統計数理 (2001)
第 49 巻 第 2 号 333–343

Java 言語による統計解析システム Jasp

藤原 丈史[1]・中野 純司[2]・山本 由和[3]・小林 郁典[4]

（受付 2001 年 4 月 13 日；改訂 2001 年 9 月 17 日）

要　　旨

安価で強力なパーソナルコンピュータとインターネットの普及により，計算機の環境および技術は，現在，大きく変化し，発達している．統計解析の際に，それらを効果的に利用できるようにするための統計解析システムを開発することは，今後の統計学の普及と発展のために欠かせないものである．

われわれは，最新の技術を容易に利用できるように設計されている Java 言語を用いて，新しい統計解析システム Jasp (JAva based Statistical Processor) を開発している．Jasp は，先進的な計算機技術を統計解析において簡単に利用できるようにすることを目的とする．そのために，関数を中心とする言語とオブジェクト指向言語を融合した統計解析のための言語，独立に，または統合的に利用できるグラフィカルユーザインタフェースとキャラクタユーザインタフェース，ネットワークを用いた分散処理，他言語とのインタフェースを含む高い拡張性，などを実現している．

本稿では，既存の統計システムを考察しながら，Jasp の必要性，目的および特徴などを述べる．

キーワード：サーバ／クライアント，統計解析システム，分散処理，ユーザインタフェース，Java.

1. Jasp を開発する必要性

近年，コンピュータ環境およびそれを実現している技術は，ハードウェアとソフトウェアの両面において目覚しく進歩し続けている．ハードウェアにおいては，CPU 処理速度の向上，記憶容量の増大，価格の低下が急速に進んでいる．以前のメインフレームコンピュータの性能が，現在では低価格のパーソナルコンピュータで実現されており，一般のユーザが複雑な処理を机上で処理することが可能となっている．そして，通信関連のインフラストラクチャの整備により，世界規模でコンピュータが結合されるようになっている．

ソフトウェアもまた，ハードウェアほど数値的に比較可能なものではないが，着実に進化している．オブジェクト指向技術の浸透，グラフィカルユーザインタフェースライブラリの整備，対話型開発環境の普及などは，プログラムのスタイルを変化させたと言える．また，ソー

[1]総合研究大学院大学 統計科学専攻：〒106–8569 東京都港区南麻布 4–6–7; fuji@ism.ac.jp
[2]統計数理研究所：〒106–8569 東京都港区南麻布 4–6–7; nakanoj@ism.ac.jp
[3]徳島文理大学 工学部：〒769–2193 香川県志度町志度 1314–1; yamamoto@is.bunri-u.ac.jp
[4]徳島文理大学 工学部：〒769–2193 香川県志度町志度 1314–1; ikunori@es.bunri-u.ac.jp

スコードの自由な(再)配布，派生的作業(修正)の許可，著者とソースコードの一貫性の主張，個人や集団，適用領域，ライセンスなどに関する無差別性などを特徴とするオープンソースの開発モデルは，ソフトウェア資源に関する人々の考え方を一変させつつある．

そして，ハードウェア，ソフトウェア両方にまたがる重要な革新として，インターネットの爆発的な普及がある．インターネットは当初，大学や研究所など研究機関のものとして一部で使われているにすぎなかったが，現在では一般のユーザでも常時接続が可能となり，ネットワーク自体を身近なリソースとして利用できるような状態にある．この環境では，数値計算を必ずしも必要としない大多数の人々にとっても，コンピュータが，個人用情報処理装置として必要不可欠なものとなった．

このようなコンピュータ環境の発展を背景に，新たな開発言語として登場した Java 言語は重要な技術である．Java 言語は，既存の開発言語にない，または実現が容易ではない様々な機能や特徴をもつ．例えば，オペレーティングシステムに依存しない汎用的な実行環境，オブジェクト指向プログラミング，グラフィックスやインターネット通信に関するよく整備された標準ライブラリなどがあげられる．そして膨大な関連文書がインターネット上で公開されている[1]．

このように，コンピュータ環境はハードウェアとソフトウェアが独自に，さらにはお互いに融合しながら発展してきている．このような状況において，統計解析システムも，また，変化し発展している．商用の統計システムとして，SPSS[2] や SAS[3] といったメインフレームの時代から存在する統計解析システム，それよりやや新しい S (Becker et al. (1988)) および S-PLUS[4] や XploRe[5] (Härdle et al. (1999)) などは，その信頼性の高さや，提供されている豊富な機能などにより，非常に多くのユーザを獲得している．これらの統計システムでは，初期の設計を維持しながらも新しい技術をうまく取り入れ続けており，その機能や使いやすさは十分に改良されている．ただ，多くの過去の資産を持ち，信頼性が最も大切なものであるという性格上，まったく新しい機能を新たに備えることはかなり難しい場合がある．また，基本設計時点の技術状況が現在と大きく異なっているために，新しい技術の導入がうまくいっていない点も見受けられる．

一方，研究用の統計システムでは，商用のシステムとは異なり，コンピュータ環境の発展を取り込んだ，先進的な機能を実現しているシステムがいくつか提案されている．例えば，Lisp-Stat[6] (Tierney (1990)) は Lisp 言語を基本として，新しい統計グラフィックスを作成することが容易であるように設計されている．Manet[7] (Unwin et al. (1996)) は Macintosh コンピュータの優れたグラフィカルユーザインタフェースを利用して欠測値の処理や新しいグラフの実験的な実装を行っている．他にも多くの実験的なシステムが発表されているが，そのほとんどのシステムは実用に使われているとは言えない．ただ，S と互換性を持つ R[8] は，十分な実用性を持った，はじめての広く使われている非商用統計解析システムと言えるだろう．

これらの商用または研究用の統計解析システムにおいても，作成後のコンピュータ環境の発展に応じて，ある程度改良されているものも多い．しかし，それらはごく一部の機能の付加であったり，当初からの機能との整合性が不十分であるなど，新しい技術を効果的に利用しているとはいいがたいものがほとんどである．現在のコンピュータ環境を十分に利用するためには，新たな設計の統計解析システムが必要であると思われる．

ところで，パーソナルコンピュータの普及によって，それを用いて統計解析も行ってみよう，と考えるユーザは激増している．そのような初心者に使いやすいシステムを作成することは，統計解析の普及のためには非常に重要である．また，専門家が道具として使いやすいシステムを作成できれば，(計算機)統計学のあらたな発展の手段を提供することにもなる．従って，現在のコンピュータ環境や技術を統計の分野で最大限に活かし，様々な分野のユーザに利用してもらうことを念頭に置きながら，新しい統計解析システムを開発することは統計学の重要な一

分野であろう．

このような問題意識から，われわれは統計解析システム Jasp (Java based statistical processor) を開発することにした (Nakano et al. (2000))．われわれは先進的なコンピュータ技術を効果的に，そして容易に導入するために，システム構築のための開発言語として Java 言語を選択した．そして，統計解析のための分かりやすく高機能である言語，キャラクタユーザインタフェース (CUI) とグラフィカルユーザインタフェース (GUI) を融合した使いやすいユーザインタフェース，大規模な処理のための分散処理，ユーザと開発者両方にとって有用な拡張機能などの特徴を有する新たな統計解析システムとして，その改良とさらなる機能の付加を行っている．

本論文では，現時点での Jasp の主要機能に関する考え方と基本的な設計方針を述べる．

2. Jasp の機能および特徴

2.1 統計解析言語

実際の統計解析において，システムに組み込まれている解析手法を一通り用いるだけで十分であるということはほとんどない．ユーザはそれらを複雑に組み合わせたり，さらには実装されていない手法を自ら新たに作成する必要があることが多い．そして，新しく作成された手法は別の解析にも利用できるだろう．現在のところ，これらは統計解析言語を用いることにより実現されている．したがって，統計解析言語には，プログラムの作成が容易で，なおかつそのプログラムを再利用しやすいような言語であることが重要である．われわれはそのために，独自の新たな統計解析言語を開発することにした．その主たる特徴は，関数を中心とする言語とオブジェクト指向言語の両方の機能を併せ持っていることである．

関数を中心とする言語は，S や XploRe など，比較的新しい統計解析システムで採用されている．関数を用いることにより，数式の記述がしやすく，簡単な処理をモジュールとしてまとめやすい．そして簡単な関数を単純に組み合わせることで，複雑な処理を組み立てることができるので，初心者でも理解しやすく，直観的であると言える．ただし，作成された関数を関連するものでまとめたり，整理して記録や提示を行う機能などは，一般的には整備されていない．よって，多数の関数を作成した場合には，それらの検索を行い，的確に使用することはそれほど容易ではない．

このような関数を中心とする言語の欠点が，オブジェクト指向言語では改善されていることが知られている．オブジェクト指向言語は，データ（属性と呼ばれる）とそのデータに対する処理（メソッドと呼ばれる）をオブジェクトとしてひとつにまとめることで，現実世界の問題をそれに対応させて記述するものである．また，共通の構造を雛型（クラス）としてまとめ，そこから基本的な部分を受け継ぎ，異なる部分だけを記述することが可能となる継承（インヘリタンス）という機構があり，再利用性の高いプログラムを作成できる．これらは，グラフィカルユーザインタフェースの作成，シミュレーション，知識表現などの大規模なプログラミング作業での有効性が実例によって認められている．

統計の分野でも，このオブジェクト指向の考え方を用いれば，関連する統計手法をメソッドとしてひとつのオブジェクトにまとめることが可能となる．統計解析の際，初心者は自分の分析したい問題に対して，さまざまな統計手法の中から適当なものを選択することが困難である．そのため，組み込まれている多くの統計手法をいかに適切に整理して提示するかということが問題であるが，クラスを使用することはそのためのひとつの解決策と考えられる．さらに，継承機構による差分プログラミングの技術により，統計手法の整理と効率的なプログラムの記述が可能となる．ただ，オブジェクト指向のプログラムを行うのは，作業全体についての考えがまとまったあとでの整理には便利であるが，統計解析の最初の段階のように，考えられる手法

を試行錯誤的にいろいろ試してみるという操作には適していないことに注意しなければならない．どのような統計手法をオブジェクトとしてまとめればよいかは，実際に解析していろいろ試してみて初めてわかるものだからである．

このように，関数型言語とオブジェクト指向型言語はそれぞれ利点と欠点をもつ．そこで Jasp 言語では，この 2 つのタイプの言語の特徴を共に持つように設計している．ユーザは，プログラミングと統計知識のレベルに応じてプログラミングスタイルを自由に選択できる．簡単な統計解析を行う場合や統計学およびプログラミングの初心者は関数を中心にプログラミングし，再利用を考慮した処理や統計手法の整理をするという目的にはクラスを中心にプログラミングをする，ということが可能である．すなわち，解析の初期では，必要な機能を関数により実現する．そして，組み込み済みの関数とユーザ自身が作成した関数を使用して，その返される結果を変数に代入し記録しながら解析を進めればよい．その作業が一段落し，作成した手法や解析するデータの特徴が把握できたときに，必要なものをあとで再利用するためにクラスを作成するのがよい．先に作成した関数はほとんどそのままクラス内のメソッドから利用することができる．そして関数が計算した値はクラスの属性として保持しておく．このように，関連する関数とその結果をひとつのオブジェクトにまとめておくことにより，その再利用が容易になる．

Jasp 言語の実装については，Pnuts 言語[9]を基本とし，統計解析言語として必要な拡張を行うことで実現している（図 1）． Pnuts 言語は，変数の型指定がない，簡単な文法の関数を中心とするスクリプト言語である．また，Java 言語で書かれており，Java 言語との親和性が非常に高く，言語を容易に拡張できるような機能をもっている． Jasp 言語は， Pnuts 言語を基本とすることによって，Pnuts 言語の完成度と簡潔さをただちに Jasp 言語の特徴として引き継ぐことができた．また，Pnuts 言語の特徴である， Java で書かれたライブラリを直接に利用できる機能を用いて，統計解析言語としての機能を持たせている．具体的な機能としては，オブジェクト指向プログラミングを可能とする Jasp Class の実現，行列，グラフィックス，乱数などの標準ライブラリの提供， GUI とのインタフェース，などがあげられる．また，次節で述べる GUI のための補助情報などの付加的な情報を，所定のコメントの形式で埋め込むことができるようになっている．これは言語の文法自体を複雑にすることなく，新しい機能を付け加えるためのひとつの方法を提案したものである．

2.2 ユーザインタフェース

初期の統計解析システムにおいては，ユーザはキーボードからコマンドを打つ，もしくはコマンド列からなるバッチ処理的なプログラムを作り，それを実行することで行いたい操作をシステムに伝えた．そして統計解析システムからの結果は文字列として画面上やファイルなどで受け取っていた．このような方式のユーザインタフェースを CUI (Charactor User Interface) と呼ぶ．このコマンドが統計解析言語の基本であり，したがって統計解析言語では対話的な使いやすさも重要視されてきた．

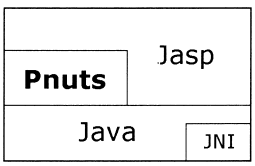

図 1. Pnuts and Jasp.

その後，コンピュータ環境の発展により，GUI (Graphical User Interface) が現れ，統計解析システムのユーザインタフェースとしてもこれが主として利用されるようになった． GUI は，画面上にメニューやアイコンを表示し，マウスでクリックすることで選択，実行を行うなど，視覚的に分かりやすく操作できるようにしたインタフェースである．GUI が好まれる理由として，操作が簡単で，直観的であることが挙げられる． CUI では，コンピュータや統計解析システムの初心者は，コマンドを記憶していなければキーボートでコマンドを打つことさえできない．したがって，コマンドのリストなどの十分な情報を，ポップアップメニューなどの形式で画面上に示すことのできる GUI を用いると，コマンドの記憶が不十分であっても利用できる．したがって，初心者や頻繁にシステムを利用するわけではない人にとっては GUI は非常に使いやすい．

さらに， GUI では，解析結果の履歴などを階層化されたアイコンとして表示することが可能である．これにより，それまでの解析の流れを容易に認識できることとなり，以後の解析の計画が立てやすくなる．そして，それぞれのアイコンを指示すると可能な手法のリストが表示され，そこから手法を実行できるようにすれば，これはデータに対してある操作を行うというユーザの実際の考え方に近いことになる．

しかし，GUI が CUI よりすべての面で優れているわけではなく，GUI が主流の現在でも CUI の利点というものは未だ重要である．多くのコンピュータの専門家やその統計解析システムの使用経験の長いユーザは，GUI による操作より，キーボードから直接コマンドを打つような CUI の方を好む傾向がある．それは，馴れてくると解析法を統計解析言語に直すのが容易になること，GUI は定型的な処理を行ったり，同じ処理を繰り返し行ったりするには，あまり有用ではないといった理由からである．さらに，GUI を用いたプログラミングは未だ研究の段階であり，実用的とは言えない．このように， CUI, GUI には，それぞれにメリット，デメリットがあることは明らかである．

そこで Jasp では， CUI, GUI の両方を活かしたユーザインタフェースの設計を行っている．まず， CUI と GUI がそれぞれに互いに制限を課さないようにしている． GUI を使う人は CUI のことをいっさい気にする必要はなく，逆もそうである．それにもかかわらず，CUI を使っている人が GUI に一時的に移り，その続きをまた GUI で行うことも可能であるようにした．すなわち，GUI と CUI は透過的になっている．これは多くの GUI を好む初心者ユーザが，馴れてくるにつれ CUI に移行する場合が多く，熟練者でも初めてのデータの解析の場合には，解析履歴を GUI 上に残して置きたい場合があることなどを考慮したためである．

現在の Jasp のユーザインタフェースは図 2 のようになっている．左のウィンドウが CUI ウィンドウ，右のウィンドウが GUI ウィンドウである．

CUI ウィンドウでは，上部エリアがエディタとなっており，下部エリアがコマンド入力および結果表示エリアである．エディタの中から一部分または全体を選んで実行することができる．それは一行ずつを下部エリアで入力することと同じである．グラフィック出力の他は，すべての結果が文字列で下部エリアに表示される．

GUI ウィンドウでは，左上部エリアがオブジェクトアイコンエリア，左下部が関数リストエリア，そして右エリアが，グラフやテーブルなどが表示される結果表示エリアである．オブジェクトアイコンエリアでは，変数とオブジェクトがアイコンとして表され，それぞれ階層構造をもっている．ある変数から計算，処理された変数はその下階層に表示されるようになっており，その履歴が階層で表示されている．また，それぞれのアイコンをマウスで指示すると可能な操作がメニューとして出てくる．特にオブジェクトの場合，それはクラスのメソッドであり，GUI 上からこのメソッドを実行できる．大抵の場合，引数を与えるためのダイアログボックスが出現する．

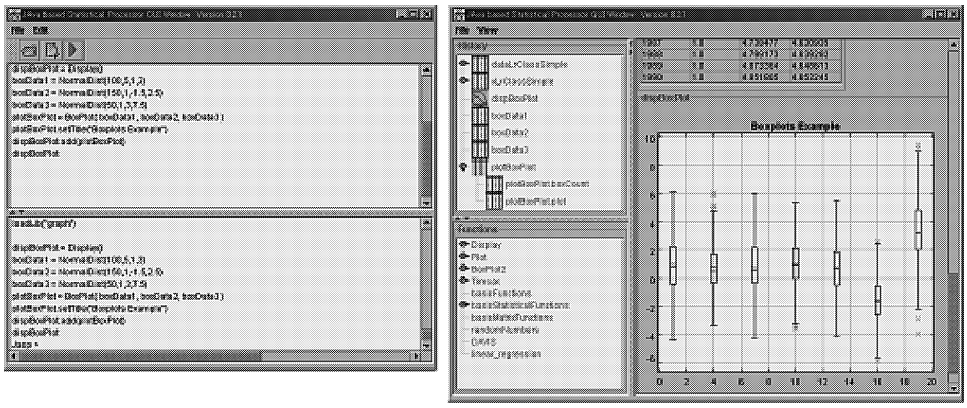

図 2. Jasp user interface.

関数リストエリアでは， function と Jasp Class のコンストラクタ(クラスからオブジェクトを生成するための初期化メソッド)が階層構造で表示されている．この構造は統計手法ごとに整理されたものになっており，具体的には，プログラムを保持しているファイル名とクラスの継承関係から階層構造を生成する．実行したいものをクリックすると，ダイアログボックスが出現するので，先のオブジェクトアイコンをドラッグすることにより，関数の実行と Jasp Class のオブジェクトの生成を行うことが GUI 操作だけで可能である．

また， Jasp のユーザインタフェースでは， CUI と GUI は互いに連携をとっており，片方での操作は，もう一方のユーザインタフェースに記録される．例えば， CUI 上で，ある変数に値を代入するというコマンドを実行すれば， GUI 上のオブジェクトアイコンエリアに対応したアイコンが表示される． GUI 上で，関数を選択し，オブジェクトアイコンをドラッグし実行する，という操作は，CUI 上では対応したコマンドが表示される．このように， Jasp のユーザインタフェースでは， CUI と GUI は統合されており，どちらで操作を行っても同様なことが可能となっている．ただ，プログラミングに関しては Jasp では 現在 CUI 上で行う必要がある．一般的にも GUI によるプログラミングは，先に述べたようにまだ研究段階であり，実用のためには今後のさらなる研究が必要である．

2.3 分散処理

Jasp は，サーバ/クライアント方式で実現している．サーバ/クライアント方式とは，計算量の多い処理を専用に行うコンピュータの「サーバ」と，そこへ処理依頼を行うコンピュータの「クライアント」に分けてネットワークを構築する方法である．前節のユーザインタフェースはクライアントプログラムとして実現されており，実際の計算などはサーバプログラムで行われる．これにより，ユーザが机上で利用するコンピュータと計算を行うコンピュータを分離することができる．したがって，ユーザマシンは，処理の比較的軽いユーザインタフェースを実行するだけの処理性能の低いものでもよく，処理の重い計算は遠隔地にあるワークステーションなどの高機能なサーバマシンで行うことができる．また，ユーザインタフェースプログラムは Java アプレットとしても実装されているので，よく用いられている WWW ブラウザ上から実行することもできる．

ここで考慮しなければならない点としては，サーバとクライアント間のトラフィックである．現在のところ，ネットワーク上のトラフィックは，コンピュータ処理のスピードと比較すると

非常に遅く，頻繁にサーバとクライアント間で通信が行われると全体としての処理効率が悪くなってしまう． Jasp では，一度表示したデータはクライアント側で保持し，同じ通信を行わないようにすることで，サーバ／クライアント間の通信は必要最小限にするとともに，クライアントでもある程度の処理を行うことで，通信量を少なくするようにしている．これは，ユーザのクライアントマシンでさえも，コンピュータ環境の発展により，かなりの性能を期待できる状況になっているからである．

さらに Jasp では，ひとつのクライアントがサーバとして複数のマシンを使用することが可能である． Jasp のクライアントは，まずメインとなるメインサーバとやり取りを行う．ユーザは，より大きな計算が必要となる場合，メインサーバ以外のサーバをプログラムで明示的に使用することができる．このときクライアントは，直接他のサーバとやり取りすることはなく，メインサーバを介して処理することにした．これは主としてクライアントプログラムを複雑にしないためである．また，クライアントだけでなく，サーバも複数のクライアントまたはメインサーバからの処理要求に対応することができる．Jasp 言語には組み込みの関数として，他のサーバにプログラムを送り，それを実行し，その結果を返してもらうという関数を実装している．ただ，分散処理には同期やエラー処理などにかなり難しい問題があるが，現在の関数はまだ基本的機能しかもっていない．よく利用されている並列処理アプリケーション用メッセージパッシングライブラリである MPI (Message Passing Interface) のような既存技術を参考にしながら，さらに研究を進める必要がある．将来的には，ユーザが明示的に他のサーバを使用した分散処理をプログラムするだけでなく，高負荷時にシステム側が自動的に処理を分散させたり，使用可能なサーバの中で現在の負荷が小さいものに自動的に割り当てるようなロードバランサー機能の実現などを考えている．

Jasp でのネットワーク分散処理の実装においては， Java 言語の機能のひとつである RMI (Remote Method Invocation) 機能を利用して実現している．RMI は Java で分散オブジェクト環境（異なるマシン間でオブジェクト同士がメッセージをやりとりできる環境）を実現するための手法およびライブラリである．

2.4 拡張性

拡張性は，統計解析システムとして Jasp を設計する上でもっとも重視した点である．商用システムでも，開発側が多種多様なユーザの要求のすべてを実現することは不可能であろう．また，コンピュータ環境は急速に変化しており，それに迅速に対応するためにも拡張性の高いシステムでなければならない．Jasp では，開発側とユーザがともに容易に利用できるいくつかの拡張性を備えている．すなわち， Java 言語ライブラリの直接の利用，他言語とのインタフェース，統計解析システム XploRe プログラムの実行，などである．

Java 言語ライブラリの利用は， Jasp 言語の基礎となっている Pnuts 言語が，Java 言語のクラス（ライブラリ）を特別な手続きをとることなく， Java 言語と同様な使い方で簡単に使用できることを用いている．これにより，現在利用可能な多くの Java 言語ライブラリを簡単に使用することができる．また，自分でプログラムした Java 言語ライブラリを利用することも同様に簡単である．進んだユーザは， Jasp 言語だけでは実現できない機能や，大量の計算や処理速度が必要なプログラムを Java 言語を利用して作成することができる．Jasp システムの基本ライブラリとして提供されている行列計算，グラフ，分布関数，乱数などのライブラリは，インターネット上で公開されている Java 言語ライブラリを用いて，この方法で実現されている．

他言語とのインタフェースは，特に C と Fortran というマシン固有の実行形式にコンパイルする言語のプログラムを Jasp から利用するためのインタフェースである．この機能により，これらの言語による多くの既存の資産を無駄にすることなく，有効に利用することが可能にな

る． Jasp において標準として組み込まれている TIMSAC パッケージは，このインタフェースを利用し実現されている．TIMSAC（赤池・中川（1972））は，統計数理研究所で開発された，Fortran で記述されている時系列解析パッケージであり，現在でも広く使用されている．このような他言語インタフェースの実装には，JNI (Java Native Interface) を使用している．JNI は，ネイティブコードで Java メソッドを実装するためのクロスプラットフォーム標準規格である．このようなメカニズムは，既存資源の有効利用というほかにも，現在 Java 言語の欠点の一つとも言える処理速度の問題を解決する手段としても有効である．ただし， Java 言語は現在も発展段階にあり，徐々にその処理速度も改善されているため，この利点は将来は重要ではなくなるだろう．また，JNI はオペレーティングシステムや CPU ごとに特有なライブラリ，例えば Windows 環境で実行時に動的にリンクされるダイナミックリンクライブラリ (Dynamic Link Library) や Unix 環境で実行時に動的にリンクされるシェアードライブラリ (Shared Library) が必要であり，Java 言語の利点のひとつである汎用性を損なう原因となる．したがって，既存の資源の再利用以外では，Java 言語によるライブラリの開発を行うべきである．

Jasp では Jasp 言語のプログラムの他に，統計解析システム XploRe の言語で書かれたプログラムもある程度実行可能になっている．これは XploRe のプログラムを Jasp 言語に翻訳することにより実現されている．現在のところ，グラフ処理を除く XploRe 言語で書かれた関数が実行可能であるが，C++ で実装されている関数は Jasp では利用できない．グラフ処理に関しては，XploRe と互換性をもつ汎用的な Java 言語によるグラフライブラリを作ることで対処可能であるが，C++ で書かれた関数に関してはそれぞれを Jasp または Java 言語で書き直す必要があり，個別の対応とならざるをえない．

XploRe は，比較的新しく設計された高機能な統計解析システムであるが，文法が単純で Jasp 言語に近く，その豊富な統計手法のライブラリの大部分が XploRe 言語で書かれているという特徴をもつ．従って，この機能により，Jasp ユーザは XploRe の豊富な統計手法ライブラリを利用することができる．また XploRe ユーザも現在の XploRe に実装されていないようなユーザインタフェースや分散処理機能などを試すことができる．現在では多くの優れた商用ソフトウェアには，互換性のある非商用ソフトウェアがあることが多く，信頼性が重要なら商用ソフトウェアを使い，手軽さや実験的な目的のためには非商用ソフトウェアを利用するというように互いに補いあって発展しているように思われる． Jasp のこの機能は，このような意味でも双方の目的にとって価値のあるものと考えている．

また，明らかに，統計解析システムに多数の統計手法を実装することは，非常に多くの時間的および人的コストがかかる．それにもかかわらず理論的には同じものである統計手法が，それぞれの統計解析システムの言語や環境で何度も繰り返し実装されていることに注意しなければならない．利用するシステムをひとつだけにすることは技術の発展のためには望ましいことではないので，言語や環境の壁を越えて，資源の再利用と有効利用を行えるようにすることが重要である．一般に，同じアルゴリズムの実装作業を繰り返すことはあまり生産的ではないので，今後はソフトウェア間の相互協力が普通になると考えられる．Jasp では，統計言語のレベルでの相互協力，つまりあるシステムの統計言語で記述されたプログラムを，他のシステムの統計言語に翻訳することで異なる統計システム間の相互協力を実現している．他の方法として，実行レベルで直接他の統計システムを起動し，結果を共有するという方法も考えられる．しかし，翻訳によるアプローチの利点である，翻訳後の統計プログラムをユーザがさらに編集できる，ということを重視し，本システムではこちらを採用した．

3. おわりに

統計解析システム Jasp は，急速に発達している最新のコンピュータ環境に対応する使いやすい統計解析システムを目指して開発されている．従来の統計解析システムには見られないような先進的な機能を実現しようとする実験的なシステムと言えよう． Jasp では，コンピュータおよび統計の初心者から研究者までの幅広いユーザの利用を想定し，より使いやすく，高機能で，広く使用してもらえるような統計解析システムを目標にしている．そのために，分かりやすく高機能な Jasp 言語，CUI と GUI を融合したユーザインタフェース，大規模な処理のための分散処理，そしてユーザと開発者両方にとって様々な拡張が可能である機能，というような特徴を持つ．

このような統計解析システムは，実際に使用するユーザの役に立つものでなくてはならないことは明らかである．開発者がいくら先進的で高機能と思うものを実現しても，実際のユーザが使いにくい，または，分かりづらいと感じるものであれば意味がない．そのため，われわれは Jasp のホームページ[10] を開設し， Jasp に関する情報や資料， Jasp のソースコードのダウンロードなどを可能にし，オープンソースポリシーでの開発を行っている．現在は，まだ開発段階であり，きわめて実験的なシステムにすぎないが，ここに寄せられた立場の異なる人々からの意見を元に，システムの完成度をより高めていきたいと考えている．

なお，現在すでに実現されている統計手法としては，グラフと基本的な統計量，重回帰分析，主成分分析，時系列解析の基本的な部分くらいしかない．これも必要なもの，希望の多いものから充実させていくつもりである．

注.

[1]http://java.sun.com/
[2]http://www.spss.com/
[3]http://www.sas.com/
[4]http://www.insightful.com/products/splus/
[5]http://www.xplore-stat.de/
[6]http://www.stat.umn.edu/~luke/xls/xlsinfo/xlsinfo.html
[7]http://www1.math.uni-augsburg.de/Manet/
[8]http://www.r-project.org/
[9]http://javacenter.sun.co.jp/pnuts/
[10]http://jasp.ism.ac.jp/

参 考 文 献

赤池弘次, 中川東一郎 (1972). 『ダイナミックシステムの統計的解析と制御』, サイエンス社, 東京. (英訳: Akaike, H. and Nakagawa, T. (1989). *Statistical Analysis and Control of Dynamic Systems*, Kluwer, Dordrecht; 新訂版: 赤池弘次，中川東一郎 (2000).『ダイナミックシステムの統計的解析と制御 [新訂版]』, サイエンス社，東京.)

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988). *The New S Language*, Wadsworth & Brooks/Cole, Pacific Grove. (邦訳: 渋谷政昭，柴田里程 訳 (1991).『S 言語 ── データ解析とグラフィックスのためのプログラミング環境 I, II』, 共立出版，東京.)

Härdle, W., Klinke, S. and Müller, M. (1999). *XploRe Learning Guide*, Springer, Berlin. (邦訳: 垂水共之 監訳，森 裕一，山本義郎，中野純司，宿久 洋 訳 (2001).『統計解析環境 XploRe

—ラーニングガイド—』, 共立出版，東京.）

Nakano, J., Fujiwara, T., Yamamoto, Y. and Kobayashi, I. (2000). A statistical package based on Pnuts, *COMPSTAT2000 Proceedings in Computational Statistics*, 361–366, Physica, Heidelberg.

Tierney, L. (1990). *LISP-STAT*: *An Object-oriented Environment for Statistical Computing and Dynamic Graphics*, Wiley, New York.（邦訳: 垂水共之，鎌倉稔成，林 篤裕，奥村晴彦，水田正弘 訳(1996).『LISP-STAT』, 共立出版，東京.）

Unwin, A. R., Hawkins, G., Hofmann, H. and Siegl, B. (1996). Interactive graphics for data sets with missing values—MANET, *J. Comput. Graph. Statist.*, **5** (2), 113–122.

A Statistical Analysis System Jasp Written in the Java Language

Takeshi Fujiwara

(The Graduate University for Advanced Studies)

Junji Nakano

(The Institute of Statistical Mathematics)

Yoshikazu Yamamoto and Ikunori Kobayashi

(Faculty of Engineering, Tokushima Bunri University)

The computing environment and technologies have been changed and developed dramatically with the popularization of cheap, high-performance personal computers and the Internet. It is indispensable to design and develop statistical systems that can fully utilize them for disseminating statistical techniques and developing computational statistical studies.

We have developed a new statistical system named Jasp (JAva based Statistical Processor) using the Java language that can use many modern technologies easily and efficiently. It has various advanced features, such as a statistical language that has characteristics of both functional based languages and object oriented ones, an interface that integrates a graphical user interface and a character user interface, distributed computing facilities using a network, and powerful extensibilities.

This paper introduces the research background and purpose, and features of Jasp by considering existing statistical systems.

Key words: Server/client, statistical analysis system, distributed computing, user interface, Java.