

6H-4

分散開発環境における共同作業用ツールの試作*

～ダイナミックマルチコネクタ～

二上俊嗣 坂本泰久†

NTTソフトウェア研究所‡

1 概要

ネットワークの発達とともに遠隔地間での共同作業が必要になってきている。例えば

- 同じドキュメントを同時に複数の人間に見せたい。
- ツールの使い方を画面を見せながら説明したい。

などである。

このようなことをするためには複数のユーザによるツールの共有が必要である。しかし複数のユーザと同時に通信できるような機能を持つツールは少ない。このようなことをツールに手を入れることなく可能にするのがダイナミックマルチコネクタである。これはネットワークに接続された複数のコンピュータを用いて共同作業を実現するための基本的なアプリケーションである。

ダイナミックマルチコネクタはツールを端末から独立させることによってツールと端末との自由な接続を可能にするものである。これにより1つのツールを複数の人間で操作したり、逆に1つの端末からの入力力で複数のツールを操作することができる。ツールと端末との通信にはTCP/IPを使っているのでネットワークを越えた操作ができる。

2 設計

2.1 モデル

端末とツール間に通信機能を持たせることにより両者の自由な接続を可能にする。端末とツールとを分離するのでシステムはサーバクライアント方式で実現する。ツールはサーバから起動されて動き続ける。サーバはクライアントからの要求があれば接続の処理をする。端末側はクライアントを起動してサーバに接続する。(fig1.)

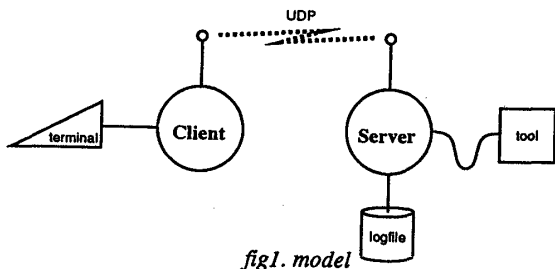


fig1. model

サーバおよびクライアントの基本的な役割は次の通りであ

*Implementation of cooperative tool on distributed environment

†Toshitsugu Futagami, Yasuhisa Sakamoto

‡NTT Software Laboratories

る。

サーバの役割

- ツールを起動する。
- クライアントからの入力を受けとりツールに渡す
- ツールの応答をクライアントに送る

クライアントの役割

- 端末からの入力をサーバに送る
- サーバの応答を端末に出力する

2.2 利用例

非常に汎用的なツールなので多くの利用方法が考えられる。以下に代表的な例を挙げる。

1. 1つのツールの共同利用 (fig2.)

複数の人が同じ画面を見ながら1つのツールを使うことができる。しかもツールを改造する必要がないので、既存のエディタを用いて複数の人によるドキュメントの同時編集ができる。またエディタをオンライン通信用の掲示版として使うこともできる。ただしこれらの例のようにツールを共同作業用として使うためには入力の制御方法が問題になる。

2. 教育・デモ・説明用 (fig3.)

オンラインで画面を見ながらツールの使い方の説明・デモなどが行なえる。特に教育用に効果があると思われる。

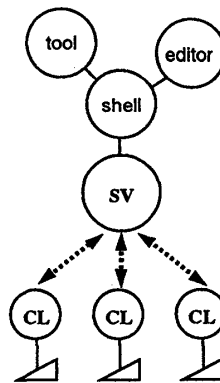


fig2. co-operation

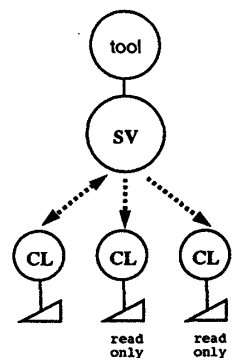


fig3. demonstration

3. 端末の切替え (fig4.)

ツールの環境を保存したまま端末だけを切替えて継続して使うことができる。例えば翌日までシェルの環境情報や履歴情報を失うことなく保存しておくことが可能になる。

4. 複数ホストに対する同時オペレーション (fig5.)

1つのクライアントから複数のサーバにつなぐことにより、それぞれのサーバにつながっているツールに対して同時に入力を与えることができる。複数マシンの同時インストールなどに応用できる。ただつなぐだけでは複数のサーバからの応答を受けとってしまうのでモニタ用のクライアントを接続する必要がある。

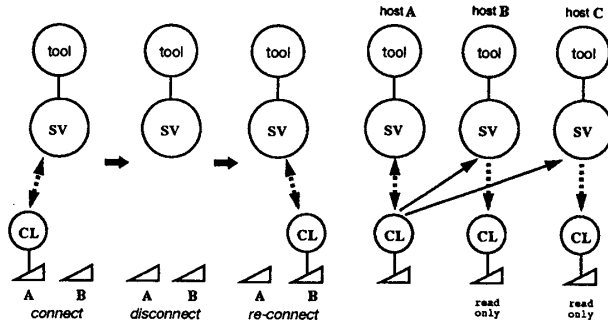


fig4. change terminal

fig5. multiple operation

3 実現

2.1のモデルに基づいてシステムを実現した。

サーバとクライアント間の通信方法としてTCP/IPプロトコルの1つであるUDPを用いた。これによりクライアントは他のマシンで動いているサーバにも容易に接続できる。

サーバの名前をmcserv、クライアントの名前をmcとした。mcはmulti-connectの略である。

3.1 付加機能

実現にあたって実際の運用時に必要になると思われる機能を付け加えた。

1. アクセス権制御

ツールはサーバを立ちあげたユーザの権利(ID)で動くので、すべてのクライアントから操作されては困る場合がある。クライアントのサーバに対する権利として次の3種類のアクセス権を設定する。

- read 権 サーバからの出力を受けとることができる
- write 権 サーバに入力を与えることができる
- notice 権 サーバの管理情報を知ることができる

これらのアクセス権は主制御権のあるクライアントによって変更される。

2. 主制御権

いくつかのクライアントはサーバに対する主制御権を持つ。主制御権を持ったクライアントは以下のことができる。

- サーバおよび全クライアントの終了
- クライアントのアクセス権の変更
- クライアントのデフォルトアクセス権の変更

クライアント起動時に主制御権が与えられるのはそのサーバを起動したユーザがサーバを起動したユーザだけである。

3. エスケープモード

システムの状態を変えるときのインタフェースとして、クライアントにエスケープモードを付けた。エスケープモードでサブコマンドを入力することにより以下のような操作を行うことができる。

- サーバとの接続を終了する。
- サーバに接続しているクライアントの情報を知る。
- クライアントの制御権、アクセス権を変更する。
- サーバを終了させる。

4. ログ機能 (プレイバック機能)

ツールの画面出力を保存し再生することができればデモや教育を行なうときに便利である。サーバはツールの応答をクライアントに送信するものであるが、オプションを指定することにより応答をファイルにも記録することができる。(fig1.) その記録ファイルを専用の再生プログラムに通すことによって同じ画面を再現できる。ファイルには時間情報を付与して記録するので、ビデオのように動画像的に再生できる。

5. 複数サーバの管理

1台のマシンでサーバをいくつも動かすこともあり得るのでサーバに名前を付けて管理する。クライアントを起動する時にサーバ名を指定すればそのサーバに接続できる。

3.2 評価

実用に耐える速度が得られた。

3.3 課題

1. イベント入力の制御方法

ツール(特にエディタ)を複数の人間で操作する場合には入力をまったく制限なく先着順に受け入れると混乱してしまうのでどのように制限するかが問題となる。

2. ウィンドウシステムへの対応

本システムは入出力デバイスとしてOS上の標準入出力しか扱っていないため、キャラクタ端末上のツールにしか対応できない。ウィンドウシステム上のビットマップを利用したツールへの対応を考える必要がある。

4 おわりに

今回はツールと端末の間を自由に接続する機能を中心にツールを試作した。今後はサーバ・クライアントに送信受信の両方の機能を持たせることによって両者の統合化を試み、モデルの汎用化を進めていく。さらにポートおよび接続の概念を一般化することにより、ツールと端末の接続だけでなくあらゆるプロセス間の自由な通信がアプリケーションレベルで実行できるような環境を構築していきたい。

参考文献

- [1] Geoffrey M.Lee: *Integrating UNIX Terminal Services into a Distributed Operating System*, USENIX Winter'89(1989)
- [2] 荒川・柴崎: ウィンドウ共有管理システム XMULTI, 第40回情報処全国大会 6S-1(1990)