テクニカルノート

WWWを利用した集合指向言語SOL遠隔実行環境の構築

横尾徳保† 重松保弘†

インターネットの普及により，WWWで情報を公開することは非常に有用となった．言語処理系の公開に際しては，フォームおよびCGIプログラムを利用して，Webブラウザへのプログラム記述，サーバ上でのコンパイル/実行，Webブラウザでの実行結果の表示という形式でプログラミングを体験できる環境が提供されることもある．しかしながら，このようなシステムでは通常プログラム実行時のデータ入力を対話的に行うことができない．そこで，対話的なプログラム実行を行うことができる遠隔実行環境として，フレームならびにJavaアプレットを利用して集合指向言語SOLの遠隔実行環境を構築した．本稿では，これらの遠隔実行環境のシステム構成を紹介する．また，WWWを利用した言語処理系の公開という視点から，他の言語処理系への応用について述べる．

Design and Development of Remote Execution Environments of Set Oriented Language (SOL) Using the WWW

Noriyasu Yokoo† and Yasuhiro Shigematsu†

Recently, it has become very significant to distribute information using the WWW. In case of presenting programming languages on the WWW, such environments are sometimes constructed that provide users programming experience using Form and CGI. However, in such systems, all input-data are required before the execution of programs. This prevents conversational execution of programs. Thus, we have designed and developed two types of environments which enable conversational remote execution of SOL programs using either Java applet or Frame in addition to Form and CGI. In this paper, we show the system architecture of these remote execution environments as case studies, and also discuss the extension to other programming languages.

1. はじめに

インターネットの発展にともない，さまざまな情報がWWW（World Wide Web）を利用して配信されるようになり，WWWの利用者も増加の一途をたどっている[1]．このようななか，大学などの公的研究機関が研究内容や研究成果をWWWで公表することは社会への大きな貢献であると考えられ，今後そのような要望がますます高まると予測される．

これまで，筆者らは集合指向言語SOL（Set Oriented Language）の設計ならびに言語処理系の開発/拡張を行ってきた[2),3)]．このようなプログラミング言語について知ってもらうにはプログラミングの体験が重要であり，そのためには言語処理系を利用できる環境が必要になる．このような場合，利用者の利便性を考慮するとWWWを利用してプログラミング環境を構築することが望ましい．そこで，筆者らはSOLを対象として，プログラミングおよびコンパイル/実行をWebブラウザから行うことができる遠隔実行環境の構築を行った．

本遠隔実行環境においては，Webブラウザ上でのソースプログラムの入力およびサーバでのコンパイル/実行は最低限必要な機能である．このような機能はフォームとCGIを利用することで比較的容易に実装可能であり[4)]，SOLと類似したプログラミング言語であるSETLにおいても同様のシステムを公開している[5)]．しかしながら，CGIプログラムの特性上，単純な実装では連続したデータ入力を対話的に行うような処理はできない．対話的なデータ入力ができなければ，実行開始時にすべての入力データ列が必要になるが，つねにすべての入力データ列を用意することは不可能である．したがって，対話的なデータ入力は不可欠である．本稿では，これらの点を考慮して開発を行った

† 九州工業大学
Kyushu Institute of Technology

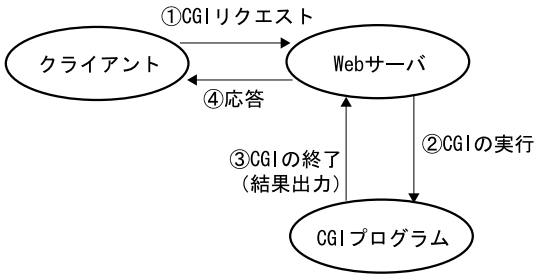

図 1 標準的な CGI の処理手順
Fig. 1 Standard processing model of CGI.

SOL 遠隔実行環境のシステム構成を紹介し，その利便性ならびに他の言語処理系への応用について述べる．

2. SOL 遠隔実行環境の構築

標準的な CGI の処理手順を**図 1** に示す．基本的に HTTP リクエストとして起動される CGI プログラムは，プログラムの実行が終了してから実行結果をレスポンスとしてクライアントへ返す．このために，CGI プログラムを利用すると入力に応じた処理を行うことはできるが，実行経過を見ながら次の入力を行うような継続した処理は通常行えない．紹介するシステムでは，SOL コンパイラ/インタープリタ（以降，SOL 処理系と略す）を CGI プログラムから起動し，そのプロセスは実行させたまま，(1) 非解析ヘッダを持つ CGI プログラムとフレーム機能を利用する方法，(2) 通常の CGI プログラムと Java アプレットを利用する方法，の 2 つの方法で対話的なプログラム実行を実現している．以降，それぞれを Frame 版，Java 版と呼ぶことにする．

Frame 版，Java 版ともに，サーバはクライアントから送信されてきたソースプログラムに対しコンパイルを行った後，コンパイルエラーなどの異常がなければプログラムの実行を開始する．プログラム実行中は，クライアントからの要求に応じて入出力処理を行い実行結果を出力するので，これをクライアントに送信する．CGI プログラムはクライアントと SOL 処理系とのゲートウェイ・インタフェースとなっており，必要な情報の受け渡しを行う．

2.1 システム構成と通信手順

Frame 版のシステム構成と通信手順を**図 2** に示す．サーバ側は Web サーバ，CGI プログラムおよび SOL 処理系，クライアント側はフレームを表示可能な Web ブラウザで構成した．まず，Program Source Frame からの HTTP リクエストで CGI プログラムが起動され（①），一連の処理を開始する．次に，CGI プログラムは②の手順で SOL 処理系を起動し，同時に識別

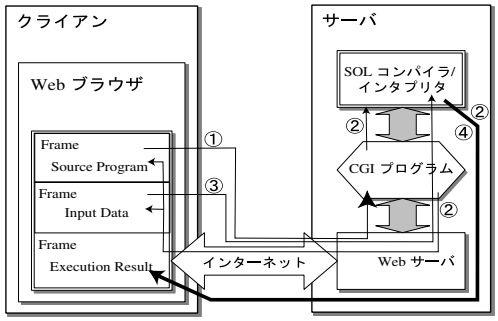

図 2 システム構成と通信手順（Frame 版）
Fig. 2 System architecture and procedure of communication (Frame version).

表 1 クライアント/サーバ間の通信規約（Frame 版）
Table 1 Client/Server communication protocol (Frame version).

Content	Direction	Data format
Send Source	Send	`SRC=[SOL program]`
	Receive	PID
Input/Output	Send	`PID=[PID]&INPUTDATA=[Data]&OPTION=[Option]`
Receive Result	Receive	Output data

用の ID を発行する．この ID を隠蔽情報として Input Data Frame に渡す．このとき，Input Data Frame と Source Program Frame にはこの ID を含め表示する情報がすべて出力されるので，Web ブラウザ上での表示は完了する．また，Execution Result Frame への出力は非解析ヘッダを持つ CGI プログラムを利用して作成する．非解析ヘッダを持つ CGI プログラムの出力は，サーバでバッファリングされずにクライアントに渡され，その結果，逐次的な情報の表示が可能であり，非解析ヘッダを持つ CGI プログラムから SOL 処理系を呼び出すことで逐次的な実行結果の表示が実現できる．データ入力を行う場合には，Input Data Frame からデータ入力を行う（③）．このとき，前述の ID も同時に Web サーバに送り，CGI プログラムはこの ID をキーに UNIX Domain Socket を利用して実行中の SOL 処理系に入力データを渡す．②で確立した実行中の SOL 処理系と Execution Result Frame の接続は SOL 処理系が終了するまで継続されるので，実行結果は逐次的に Execution Result Frame に表示される（④）．また，それぞれのデータは**表 1** の形式で送信される．これらのデータは，フォームデータの形式に従って “名前=値” の対から構成され，それぞれの対はアンパサンド（`&`）で区切られる．

次に，Java 版のシステム構成と通信手順を**図 3** に示す．Frame 版とほとんど同じ構成であるが，プログラミングの支援機能を強化するために，ユーザインタ

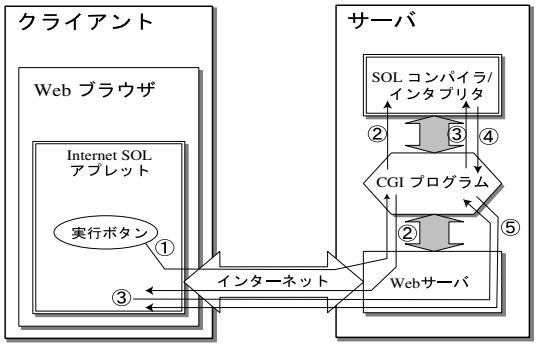

図 3 システム構成と通信手順（Java 版）
Fig. 3 System architecture and procedure of communication (Java version).

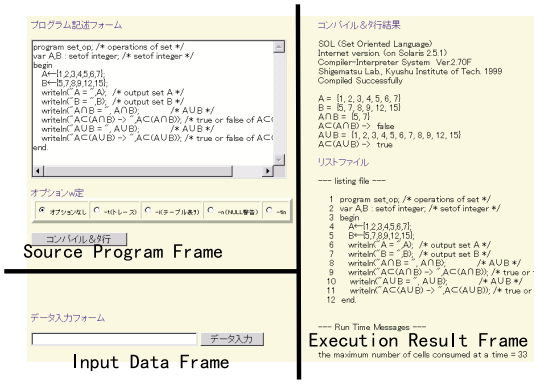

図 4 クライアント側の実行画面例（Frame 版）
Fig. 4 Screen example on client (Frame version).

フェースに Java アプレットを使用した．まず，アプレット上の実行ボタンを押すことで HTTP リクエストを発行し（①），一連の処理を開始する．このとき，②の手順で CGI プログラムが SOL 処理系を起動する．同時に識別用の ID を発行し，この ID をレスポンスとしてクライアントへ返す．以降，アプレットが定期的に HTTP リクエストを発行し，入力データがある場合には，入力データの受け渡しが行われる（③）．このとき，前述した ID を用いて起動中の SOL 処理系のプロセスを特定する．CGI プログラムは③のリクエストがあった時点までの実行結果を SOL 処理系から受け取り（④），その実行結果をクライアントへ渡す（⑤）．HTTP はセッションごとに接続/切断を繰り返し，接続を継続しないために操作時間に比べて通信時間が短いシステムではサーバ資源の節約に有効であり，特に Java 版ではこの利点が活かされている．また，それぞれのデータはフォームデータの標準的な形式に合わせており，基本的に表 1 と同様の形式で送信される．

Frame 版は（1）Java アプレットを使用しないので Java VM の起動負荷がない，(2）フレームを表示可能な Web ブラウザを用意するだけで，容易にクライアント環境が構築でき，ユーザは簡便に SOL プログラミングを体験することができる，(3）言語処理系の仕様に変更があった場合，サーバ上のプログラムを更新するだけで最新版を利用できる，(4）機能の実装を最小限にとどめているので，他の言語処理系へそのままの形で応用ができる，などの特徴を持つ．

Java 版は Frame 版の（3）の特徴に加えて，(1）Java アプレットを用いたプログラム作成支援により，プログラム中に使用できる各種数学記号の簡便な入力，リストファイルやエラーメッセージのマルチウィンドウ表示などを実現し，快適なプログラミング環境を提供

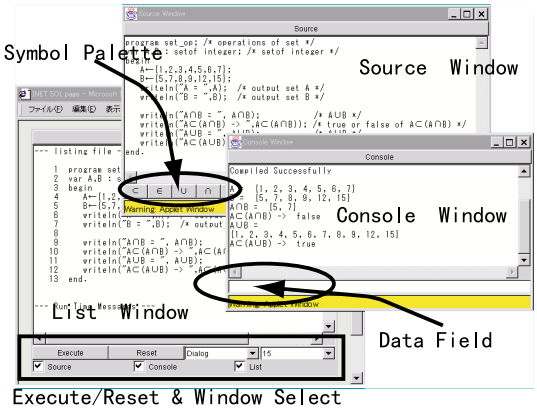

図 5 クライアント側の実行画面例（Java 版）
Fig. 5 Screen example on client (Java version).

する，(2）Java アプレットを実行可能な Web ブラウザを用意するだけで，容易にクライアント環境が構築可能である，(3）Java アプレットの機能を拡張することで，プログラミングの環境向上が期待できる，などの特徴を持つ．

2.2 ユーザインタフェース

Frame 版と Java 版それぞれのクライアント実行画面例を**図 4**，**図 5** に示す．Frame 版は Source Program Frame，Input Data Frame および Execution Result Frame から構成される．使い方は Source Program Frame に用意してあるフォームにプログラムを記述し，実行ボタンを押すことで処理が開始される．その後は，必要に応じて Input Data Frame に入力データを入力すれば，Execution Result Frame に逐次的に実行結果が表示される．

Java 版は Source Window，Console Window および List Window の 3 つのウインドウを中心に構成される．ユーザは，Source Window でプログラムの編集を行い，Execute ボタンを押すことで SOL プログ

ラムをサーバ側で実行することができる．このとき，実行結果は Console Window に表示され，入力データがある場合には，Console Window の下部にある Data Field を利用する．コンパイルに失敗した場合には，List Window でリストファイルを参照することもできる．これらのウインドウはアプレット下部のチェックボックスをマークすることで表示される．初期状態では，Source Window のみ表示される．各ウィンドウはタイトル部分をクリックすることにより，独立したウィンドウとして動作可能であり，自由なレイアウトで作業を進めることができる．また，SOL ではプログラム中に各種の数学記号が記述できるために，これらの入力を支援する Symbol Palette を作成し，Source Window 下部に配置している．

2.3 セキュリティ

本システムは利用者に対してプログラムの遠隔実行を許しているので，サーバへの不正アクセスには注意を払う必要がある．不正アクセスは大別するとサーバ内のファイルを不正に閲覧/書き換えするものと，サーバの許容量を超える多数のアクセス/書き込みなどをしてサーバをダウンさせるものに分けられる．前者については，現在，ファイルの読み出し/書き込みとも手続き自体を禁止しており，コンパイル時にエラーとなるようにすることで問題を回避しているが，制限付きでファイルアクセスを許可することにより，より柔軟なシステムになる可能性もあり，検討の余地が残されている．後者については，現在，すべてのプログラムに対して時間制限を設け，一定時間内に終了しないプログラムは強制終了することで問題を回避している．また，新規にコンパイル/実行を始める前に，実行中のプログラム数をカウントして，一定数以上にならないようにしている．

3. ま と め

本稿では，Web ブラウザが利用可能な環境を用意するだけで，SOL プログラムの作成およびコンパイル/実行を容易に行うことが可能な遠隔実行環境を紹介した．本環境では対話的なプログラム実行が可能であり，快適にプログラム実行を行うことができる．また，マルチウィンドウなどのプログラム作成支援を行うことで，プログラミング環境も向上した．

また，本システムは他の言語処理系への応用を考慮しており，通常の言語処理系であれば，Frame 版，Java 版ともに CGI プログラムの言語処理系の呼び出し部分を書き換えるだけで容易に置き換えることができる．また，Java 版においてはユーザインタフェースに Java アプレットを用いているために，さまざまな付加機能を実装することができる．Java 版における数学記号の入力支援もその 1 つである．このような特殊記号を使用する言語（たとえば APL など）には，Java 版のような入力支援が有効であると考えられる．また，このほかにもデバッグの支援機能なども提供することができる．しかし，機能の充実にともない起動にかかる負荷が大きくなるので，一般の試用を目的とする場合，機能を付加しすぎないことも重要である．逆に，教育での利用を目的とする場合は，さまざまなヘルプ機能を実装し，効果的な教育支援を行うことも可能であると考えられる．なお，本システムは以下の URL で公開している．

`http://www.cs.comp.kyutech.ac.jp/~sol/sol_project/inetsol1/inetsol_index.html`

また，開発環境としては Web サーバならびに Web ブラウザは既存のものを使用し，CGI プログラムは新しく作成した．SOL 処理系についてはセキュリティ対策のためソースプログラムに若干の修正を行った．サーバ側では Web サーバとして Apache，CGI プログラミングには Perl，C 言語および C++言語を用いた．クライアント側での Web ブラウザとしては Netscape Navigator 4.51（Windows）および Internet Explorer 5.0（Windows）で動作確認を行った．また，筆者らのサーバ環境（SPARCstation S-7/300U）では，計測の結果 100 クライアント程度までは同時に対応可能であることが確認できており，言語処理系の試用などの用途であれば十分適用可能であると考えられる．

参 考 文 献

1) 平成 11 年度版通信白書．http://www.mpt.go.jp/policyreports/japanese/papers/99wp/99wp-0-index.html
2) 重松保弘，吉田　将：集合指向言語 SOL の拡張とフローグラフのインターバル解析への応用，情報処理学会論文誌，Vol.34, No.2, pp.229–238 (1993).
3) Yokoo, N. and Shigematsu, Y.: Development and Extension of Set Oriented Language (SOL), *Proc. 2nd International Conference on Parallel and Distributed Computing and Networks*, pp.64–69 (1998).
4) http://www.cs.comp.kyutech.ac.jp/~sol/sol_project/inetsol1/isolform/
5) http://www-robotics.eecs.lehigh.edu/~bacon/setl-server.html

(平成 12 年 1 月 20 日受付)
(平成 12 年 7 月 5 日採録)