調プログラミングをしていくことが可能になる。

3 開発の内容

我々は、全てのユーザに強力な共有空間を提供することを目指している。これを達成するため、複数レイヤの共有デスクトップについてアイデアを練った。

基本となる概念は、かなりシンプルなもので、1つの場所にデスクトップを重ねただけである。特異な点として、それらのデスクトップは、背景が半透明と考える。そのためユーザは、他のメンバのアクティビティを見通すことができる。

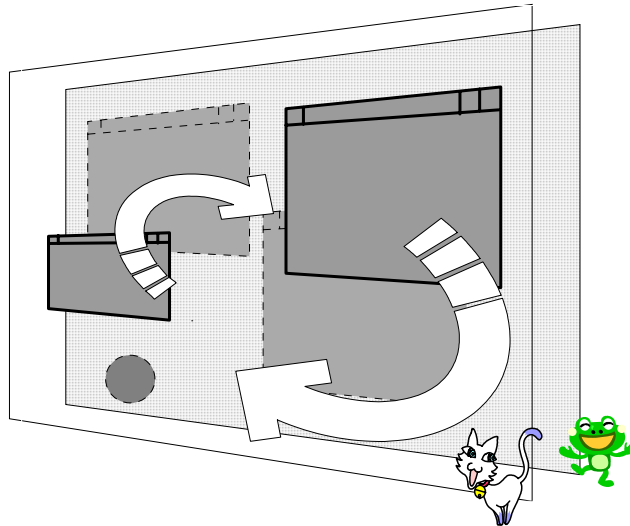


図1:複数レイヤの共有デスクトップ

この仮想空間は、4つの特徴を持つ:

1. 各人は、自身のパネル(デスクトップ)を持っている
2. 各パネルは、半透明である。従って、他のパネルを同時に見ることができる
3. ユーザは、別のパネルを訪問できる
4. 全てのオブジェクトは、共有可能である

このインタフェースは少し奇妙に思われるかもしれないが、我々は注目すべき長所があると考えている。まず、ユーザは意識的にも無意識にも他のアクティビティを見ることが可能になる。プライベートモードと協調作業モードとを非常に簡単に行き来することができる。タイル上にフラットにウィンドウを並べた形式の遠隔デスクトップと比較すると、必要とされるデスクトップの領域も節約される。

Smalltalkの実装の1つであるSqueakには、すでにNebraskaと呼ばれる非常にユニークなデスクトップシェアリングシステムがある。そのシステムでは、ユーザは、デスクトップを容易に公開し、他の遠隔ユーザに対して対話型セッションを始めることができる。しかし、システムにはいくつかの問題がある。小規模のタイルスク립ティングをするのには非常に有益であるが、より複雑なスク립ティングやプログラムを行おうとすると、破綻してしまう。

SeeThroughTalkでは、協調作業用ユーザ・インタフェース(アバター、エアーチャット等)を提供する。さらに、ローカルのイメージでのあらゆるプログラム上の変更は、他のユーザに自動的にアップデートされる。編集中のクラスやメソッドを明示的にロックすることをユーザに強要しない。この機能により、多くの伝統的なソースコード管理システムで不可欠と考えられている退屈なチェックイン/チェックアウトの作業が取り払われる。グループ内での全てのイメージは、原則としていつでも同じクラス、メソッドを保持することになる。ただ、オーバーラップしたデスクトップの順番が異なるのみである。

4 従来の技術との相違

実装には Squeak を用いた。開発期間の中で、すでに下記のようなユニークな機能を実現している。

• オーバーラッピング遠隔デスクトップ

SeeThroughTalk は、共有された仮想デスクトップ空間を提供する。ユーザは、半透明の背景から他のデスクトップを見ながら、デスクトップ体験を共有できる。アバターを単にクリックすることによって、ユーザは、最も興味深いデスクトップに飛び込むことができる。それは、協調作業へのオーバーヘッドを減少させ、奨励する。

• Morph とプロジェクトの移動

仮想デスクトップ上のオブジェクトは、全て共有可能である。ユーザは、Morph をターゲットアバターへ単にドラッグするだけで、リモートのデスクトップに移動させることができる。Morph は、スクリプトや関係を保ち、「生きたまま」他のデスクトップに移動する。

さらに、イメージ間でスクリプティングプロジェクトを直接交換することも可能である。これは、他人からのフィードバックを得てスクリプティングプロジェクトを発展させるような場合に有効である。

• 変更伝播

ソースコードは、Smalltalk においては「オブジェクト」である。このため、SeeThroughTalk がソースコードを共有可能にすることは、自然と言える。ローカルなイメージになされたあらゆるソース上の変更は、自動的に他のイメージに適用される。これは、動的な協調プログラミングのための本質的機能である。ネットワークエラーに対処するため、ユーザは、コードの特定部分を Browser からのドラッグ&ドロップで送ることができる。(前述の Morph 移動を使えば、ユーザが書きかけの Browser そのものを他のイメージに投げ送ることすら可能である)。

• 協同作業支援 UI

アイコンなどの工夫により、各メンバのカーソルやウィンドウは、容易に判別できる。カーソルは、吹き出しによる会話「エアーチャット」をサポートする。アバターは、ユーザを代表し、そして、ユーザ設定を格納し、またユーザの状態を効果的に表示できる。伝統的な開発ツールは拡張され、アバターアイコンでリストを装飾するようになった。この結果、メンバは、一見してコードの変更履歴をつかむことができる。

• UltraSwiki

UltraSwiki は、協同作業による成果を公開するための新しい wiki である。それは、容易な保守のためにチームプロファイルを格納する。簡単なオペレーションによって、ユーザは、プロジェクト(プログラム、タイルスクリプト)をサーバにアップロードできる。UltraSwiki は、スクリーンショットを表示し、成果物を HTML としてリストする。他のユーザは、通常のウェブブラウザによってそれらをダウンロードして、試すことができる。

5 期待される効果

SeeThroughTalk によって、遠隔地での真の意味での協調プログラミングが可能になる。これによって、インターネットを通じた、各種のオープンソースプロジェクトの開発が、よりインタラクティブに行えるようになる。また、LAN のような比較的狭い範囲であっても、XP で言うところのペアプログラミングの支援環境として使用することができる。

これらは、プログラミング支援環境として SeeThroughTalk を見た場合の基本的な効果であるが、SeeThroughTalk 自体は、汎用的な遠隔協調環境でもあり、そのほかの用途に応用することも可能である。

たとえば、遠隔デスクトップを利用してのリモートサーバの監視、変更伝搬を利用しての、複数マシンへのソフトウェア一括メンテナンス、チャット GUI を通じての e-ラーニングなどに用いることができる。

6 普及の見通し

SeeThroughTalk は、オープンソース、かつフリーのものとして、誰もが利用できる状態で、Web 上に公開する予定である。Squeak 上で開発されていることもあり、システムは、Windows、Linux、Mac など、非常に多くのプラットフォーム上で動作する。

配布の形態は Squeak で一般的になりつつある sar 形式とし、SqueakMap(Squeak コミュニティでのソフトウェア検索サービス)に登録する。インストールは Squeak 上のパッケージ管理ツールである Package Loader から、メニューによる選択のみで完了するようにする。

これらの工夫を行うことにより、まず Squeak コミュニティ内での、認知度を高め、ユーザを徐々に増やしていく。特にユーザビリティの向上が、この種の環境の普及には非常に重要となるので、正式なリリースを行う前に、少人数でのワークショップを行い、フィードバックを得ることも考えている。また、ユーザ間での情報交換や、新たなプログラミングパートナーを探すためのツールとして、UltraSwiki による SeeThroughTalk のポータル・サイトを構築することも予定している。

7 開発者名

梅澤真史 (株式会社 豆蔵 ビジネスソリューション事業部、email: umejava@mars.dti.ne.jp)

8 URL

詳しくは下記を参照のこと。

SeeThroughTalk ポータル・サイト: <http://swikis.ddo.jp/SeeThroughTalk>