

A Interactive Environment for Network Disributed Functions Using Java

佐塚 秀人
久留米工業大学

小川 浩規
久留米工業大学

廣川 佐千男
九州大学

<https://hdl.handle.net/2324/1376273>

出版情報 : 全国大会講演論文集. 55 (3), pp.757-758, 1997-09-24. Information Processing Society of Japan (IPSJ)

バージョン :

権利関係 :



分散する機能利用を支援する Java を用いた対話環境

4 U-1

佐塚 秀人

小川 浩規

廣川 佐千男

久留米工業大学

九州大学

1. はじめに

現在、Web上で型推論と証明探索システムを公開している。本システムは、インターネット上での機能提供として、サーバが提供する機能をWebブラウザで誰でも利用できることを目的に開発した。このシステムで、Lispで作られたGUIをもたないツールに対して、アダプタを介することでJavaを利用してGUIとネットワーク利用環境を提供する[1]。

しかし、このシステムは定型的な問い合わせしか実現できていない。より対話的なシステムの提供を目的として、利用者が自由にネットワークで提供される機能を組み合わせて対話的に利用できるGUI環境をJavaを利用して試作した。

2. ネットワークでの機能提供

証明図の3D表示を行うGUIをTcl/Tkを用いて作成した。このGUI機能をネットワークを介して、公開することを目標としてJavaによるプロジェクトを開始した。

Tcl/Tk 版ではKCLによる証明エンジンをUNIXのバインド機能で結合していた。これをWeb上のJava Appletに書き直す作業を行った。Javaに提供されているAWT(Abstract Window Toolkit)は強力ではないが、比較的容易にTcl/Tkでの機能を実現できた。通信はKCL側に、AppletとTCP/IPを用いて通信するアダプタを作成し、WebのHTTPと独立したコネクションによって、Javaで書かれたAppletと通信する[図1]。

GUI 部と証明エンジンは Lisp のリスト構造(S式)を用いて行う。この機能を実現するために Java で Lisp のリストの入出力と内部表現を扱うクラスを作成した。移植作業は次のようなステップになる。

- (1) Tcl/Tk の GUI 機能の Java AWT による記述
- (2) Java によるリスト処理機能の記述
- (3) 証明エンジンのネットワーク・アダプタの作成

Javaによるプログラミングは、表示と通信に関係する部分のみのため短期間で作業は完了した。昨年の6月からWeb上で公開し、電子メール等でコメントを得ることができた。

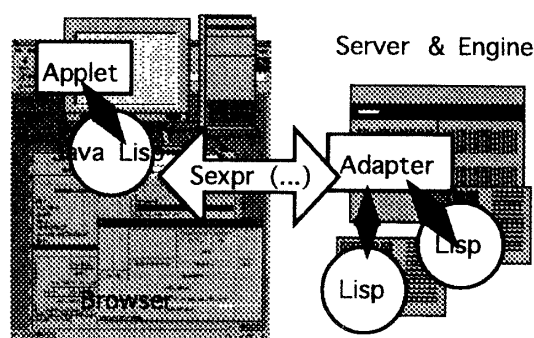


図1 インターネット・ブルーバナーの構成

3. プロトタイプの評価

システムを公開して次にあげる開発課題を得た。

- (1) 他のサーバの機能提供とその結合
- (2) GUIの部品化
- (3) 作業過程、作業結果の保存
- (4) ネットワーク機能の選択

証明エンジンのようなシステムは、他にも多く存在する。それらの機能を組み合わせて利用をしたい。Javaで作られた表示機能を汎用部品化して組み合わせたい。といった意見があった。

プロトタイプで提供した入力例題メニューは初めて利用する人に評判がよく、このようなメニューはGUI 部品化しておくことによって、簡単にWebページ上にはめ込むことができれば便利である。

Webページ構築の意見の一方、本格的にツールを利用したい場合、Webページ提供側の用意したインターフェースでは細かな作業がしにくい。サーバからの出力を保存したり、分散するサーバの機能を組み合わせることができないかといった意見がでた。Appletの出力は現状では保存をしておく方法がなく機能提供の形態も一方的で融通性がない。

このような評価から、機能を部品化する作業を進めることにした。

4. 部品化と結合

4.1 Java による Scheme 処理系

部品化をするにあたって、結合の手段を検討した。Webページ構築には一般に、HTMLの機能や Netscape Navigator 等で提供されている JavaScript を用いることができるが、Listデータの柔軟性を有効活用したいという方針から、JavaでScheme ライクな Lisp処理系を作成した。これは、プロトタイプで用意したリスト表現機能にSchemeの評価機能をもたせプログラミング機能を付加したものである。

組み込み関数は最小にとどめ、Java のクラスとして機能記述し、ダイナミックに組み込めるようなものにした。これによって、Javaで書かれた部品をダイナミックに組み込み、Lispプログラムの枠組扱うことができるようになる。また、この処理系自体 Javaのクラスとして実現されているため、同時に複数の独立したインスタンスを生成し、通信による協調処理をねらっている。

Webサーバ側でもJava処理系を動かすことにより、部品の結合はSchemeのオブジェクトの結合として表現できる。

4.2 デスクトップ環境

ネットワーク上の機能と、GUI機能を結合する UNIXのShellのようなビジュアル・スクリプト環境を Javaで実現できないかということで試作を行った。

このデスクトップは、対話処理によるネットワーク機能の利用、結合を支援する。現在のシステムは、通信するオブジェクトをLispのリスト形式で行っているため、中間結果を保存しておいて、それを加工したり、ビジュアライズさせたりといったことが比較的容易に実現できる。分散するSchemeのビジュアライズ環境という位置付けである。

5. ビジュアル・デスクトップの実現

5.1 ドラッグ・アンド・ドロップ

ネットワーク上のエンジンや、JavaによるGUI機能、その他ユーティリティをICONとして表現する。機能ICONをクリックすると、入力用のウィンドウまたは、入力選択のメニューを表示し、入力データの生成を促す。

入力データは、内部のLispデータとして表現される。リストデータで表すデータICONが表示され、このICONはローカルなGUI機能を表すICONやネットワーク上の機能へのチャンネルを表現するICONにドロッ

プすることによって、データが送られ、結果のデータはICONとしてデスクトップに現れる。

5.2 ICON の結合によるツールの結合

機能を表現するICONを結合して、UNIXのPipeのような結合を実現できる。新たなICONを登録し、クリックすることによって、登録用のWindowが表示される。そのウィンドウに機能ICONをドラッグし、接続を表現する矢印で結合することによって新たなマクロ機能を登録できる。

6. 現状と課題

Javaによるスタンド・アロン・プログラムとしてデスクトップ環境を作成中で、プロトタイプが動いており、Webで提供した機能の対話的な利用と、ツールの結合ができている[図2]。

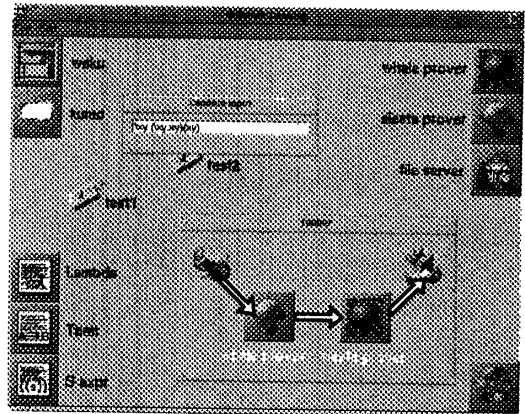


図2 ネットワーク・デスクトップの画面

課題として、機能提供側の枠組みを検討しなくては行けない。既存のツールの機能を提供する場合、柔軟性を考えた処理系は、セキュリティ・ホールを作り出してしまふ。それをふさぐ一般的な方針は得られていない。

7. 終わりに

インターネット上にサーバとして機能提供を行うメカニズムと、それを対話的に統合するシステムの試作について述べた。通信をLispのリストデータとし、GUIを利用してダイナミックに結合する環境を用意することで、ネットワーク上の機能資源を容易に利用できるようになることを示した。

参考文献

- [1] 佐塚秀人, 長沢武司, 廣川佐千男, 分散する証明推論エンジンのWEB上での結合について, マルチメディアと分散処理ワークショップ, 1996