LTB－シェルの構成

4C-2

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、ユーザは関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を表す引数が異なり、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、送受のアクセスが起きたことを知らせるクラスを動的に設定することができ(Notify機能)、実行時間の計測や入出力データのウインドゥ表示などができるようになっている。

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、送信用のストリームを選択する機能を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するとともに、ヘルプの途中で例題を実行することなどを可能にした。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。

シェルを起動すると、核プロセスとシェルのウインドゥ・プロセスが生成され、シェルを起動したプロセスに核プロセスと通信するためのオブジェクトが返される。このオブジェクトを用いて起動するユーザ・プロセスの指定や、通信路の配線法の指示などを行う。シェルを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可能になる。

Design of LTB Shell

Takashi TAKIZUKA* and *Ryouichi* SUGIMURA**
* KDD Kamifukuoka R & D Laboratories, ** ICOT

ウインドゥ・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に対し読み込みを行うと、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インドゥを消去するコマンドを、核プロセスを経由してウインドゥ・プロセスに送ることができる。ウインドゥ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、シェルのウインドゥが再度開かれる。

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一手順で処理しているため、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、その下で別のシェルを起動することもできる。シェルは、シェルを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。その通信路により、あるツール内で、ツール独自の開発環境に移りプログラムを修正し、元の環境に戻って実行を継続することも行える。

プログラム中にクラス名を埋め込んでしまうと、別のクラスに置き換えるときや、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。

定義ファイルに、プロセス名と関連情報(プログラム名,通信方式,アイコンなど)との対応を記述する。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、および名前と番号の対のテーブルから成るクラスを動的に作成しコンパイルする方式の2方式で実現される。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、使い勝手の良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全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、検索内容がウインドゥに表示される。説明語句は、さらにヘルプが可能な語句であり、ウインドゥ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インドゥを開くか、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

シュル上のツール群やカスタマイズされたシュル間の操作方式は、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ュ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ュルが管理する。

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、ツールに組み込まれる。そのため、変換のログ機能やファイルの用途別管理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機能を実現する予定である。

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。