

LTB - シェルの構成

瀧塚 孝志 ・ 杉村 領一

国際電信電話(株) ・ 新世代コンピュータ技術開発機構

1. はじめに

文解析や文生成など独立に開発された様々なツールが集まってLTBを構成する。これらのツールが独自のインタフェースを持ち、密接に結合していると、ユーザは感心を持つ一部の機能を自分用に変更することが難しくなる。

そのため、ツールを柔軟に結合し、統一的なインタフェースを提供するシェルを開発することにした。本稿では、現在開発中のシェルの構成方式について述べる。

2. 結合方式

様々なモジュールを柔軟に結合する方式として、オブジェクト指向プログラミングによる方式とPIPE機構による方式があり、ソフトウェアの再利用方式として知られている。前者は、クラスの階層関係やメッセージ・プロトコルが理解し易い密接に関連したモジュールの結合に適する。後者は、標準入出力という単純な結合で済む独立性の高いモジュールの結合に適する。

解析や生成などのLTBのツールの多くが比較的独立に動作するので、本シェルでは各ツールをプロセスとして管理する。そして、プロセス間を動的にPIPE機構により結合する方式を採用することにした。

プロセスに送信用と受信用の2本の通信路(チャンネル)を与えると、送り手と受け手で送受が逆転するため、通信路の名前付けで混乱を起こすことが多い。そこで各プロセスには、双方向通信が可能な通信路を、シェルとの通信用とプロセス間通信用の2本を設定する。

プロセス間の通信方式として、①パイプライン的にデータを流す方式と、②階層プロセスのように隣接する2つのプロセスの双方とデータの送受を行う方式と、③沢山のプロセスが通信し合う方式が考えられる。この3つのモデルにおいて、データの送受際の相手の引数が異なり、①では相手に関する引数

がなく、②では左右で区別され、③ではプロセスの名前により区別される。

シェルは、各プロセスで指定された通信方式を満たしながら、この3方式間の相互接続が行えるように通信路を接続する。シェルとの通信路は、シェルから③、ユーザ・プロセスから①の方式で通信が行えるように接続される。プロセス間の通信路は、プロセス起動後に動的に接続されるので、そのプロセス自身で接続相手を指定することもできる。また通信路には、アクセスを知らせるクラス(Notify機能)を動的に設定することができ、実行時間の計測や入出力データのウィンドウ表示などができるようにになっている。

基本的な通信路は、SIMPOSのストリームを、スタック上のデータも流せ、putとgetの差をカウントするように拡張して実現した。ストリームは、複数のプロセスからのデータを待ち合わせられるので、送信時にプロセスの名前をデータに付与することにより、容易に多方向通信が実現できる。

3. シェルの構造

本シェルは、ユーザ・プロセスの生成/消滅や通信路の接続を管理するシェルの核となるプロセスと、インタフェースやヘルプ機能など提供する複数のサービス・プロセスからなる。機能毎にプロセスとすることにより、シェルをカスタマイズし易くなるとともに、ヘルプの途中で例題を実行することなどが可能となる。さらに、サービス・プロセスが止まったときなどでも、そのプロセスを強制終了させ再起動できるので操作性が向上する。

シェルの起動すると、核プロセスとシェルのウィンドウ・プロセスとが生成され、シェルの起動したプロセスには核プロセスと通信するためのインスタンスが返される。このインスタンスを用いて起動するプロセスの指定や、通信路の接続法の指示などを行う。シェルの起動するプロセスに定義ファイルを読み込む機能を持たせることにより、ユーザの好みの初期設定を行うことができる。

Design of LTB Shell

Takashi TAKIZUKA* and Ryouichi SUGIMURA**

* KDD Kamifukuoka R & D Laboratories, ** ICOT

ウィンドウ・プロセスは、ユーザ・プロセスをアイコンとして表示する。マウスなどにより、アイコンにコマンドが渡されると、プロセス名とコマンドの種類を組にしたメッセージを核プロセスに送る。核プロセスは、受信したメッセージがプロセスの動作に関するコマンドであれば、対応するプロセスにコマンドを送り、ヘルプコマンドなどであれば、対応するサービス・プロセスを起動する。

核プロセスが管理するプロセスの状態として、初期化モードと、核プロセスからのコマンドを受け付ける従モードと、動作を行っている最中の主モードを設定した。プロセスが起動され、最初にシェルとの通信路から読み込みを行うと、通信路に設定されたNotify機能により、そのことが伝えられ、初期化モードから従モードに移る。核プロセスがプロセスに動作指示コマンドを送るとき、状態を従モードから主モードに移す。

主モードのプロセスは、邪魔になるシェルのウィンドウを消去するコマンドを、核プロセスを経由してウィンドウ・プロセスに送ることができる。ウィンドウを消去したプロセスが強制中断されたときや、従モードになるコマンドを送信したとき、シェルのウィンドウが再度開かれる。

4. 柔軟性

核プロセスは、総てのプロセスからのメッセージを同様に処理しているので、ユーザ・プロセスが別のプロセスを起動したり、別のプロセスに動作指示を与えたりすることもできる。さらにユーザ・プロセスが、別のシェルをその下で起動することもできる。シェルは、起動したプロセスと内部のプロセスを接続することのできる通信路を持っている。あるツール内で、そのツール独自の開発環境に移りプログラムを修正し、再度結合して実行することも容易に行える。

プログラム中にクラス名を埋め込んでしまうと、別のクラスで置き換えるときや、ロード後に埋め込んだクラスをコンパイルするときに問題が起こる。そのため、シェルが使うクラスやユーザ・プロセスを記述したクラスは、起動時にライブラリからロードするようにしている。シェルが使うクラスは、起動時に写像専用のクラスを呼んで、クラスをロードし、スロットに格納する方式を採っている。

プロセス名とプログラム名や通信方式、アイコンなどとの対応は、定義ファイルにより行われる。実行時のプロセス名とそのインスタンスとの対応は、ヒープ上に連想リストを作る方式と、名前とイン

デックスとのテーブルを表したクラスを動的に作成しコンパイルする方式の2方式が可能である。前者は変更が高速であり、シェルはこの方式でプロセスを管理する。後者はアクセスが高速であり、多方向通信の実現法として選択することができる。

5. ヘルプ機能

LTBには様々なツールが登録されるので、使い勝手のよいヘルプ機能が必要である。最初にヘルプ・コマンドが発せられたとき、シェルのヘルプ用のクラスとシェル上のプロセスが持つヘルプ用のクラスをまとめて継承するヘルプ辞書クラスを作る。

ヘルプ辞書は、キーワードとコンテキスト識別子により検索される。キーワードがヘルプ辞書中に存在しない時は、ヘルプ辞書中の類義語テーブルを再検索する。ヘルプ辞書で検索される内容は、文字列と説明語句の並びで表現され、検索内容がウィンドウに表示される。説明語句は、さらにヘルプが可能な語句であり、ウィンドウには白黒反転で表示される。

説明語句は、辞書引きのキーワードと識別子、またはキーワードとヘルプ手続を実行するクラス名からなる。マウスにより、説明語句のヘルプが要求されると、記載内容に従い、さらに辞書引きしウィンドウを開くか、ユーザ定義のヘルプ手続が呼ばれるかが選択され実行される。

6. 統一環境に向けて

シェル上のツール群やカスタマイズされたシェル間の操作方式は、統一が取れていることが望ましい。そのため、ツールが入出力に使うクラスやシェルが使うクラスをパッケージにまとめ、継承して使えるように充実する。また、ツール間で共用するサブプロセスなどの資源もシェルが管理する。

LTBの辞書や文法は複数のトランスレータを介して変換されてツールに組み込まれる。そのため、ファイルの種別管理や変換のログ機能を持つファイル・システムを作る必要がある。さらに変換されたファイル自体にも、標準化されたコメントの形で、トランスレータ名、入力ファイル・リスト、作成日、編集日を自動的に付与し、履歴管理やmake機能を実現する。

7. おわりに

本稿では、LTBシェルの中で、実装の終わっているプロセスの通信・管理方式やヘルプ機能を主体に述べた。今後は、FGCS'88に向けて統一環境の構築とその充実を進めていく予定でいる。