ショートノート

UNIX アプリケーションの機能拡張とプロセス間通信に関する一考察†

三好 力†† 深海 悟†† 馬野元秀†††

UNIX 上で，Lisp や Prolog の処理系など既存のアプリケーションを利用して開発されたシステムに対して，ウインドウを用いたユーザインタフェースなどの機能拡張を行う場合，システムの一部を他プロセスとして記述したいという要求がしばしば生じる．この場合，UNIX プロセス間の通信を行う必要がある．本論文では，この要求に対して，選択可能な方法を検討した上で，Common Lisp 上のシステムと通信を行うファジィプロダクションシステム用のオブジェクトエディタを開発するに際して用いた解決方法を述べる．この方法の主な特徴は，アプリケーションの拡張を行う必要がないこと，アプリケーション側のプログラム開発者は特に通信を意識する必要がないことである．

1. はじめに

UNIX 上で，Lisp や Prolog などの言語や既存のデータベース用言語などを用いてすでに開発されたアプリケーションに対して，使いやすさを向上させるためにウインドウによるユーザインタフェースなどの機能拡張を行うことが多い．UNIX で X Window を利用する場合，効率的な開発を行うには，開発用言語としてC言語の利用が一般的である．したがって，システム全体を複数のプロセスで構成することになり，アプリケーションは利用者と他のプロセスの両方と情報交換を行う必要が生ずる．しかし，多くの言語には他のプロセスと通信する機能が提供されておらず，その仕組みを開発する必要がある．

この論文では，従来のプロセス間通信での問題点を指摘した上で，我々がファジィエキスパートシステム開発支援ツール LIFE FEShell[2)~4)]を開発するに際して用いた方法について述べる．

2. 動機

LIFE FEShell には，ファジィプロダクションシステム（以下 FPS と略する）と，その知識を編集するオブジェクトエディタ（以下 FPOE と略する）がある．FPS は推論エンジン部分を Common Lisp（以下 CL と略する）で記述している．FPOE はC言語で記述され，X Window を用いて FPS の知識編集のためのユーザインタフェースを提供する．FPOE はFPS のデバッグ・チューニング時にも利用するため，FPS と FPOE との間で双方向の複雑な情報交換が発生する．

システム全体は図1のような構成になるので，CL と C 言語との間では何らかの方法で通信が必要である．この部分についての検討が本研究の動機となった．

3. プロセス間通信の方法と問題点

システムの開発にあたって，システム規模が大きく要求が複雑な場合は，各部分を独立のプロセスとして開発する場合が多く，プロセス間通信を行う必要が生じる．

UNIX はプロセス間通信の方法として，パイプ，ソケット，共有メモリなどを提供している．これらの方法のいずれを利用する場合も，各プロセスごとに通信のための特別な処理を必要とし，例えば，割り込み処理や，ループでの入力の監視などを，通信を行うプロセスごとに記述する必要がある．

しかし，ベースとして Lisp や Prolog などの言語や既存のデータベース用言語など（以下ベース処理系と呼ぶ）を用いてアプリケーション（以下，ベース処理系と開発部分を含めて単にアプリケーションと呼

† A Consideration about Functional Extension of UNIX Applications and Inter-process Communication by TSUTOMU MIYOSHI, SATORU FUKAMI (Laboratory for International Fuzzy Engineering Research) and MOTOHIDE UMANO (Department of Precision Engineering, Faculty of Engineering, Osaka University).

†† 国際ファジィ工学研究所
††† 大阪大学工学部精密工学教室

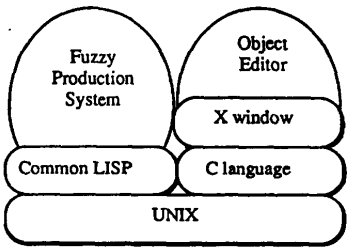

図 1 LIFE FEShell の構成
Fig. 1 Structure of LIFE FEShell.

ぶ) を開発している場合には，これら UNIX のプロセス間通信機能を利用するための組み込み機能は提供されていないことが多い．そこで，他のプロセスとの入出力も処理できるようにベース処理系自身を拡張する必要が生ずる．一般的な例を図 2 に示す．

図 2 において，機能拡張部はユーザインタフェースとして各種のウインドウを提供している．したがって，アプリケーションは，利用者とは X Window のターミナルウインドウから標準入出力 (stdin と stdout) を利用して情報交換し，機能拡張部とは通信用に記述された専用インタフェースを用いて情報交換を行っている（一般的には，機能拡張用の任意のプログラムであってもよい）．図 2 では機能拡張部はひとつのプロセスであるが，複数のプロセスからなる場合も同様である．

通信用インタフェースを開発するにあたって，プロセス間通信は UNIX が提供するいずれの機能を用いてもよいが，アプリケーションと機能拡張部の双方に処理部（図 2 の網かけ部分）が必要となり，ベース処理系を拡張することになる．この拡張には，ベース処理系に対する高度の専門知識が必要であり，各処理系固有の問題を解決する必要が生じる．

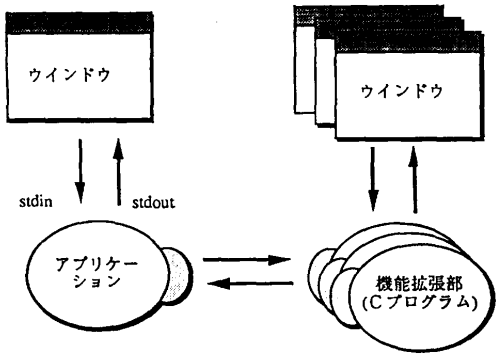

図 2 普通の通信方法
Fig. 2 Ordinary communication method.

4. 提案する通信方法

ベース処理系の拡張を行わないで，利用者だけでなく，他のプロセスとも，標準入出力を利用して通信を行い，機能拡張を行うことを考える．

今回は，X Window を用いたシステムを前提とするため，アプリケーションはターミナルウインドウを利用して利用者との情報交換を行っている．標準入出力を含む管理用モジュールを開発すればアプリケーションの標準入出力も機能拡張部で管理することができる（図 3 参照）．

利用者からアプリケーションへの入力は，機能拡張部の通信用インタフェース部を経由して stdin を通してアプリケーションに送られる．他のプロセスからの情報もこれと同様に送られる．アプリケーションから機能拡張部に対しては，エスケープシーケンスを付加することで，stdout からの情報が利用者に対する表示情報なのか他のプロセスに対するものなのかの識別を行う．

図 3 で示すとおりこの方法では，通信用インタフェースをすべて機能拡張部側で記述することができる．そのため，ベース処理系は外部との通信を標準入出力だけで行うことができ，通信のための拡張をする必要がなくなる．さらに，開発が機能拡張部に限定されることで，デバッグが容易になるなどの開発効率の向上が期待される．

標準入出力を利用すると改行までのバッファリングによる遅れが生じる．これは，ベース処理系ではキー入力を前提とするため，あまり問題にならないが，機能拡張部で文字単位の通信に比べ処理が遅くなる可能性がある．

この方法の利点は，ベース処理系を拡張し，通信のための特別な処理を記述する必要がないことである．

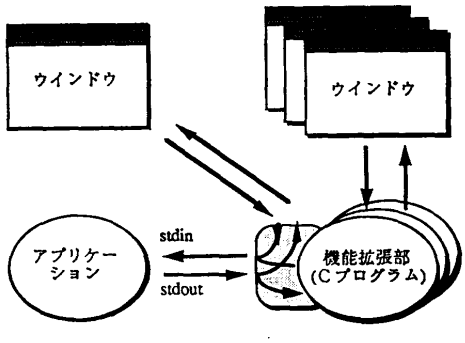

図 3 採用した通信方法
Fig. 3 Proposed communication method.

利用者からの入力と他プロセスからの入力を識別する必要がない場合には，アプリケーション上の記述を全く変更する必要がない．

機能拡張の範囲がユーザインタフェースの向上程度である場合，アプリケーションにほとんど手を加えることなく拡張することができる．さらに複雑な機能拡張の場合も，通信にあたって通信相手を指定する程度の処理で拡張することができ，従来の方法に比べ，大幅に作業量が少なくなる．

この方法でも，拡張部に通信用インタフェース部を作成する必要があり，標準入出力での判定が付け加わるが，他の方法に比べ実現が容易である．

5. ファジィプロダクションシステムとオブジェクトエディタ

図1のように，LIFE FEShell には，ファジィプロダクションシステム（FPS）と，その知識を編集するオブジェクトエディタ（FPOE）がある．

FPS は馬野のシステム[1]を基に拡張を行ったものであり，研究用であることを考慮して，推論エンジン部を Common Lisp で記述している．FPOE は X Window を用いて FPS の知識編集のためのユーザインタフェースを提供する．FPOE はFPS と独立に別プロセスとして開発され，C言語で記述されている．

一般に，推論エンジン部分で，ユーザインタフェースやウインドウシステムを意識して設計されているものは少ない．FPS では，知識入力の方法として，1)定まった形式に従ってファイルとして作成された知識を読み込む機能，2)同様の形式に従って利用者がキー入力を行う機能を持っている．1は主に初期の知識入力に，2は主に知識のチューニングに利用される．

FPOE は利用者の知識情報の入力を助け，誤りを少なくすることを主な目的として開発した．1のインタフェースを改善するため，FPS が扱う知識ファイルと同様の形式で知識の読み出し，書き込みを行う機能を持っている．これによって利用者は FPOE のインタフェースを利用して，FPS と独立に知識の編集が可能になる．

2のインタフェースを改善するためには，FPS と FPOE が互いに通信を行い，FPOE のインタフェースを使って編集した知識を何らかの方法で FPS の側に引き渡す必要が生じる．利用者からの入力にはFPS に対するコマンドなど，知識以外の入力も多いため，FPS には FPOE と利用者の両方からの入力が必要である．FPOE のインタフェースを FPS のデバッグ・チューニングに利用するには，FPS から FPOE に制御コマンドを送る必要がある．

この問題を解決するため，提案した通信方法をもとに，FPOE の一部として FPS の入出力用ウインドウを開発した．このウインドウは，FPS のユーザインタフェースであるとともに，FPOE との情報交換をおこなうための径路にもなっている．ウインドウ起動時にアプリケーションを子プロセスとしてフォークし，通信にはパイプを利用した．これはアプリケーション側に負担（プログラムの書き換えなど）が最も少ない方法である．

FPOE からの知識情報は FPS が取り扱う形式に変換され，stdin を経由して FPS に送られるため，FPS 側では，知識情報が FPOE からのものか利用者がキー入力したものかの区別をする必要がない．このため，FPS は FPOE からの入力を管理する部分を持つ必要がなくなった．

FPS から，FPOE に対する制御コマンドを送る場合，行頭の特定の文字列（エスケープシーケンス）で表示用文字列と識別している．このため，CL に標準のプリント関数を用いてコマンドを送ることができる．

CL では改行を入力の区切りとしており，FPOE に対する制御コマンドも改行で終了するので，通信はすべて行単位に行われる．

なお，FPOE は何等かの原因でシステムがダウンした場合に備え，定期的にデータベースのバックアップを行う予定である．

このように，双方向通信が必要な場合も，FPS 側では開発を大幅に簡略化することができた．これは，FPOE との通信に際して，FPOE のコマンドを管理する部分を開発するだけでよく，通信処理を含めて，Lisp プログラムのレベルですべての処理を記述することができるためである．

6. おわりに

UNIX 上で X Window を使ったユーザインタフェースを提供するようにアプリケーションの機能拡張をするに際して，アプリケーションに特別の処理部を記述することなく他プロセスと通信する方法を提案し，ファジィエキスパートシステム構築支援ツール LIFE FEShell 上での実現例について述べた．

Lisp や Prolog などの言語処理系によってシステ

ムを開発する場合の多くは，研究用やプロトタイピングであるため，あまり本質的でない部分はできるかぎり簡単化して開発の効率を上げたい．各言語処理系の拡張のためには，処理系固有の問題を解決する必要がある．また，既存のアプリケーションの拡張はライセンスの問題などの，技術面以外の問題をひきおこす可能性が高い．

以上のような理由により，UNIX 上で既存の言語系などを利用して開発されたシステムを機能拡張するに際して，今回提案した通信方法は有効であると考える．

参考文献

1) Umano, M.: Implementation of Fuzzy Production System, *Proc. of 3rd IFSA Congress*, pp. 840-843 (1987).
2) 深海 悟，三好 力，小山 宏，高木友博，馬野元秀：ファジィエキスパートシステム構築支援ツール LIFE FEShell，第6回ファジィシステムシンポジウム，pp. 259-262 (1990).
3) 小山 宏，三好 力，深海 悟，高木友博，馬野元秀：LIFE FEShell におけるファジィプロダクションシステム，第6回ファジィシステムシンポジウム，pp. 263-266 (1990).
4) 三好 力，深海 悟，小山 宏，高木友博，馬野元秀：LIFE FEShell におけるオブジェクトエディタ，第6回ファジィシステムシンポジウム，pp. 267-270 (1990).

(平成2年10月5日受付)
(平成3年5月7日採録)

三好 力（正会員）

1961年生．1985年大阪府立大学工学部電気工学科卒業．同年シャープ株式会社入社．技術本部にて，日本語文書処理システム，構造化文書，マルチメディア文書の研究を経験．1989年技術研究組合国際ファジィ工学研究所に出向．ファジィエキスパートシステムシェル，ファジィフレームシステムと知識表現に関する研究に従事．現在同研究所主任研究員．日本ファジィ学会，ACM 各会員．

深海 悟（正会員）

昭和29年9月生．昭和52年3月，大阪大学基礎工学部情報工学科卒業．昭和54年3月同大学大学院修士課程修了．同年日本電信電話公社横須賀電気通信研究所入所．現在 NTT データ通信(株)開発本部所属．この間平成元年4月から平成3年3月まで技術研究組合国際ファジィ工学研究所に出向．工学博士．

馬野 元秀（正会員）

1951年生．1974年，大阪大学基礎工学部情報工学科卒業．1979年，同大学院後期課程修了．工学博士．同年，岡山理科大学理学部応用数学科講師．1985年，大阪大学大型計算機センター助手．その後，同センター講師，助教授を経て，1991年より，同大学工学部精密工学教室助教授．ファジィ集合論の応用，特に，プログラミング言語，データベース，知識情報処理への応用に関する研究に従事．電子情報通信学会，日本ソフトウェア科学会，人工知能学会，システム制御情報学会，日本ファジィ学会，ACM, IFSA (International Fuzzy Systems Association) 各会員．